

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université A. Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Laboratoire d'Informatique Médicale (LIMED)

THÈSE

Présentée par

SKLAB Youcef

Pour l'obtention du grade de

DOCTEUR EN SCIENCES

Filière : Informatique

Option : Cloud Computing

Thème

**MÉCANISMES DE FORMATION DE COALITIONS MULTI-AGENTS AVEC
EXTERNALITÉS**

Soutenue le 09/09/2021 devant le jury composé de :

M. SEBAA Abderrazak	MCA	U. de Béjaia, Algérie	Président
M. TARI Abdelkamel	Professeur	U. de Béjaia, Algérie	Rapporteur
M. AKNINE SAMIR	Professeur	U. de Lyon 1, France	Rapporteur
M. AHMED NACER MOHAMED	Professeur	USTHB, Algérie	Examineur
M. MARILLEAU NICOLAS	HDR	IRD, France	Examineur
M. KHANOUCHE MOHAMED ESSAID	MCA	U. de Béjaia, Algérie	Examineur

Année Universitaire : 2020/2021

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université A. Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Laboratoire d'Informatique Médicale (LIMED)

THÈSE

Présentée par

SKLAB Youcef

Pour l'obtention du grade de

DOCTEUR EN SCIENCES

Filière : Informatique

Option : Cloud Computing

Thème

**MÉCANISMES DE FORMATION DE COALITIONS MULTI-AGENTS AVEC
EXTERNALITÉS**

Soutenue le 09/09/2021 devant le jury composé de :

M. SEBAA Abderrazak	MCA	U. de Béjaia, Algérie	Président
M. TARI Abdelkamel	Professeur	U. de Béjaia, Algérie	Rapporteur
M. AKNINE SAMIR	Professeur	U. de Lyon 1, France	Rapporteur
M. AHMED NACER MOHAMED	Professeur	USTHB, Algérie	Examineur
M. MARILLEAU NICOLAS	HDR	IRD, France	Examineur
M. KHANOUCHE MOHAMED ESSAID	MCA	U. de Béjaia, Algérie	Examineur

Année Universitaire : 2020/2021

RÉSUMÉ

Les Systèmes Multi-Agents (SMA) permettent la distribution de l'expertise sur un groupe d'agents. Toutefois, cela fait surgir plusieurs problèmes, notamment celui de la coordination décentralisée de l'activité collective. Pour obtenir une coopération efficace, une telle coordination impose la mise en œuvre de protocoles d'interaction sophistiqués notamment par le biais de la formation de coalitions entre les agents. Dans cette thèse, nous considérons de multiples agents égoïstes et rationnels, ayant chacun un but qu'il veut atteindre. Les buts peuvent être atteints en exécutant un plan comprenant un ensemble de tâches interdépendantes. Certaines tâches présentent des dépendances temporelles et nécessitent une exécution séquentielle. Dans un plan, il peut y avoir plusieurs sous-ensembles de tâches (alternatives). Ainsi, l'exécution des tâches est plus bénéfique lorsqu'elle est faite collectivement et non pas individuellement. Dans le cadre de ce travail, nous étudions le problème de la formation de coalitions avec des externalités dynamiques et changeantes dans deux contextes : distribué et centralisé. Nous décrivons le problème dans les deux contextes et introduisons deux algorithmes de résolutions permettant aux agents de converger vers une solution et de réduire les coûts de la réalisation de leurs buts.

Mots Clés : Formation de coalitions, externalités dynamiques, dépendances entre tâches, négociation multi-agents.

ABSTRACT

Multi-Agent Systems (MAS) allow the distribution of expertise over a group of agents. However, this raises several problems, in particular that of decentralized coordination of collective activity. To achieve effective cooperation, such coordination requires the implementation of sophisticated interaction protocols, notably through multi-agents coalition formation. In this thesis, we consider multiple selfish and rational agents, each with a goal that it want to achieve. The goals can be achieved by executing a plan comprising a set of interdependent tasks. Some tasks are time-dependent and require sequential execution. Within a plan, there may be several subsets of tasks (alternatives). Thus, task execution is most beneficial when done collectively rather than individually. In this work, we study the problem of coalition formation with dynamically changing externalities in two contexts : distributed and centralized. We describe the problem in both contexts and introduce two resolution algorithms that allow agents to converge towards a solution and reduce the costs of achieving their goals.

Keywords : Coalition formation, Coalitions externalities , Dynamically changing externalities, Task dependencies, Multi-agents negotiation.

ملخص

تهدف الأنظمة متعددة العوامل أو الذكاء الاصطناعي الموزع إلى سد الفجوات في النهج الكلاسيكي للذكاء الاصطناعي من خلال اقتراح توزيع الخبرة على مجموعة من الوكلاء . ومع ذلك ، تنشأ العديد من المشاكل ، ولا سيما تلك المتعلقة بالتنسيق اللامركزي للنشاط الجماعي . للحصول على تعاون فعال ، يتطلب هذا التنسيق تنفيذ بروتوكولات تفاعل معقدة لا سيما من خلال تشكيل تحالفات بين وكلاء . في هذه الأطروحة ، نعتبر عوامل متعددة أنانية وعقلانية ، لكل منها ، هدف يريد تحقيقه . يمكن تحقيق الأهداف من خلال تنفيذ خطة تتضمن مجموعة من المهام المترابطة . بعض المهام لها تبعيات زمنية وتتطلب التنفيذ المتسلسل . في الخطة ، يمكن أن يكون هناك عدة مجموعات فرعية من المهام البديل التي تحقق الهدف . وبالتالي ، فإن تنفيذ المهام يكون أكثر فائدة عندما يتم بشكل جماعي وليس بشكل فردي . في هذا العمل ، ندرس مشكلة تشكيل تحالفات مع عوامل خارجية ديناميكية ومتغيرة في سياقين مختلفين : موزعة ومركزية . نحن نصف المشكلة في كلا السياقين ونقدم خوارزميتي حل مما يسمح للوكلاء بالتوصل إلى حل مشترك وبالتالي تقليل تكاليف تحقيق أهدافهم

الكلمات المفتاحية : بناء التحالفات ، العوامل الخارجية الديناميكية ، التبعيات بين المهام ، التفاوض متعدد الوكلاء

Table des matières

Table des matières	i
Liste des figures	v
Liste des tableaux	vii
Liste des Algorithmes	viii
Introduction Générale	1

I La coordination par formation de coalitions multi-agents 10

1 La coordination dans les systèmes multi-agents	11
1.1 Introduction	11
1.2 Les systèmes multi-agents	12
1.2.1 Interaction multi-agents	14
1.2.2 Agent coopératif vs agent compétitif	15
1.3 La coordination dans les systèmes multi-agents	16
1.3.1 Concepts et définitions	17
1.3.2 Approches de coordination d'agents	19
1.4 Conclusion du chapitre	28
2 La formation de coalitions multi-agents	29
2.1 Introduction	29
2.2 Concepts et définitions	30
2.3 Formation de coalitions avec des agents coopératifs	35
2.3.1 Jeux coalitionnels	35
2.3.2 Calcul des valeurs des coalitions	36
2.3.3 Recherche de la structure de coalitions optimale	37
2.3.4 Allocation de tâches	42
2.4 Formation de coalitions avec des agents compétitifs	44
2.4.1 Jeux non coopératifs	44
2.4.2 Jeux à utilité transférable	45
2.4.3 Jeux à utilité non transférable	48
2.5 Conclusion du chapitre	50

II	CFDEP : un problème de formation de coalitions avec externalités dynamiques (Analyse et solutions)	51
3	État de l'art sur la formation de coalitions avec externalités	52
3.1	Introduction	52
3.2	Externalités de coalitions	53
3.2.1	Concepts et définitions	54
3.2.2	Les externalités dans CFDEP	55
3.3	Formation de coalitions avec externalités dynamiques : cas d'étude .	58
3.3.1	Contexte distribué : MATP	59
3.3.2	Contexte centralisé : CMATP	64
3.4	CFDEP et les approches de formation de coalitions	66
3.4.1	Formation de coalitions avec externalités	66
3.4.2	Formation de coalitions avec chevauchements	72
3.4.3	Formation de coalitions à base de graphes	73
3.4.4	Formation de coalitions multi-robots	74
3.5	CFDEP et quelques problèmes multi-agents	76
3.5.1	Problème de négociation multi-agents	76
3.5.2	Problème de planification multi-agents	78
3.5.3	Problèmes liés aux déplacements d'individus	79
3.6	Domaines d'application de CFDEP	80
3.7	Conclusion du chapitre	81
4	Une approche distribuée pour la résolution de CFDEP	82
4.1	Introduction	82
4.2	CFDEP dans le contexte distribué	83
4.3	CFDEA : un algorithme distribué de formation de coalitions	87
4.3.1	Génération des structures de coalitions	91
4.3.2	Représentation de l'espace de recherche	93
4.3.3	Protocole de négociations multilatérales	94
4.3.4	Ordonnancement et prise de décisions	100
4.3.5	Raffinement individuel de la solution	112
4.4	Scénario illustratif	115
4.4.1	Paramètres	115
4.4.2	Phase d'initialisation	116
4.4.3	Phase des négociations	117
4.4.4	La solution	123
4.5	Conclusion du chapitre	126
5	Évaluation expérimentale de l'algorithme CFDEA	127
5.1	Introduction	127

5.2	Approche expérimentale	128
5.2.1	Paramètres	129
5.2.2	Générateur aléatoire de plans	131
5.2.3	Algorithme de la meilleure alternative	131
5.2.4	Mesure de la performance	132
5.3	Étude de la dynamique de formation de coalitions avec CFDEA . .	133
5.3.1	Cycles des négociations	133
5.3.2	Effet de l’heuristique $\mathcal{IS\&C}$ sur le comportement des agents	134
5.4	Performances de CFDEA	139
5.5	Conclusion du chapitre	142
6	Une approche centralisée pour la résolution de CFDEP	144
6.1	Introduction	144
6.2	CFDEP dans le contexte centralisé	146
6.3	PSD : un algorithme centralisé de formation de coalitions	149
6.3.1	Partitionnement	152
6.3.2	Réduction de l’espace de recherche	153
6.3.3	Parcours de l’espace de recherche	155
6.3.4	Algorithme PSD	158
6.4	Scénario illustratif	162
6.5	Évaluation expérimentale	173
6.5.1	Paramètres	175
6.5.1.1	Poids additif des agents	176
6.5.1.2	Distance de dépendances	176
6.5.1.3	Performance d’itérations	176
6.5.2	Gain de temps	176
6.5.3	Résultats des expérimentations	177
6.6	Conclusion du chapitre	184
	Conclusion et perspectives	186
	Bibliographie	198

LISTE DES FIGURES

1.1	Interaction directe d'agents.	15
1.2	Illustration d'un problème de coordination d'agents.	17
1.3	Le problème des blocs.	19
1.4	Le protocole Contract Net.	22
2.1	Exemple de formation de coalitions.	30
2.2	Représentation de l'espace de recherche pour un problème avec 4 agents.	38
2.3	Le graphe des <i>Integer-Partitions</i>	40
3.1	Des exemples d'externalités.	55
3.2	Externalités vers l'extérieur.	56
3.3	Externalités vers l'intérieur.	57
3.4	Externalités d'existence.	57
3.5	La carte des chemins possibles de chaque individu.	60
3.6	Les possibilités de déplacements des individus.	65
3.7	Représentation graphique des intentions des agents.	68
3.8	Modèle BPMN de tâches.	71
4.1	Les plans des agents.	83
4.2	Les étapes des négociations.	90
4.3	L'espace de recherche de l'agent α_1	93
4.4	Illustration graphique du processus des négociations.	120
4.5	Les structures de coalitions retenues.	124
4.6	Les dépendances entre les coalitions.	125
5.1	Le nombre de cycles suivant les valeurs de ϕ^{limit}	134

5.2	Le comportement des agents avec l'heuristique $\mathcal{IS\&C}$	135
5.3	Le taux de contribution des paramètres $\bar{\epsilon}^\epsilon, \bar{\omega}^\omega, \bar{\gamma}^\gamma, \bar{\mu}^\mu, \bar{\rho}^\rho$ dans \mathcal{V}_α	138
5.4	Efficacité du système suivant le paramètre de valorisation des coûts.	140
5.5	Efficacité du système suivant le nombre d'agents négociateurs et le nombre des alternatives par agent.	140
5.6	Le taux de succès suivant le nombre d'agents et le nombre de leurs alternatives.	141
5.7	La déviation sur les valeurs d'efficacité des agents, suivant leur nombre et le nombre des alternatives par agent.	141
5.8	Les performances du système avec 10 agents suivant les valeurs de ψ	142
5.9	Les performances du système avec 20 agents suivant les valeurs de ψ	142
6.1	Les structures de coalitions pour 4 agents.	147
6.2	L'espace de recherche de l'agent \mathbf{a}_1	148
6.3	Évolution de \mathbf{v}^{\max} et \mathbf{v}^{\min} pendant le parcours de \mathcal{S}	156
6.4	La distance de dépendances (Δ_{\succ}) suivant le nombre d'agents.	178
6.5	La distance de dépendances (Δ_{\succ}) suivant le nombre de tâches.	178
6.6	Le poids additif des agents (\mathcal{W}_A) suivant le nombre d'agents.	179
6.7	Le poids additif des agents (\mathcal{W}_A) suivant le nombre de tâches.	179
6.8	Évolution du gain de temps (Δ_{temps}) suivant la distance de dépendances (Δ_{\succ}).	180
6.9	Évolution du gain de temps (Δ_{temps}) suivant l'indicateur performance d'itérations (I).	181
6.10	Évolution du gain de temps (Δ_{temps}) suivant le poids additif des agents (\mathcal{W}_A).	181
6.11	Évolution de l'indicateur performance d'itérations (I) suivant l'indicateur poids additif des agents (\mathcal{W}_A).	182
6.12	Évolution de l'indicateur performance d'itérations (I) suivant la distance de dépendances (Δ_{\succ}).	183
6.13	Évolution de la distance de dépendances (Δ_{\succ}) suivant l'indicateur poids additif des agents (\mathcal{W}_A).	184

LISTE DES TABLEAUX

3.1	Les chemins possibles pour chaque individu.	61
3.2	Les coûts des segments des chemins.	63
3.3	Les chemins individuels des agents.	65
3.4	Positionnement de CFDEP.	77
4.1	Les alternatives individuelles des 4 individus du problème MATP.	85
4.2	Les tâches communes et les ensembles communs des agents.	87
4.3	Les structures de coalitions possibles (dans leur état initial).	117
4.4	Les coûts des alternatives.	118
4.5	Les valeurs de \mathcal{P} , $\bar{\mathcal{P}}$ et \mathcal{V} suivant r pour \mathbf{a}_1 et \mathbf{a}_2	121
4.6	Les valeurs de \mathcal{P} , $\bar{\mathcal{P}}$ et \mathcal{V} suivant r pour \mathbf{a}_3 et \mathbf{a}_4	122
4.7	L'utilité finale des agents.	125
5.1	Les paramètres du générateur aléatoire de plans.	131
6.1	Les dépendances entre les tâches.	149
6.2	L'ensemble des coalitions possibles pour les tâches dans \mathcal{T}	152
6.3	Interprétation des partitions possibles pour la tâche t_3	153
6.4	Les partitions possibles pour les tâches dans \mathbf{v}	154
6.5	L'ensemble des coalitions possibles pour les tâches dans \mathbf{v}	155
6.6	Les ensembles TeT pour les tâches dans \mathbf{v}	157
6.7	Les ensembles des tâches des agents.	162
6.8	Les sous-ensembles de tâches (alternatives) permettant aux agents d'atteindre leurs buts.	163
6.9	Les <i>Vecteurs d'agents</i> des tâches dans \mathcal{T}	163
6.10	L'ensemble réduit des partitions possibles pour les tâches dans \mathbf{v}	164
6.11	L'ensemble des partitions à l'itération 2.	166

6.12	L'ensemble des partitions à l'itération 6.	168
6.13	Évolution des ensembles des partitions à travers les itérations de l'algorithme PSD (itérations de 1 à 13).	169
6.14	Évolution des ensembles des partitions à travers les itérations de l'algorithme PSD (itérations de 14 à 26).	170
6.15	Les structures de coalitions parcourues suivant l'évolution des vecteurs \mathbf{v}^{\max} et \mathbf{v}^{\min}	172
6.16	Le nombre d'itérations pour parcourir l'espace de recherche \mathcal{S} dans les deux cas : parcours exhaustif et parcours par PSD avec 3 agents ayant le même plan, suivant le nombre de tâches et des dépendances.	174

LISTE DES ALGORITHMES

1	CFDEA(π_i)	90
2	RunStep(nextStep, Q_i, r)	95
3	DecisionMaking($cs_i, Type$)	102
4	IterativeSearchConverge($\mathcal{P}_i, \bar{\mathcal{P}}_i, \Delta$)	110
5	Scheduler(Q_i, r)	113
6	PSD()	159
7	UpdateVectors()	161

INTRODUCTION GÉNÉRALE

Les systèmes multi-agents (SMA) se définissent comme un groupe d'agents autonomes possédant chacun un but (individuel ou collectif) dont la réalisation est conditionnée par l'exécution de certaines tâches (actions). Les agents évoluent dans un espace commun appelé environnement. Pour accomplir leurs buts, ils sont amenés à coopérer selon des modes d'interactions parfois complexes afin de surpasser leurs conflits. De ces interactions émergent des structures organisées qui, en retour, contraignent et influencent leurs comportements. Par exemple, les agents peuvent partager leurs ressources et les mettre au service de ceux qui en ont besoin ou agir collectivement pour réaliser une tâche particulière. D'une manière générale, il s'agit pour chacun d'eux de construire une solution seule ou par coopération avec les autres en tenant compte des spécificités et contraintes de chacun, ainsi que celles du problème traité. Le processus de la recherche de la solution fait souvent apparaître des dépendances entre agents. Il y a ainsi une nécessité impérieuse d'une coordination explicite et efficace pour permettre aux agents de mener à bien leurs actions. Il est donc important de les doter de mécanismes pour garantir une coordination contrôlée pour éviter des comportements chaotiques. Cela passe par un modèle de coordination qui doit leur permettre d'évoluer au sein du même environ-

nement en complète synergie pour pouvoir gérer chacune des situations éventuelles engendrées par la résolution d'un problème par des agents.

Contexte et motivations

La coordination est définie comme un processus cognitif qui implique l'agencement et la répartition calculés des actions d'un système en vue d'une meilleure rentabilité de ses composants et de l'atteinte d'un objectif déterminé. Dans les SMAs, les agents sont autonomes dans leurs décisions. La coordination doit être alors, assurée de façon complètement distribuée au travers de mécanismes tels que les négociations automatiques, les formations de coalitions, la planification collective, etc. Sans coordination, les agents peuvent rapidement adopter un comportement chaotique dans le système. Notamment dans des situations où ceux-ci ne s'intéressent qu'à leurs propres intérêts. Par ailleurs, les décisions égoïstes seront préjudiciables au système si les agents ne tiennent pas compte des autres dans leur environnement.

Le problème de la coordination décentralisée de l'activité collective est un prolongement de la distribution du contrôle d'un système entre différents agents. Sa résolution peut s'avérer très difficile. Pour le surmonter et obtenir une coopération effective, les agents doivent pouvoir confronter leurs connaissances et capacités mutuelles, interagir pour résoudre collectivement leurs conflits et partager efficacement les ressources dans leur environnement commun. Par conséquent, la coordination impose la mise en œuvre de protocoles d'interaction et de mécanismes sophistiqués notamment par le biais de la formation de coalitions.

De nombreuses études se sont concentrées particulièrement sur la réalisation de tâches par la formation de coalitions [77, 73, 78, 98, 64, 15, 17, 8, 36]. Dans ce travail de recherche, nous considérons la formation de coalitions comme un mécanisme de coordination qui permet notamment aux agents compétitifs de former des groupes d'intérêt afin d'accomplir un ensemble de tâches avec plus d'efficacité. L'objectif est donc de proposer des mécanismes, des stratégies comportementales et des protocoles que les agents pourraient adopter pour assurer une certaine forme

de coordination et de coopération afin d'atteindre leurs buts individuels.

Plus précisément, nous nous intéressons à un problème de formation de coalitions avec un ensemble d'agents ayant chacun un but à atteindre suivant un plan individuel composé de plusieurs alternatives. Les agents sont compétitifs et rationnels. En effet, ils possèdent des informations privées qui concernent leurs préférences ou compétences qu'ils ne divulguent pas aux autres et leurs actions visent à accomplir des tâches qui leur permettent d'atteindre leurs buts et de maximiser leurs gains. Par ailleurs, chaque alternative est formée d'une succession de tâches (*cf.* Fig. 3.6) qui fait émerger un ordre de précedence partiel entre elles. On dit alors qu'il existe des dépendances entre les tâches de chaque alternative.

Les agents évoluent de façon autonome dans le système et sont guidés par leurs buts, qui dépendent du cadre applicatif. La mise en commun de leurs compétences permet d'atteindre en groupe, les buts individuels requis dans les plans de chacun. Lorsqu'un agent veut accomplir une tâche, il n'est pas nécessairement incapable de la réaliser seul. Néanmoins, l'accomplir en groupe peut lui être plus bénéfique. En effet, il est parfois contraint, sous la volonté de minimiser ses coûts, de recourir à une réalisation collective de ses tâches avec de moindres coûts.

Le défi qu'introduit cette situation est que chaque agent pourrait devoir former une série de coalitions à des instants différents dans le temps. La formation séquentielle de coalitions peut conduire dans l'ensemble à une formation de coalitions moins avantageuses et à une exécution sous-optimale des tâches, en particulier en raison des dépendances qui émergent entre les coalitions. Ainsi, après la formation d'une coalition à l'instant t , une autre coalition bénéfique pourrait ne plus être possible à l'instant $t + 1$ en raison des dépendances conflictuelles entre les tâches restantes. Par conséquent, les dépendances entre les tâches font émerger des dépendances entre les coalitions. En effet, comme les agents disposent de plusieurs alternatives qui peuvent contenir des tâches en commun, l'ensemble des dépendances d'une coalition dépend des autres coalitions. Alors, à chaque instant t l'ensemble des dépendances d'une coalition donnée d'un agent, sera déterminé en fonction des autres coalitions des autres agents du système. On dit alors que les dépendances changent dynamiquement.

Ces dépendances inter-coalitions, communément appelées externalités (en Anglais : *externalities*) de coalitions, sont présentes dans de nombreuses applications multi-agents du monde réel [60], et imposent un défi qui n'est pas complètement examiné dans la littérature. Pour donner un exemple illustratif, considérons un problème de covoiturage avec des individus qui souhaitent se rendre d'une ville à une autre en partageant leurs moyens de transport afin de réduire leurs coûts, etc. Chaque individu formule à son agent, chargé de lui trouver une solution, ses buts en émettant des requêtes sous la forme : «Je souhaite effectuer le trajet Paris - Marseille». L'individu exprime à son agent ses contraintes (de budget, par exemple) et ses préférences comme «Je veux partager la voiture avec des jeunes de 25 à 35 ans». Pour résoudre ce problème, les agents doivent trouver un schéma de partage (coalitions) des moyens de transport qui respecte les contraintes et les préférences des individus. Étant donné que ces agents peuvent être conçus par des concepteurs distincts et représenter des utilisateurs différents, ils doivent négocier la formation de leurs coalitions. Dans ce contexte applicatif, une coalition sera formée par un ensemble d'agents qui représente les individus qui partagent les voitures. Ces agents ont pour objectif de réduire le prix unitaire d'une place.

De nombreuses approches de formation de coalitions ont été proposées pour déterminer quelles sont les coalitions à former [76, 69, 77, 73, 78, 48, 63]. Seules quelques méthodes traitent le cas de l'existence de dépendances entre les tâches [77, 44, 98, 64, 6, 14, 102]. Cependant, aucune de ces approches ne permet une formation simultanée de coalitions multiples, où les coalitions et les dépendances entre les tâches changent dynamiquement, et plusieurs alternatives d'ensembles de tâches et de groupes d'agents peuvent mener à la satisfaction des buts. Dans ce contexte, la qualité d'une coalition dépend à la fois de facteurs endogènes et exogènes. En effet, elle est affectée par les autres coalitions possibles. Or, dans la littérature, les agents commencent généralement par former des coalitions pour les tâches courantes et reportent les tâches qui leur succèdent à des étapes ultérieures. En agissant ainsi, les agents ne tiennent pas compte en amont du processus de formation de coalitions des effets que peut avoir chaque coalition sur le reste des potentielles coalitions suivantes. Par conséquent, après avoir formé une coalition

à l'instant t , ils peuvent rater celles qui sont plus bénéfiques à l'instant $t+1$. Pour surmonter ce défi, nous introduisons de nouveaux mécanismes de formation de coalitions dans lesquels la planification et la négociation de l'ensemble des coalitions possibles se font en tenant compte, en amont, des externalités dynamiques. L'exécution des tâches n'aura lieu qu'une fois les coalitions sont établies. Nous désignons cette approche par *formation de coalitions avec des externalités dynamiques*.

Contributions

À la lumière des insuffisances identifiées ci-dessus, nous nous sommes intéressés au problème de la formation de ces coalitions et nous apportons plusieurs contributions. Premièrement, nous introduisons un nouveau problème de formation de coalitions, à savoir, le *Problème de Formation de Coalitions avec des Externalités Dynamiques* (CFDEP — *Coalition Formation with Dynamic Externalities Problem*). Ainsi, nous introduisons notamment une définition générale du problème avec des agents disposant de plusieurs alternatives pour atteindre leurs buts. Nous étudions le problème et discutons les défis qu'il présente.

Deuxièmement, pour traiter le problème CFDEP, nous nous sommes concentrés sur deux contextes différents :

- Un contexte distribué.
- Un contexte centralisé.

Nous proposons dans cette thèse l'étude du problème de la formation de coalitions avec des externalités dynamiques suivant ces deux contextes, dans lesquels évoluent des agents compétitifs, mais dont la coopération (dans le cadre des coalitions) peut permettre à chacun des agents d'obtenir une meilleure satisfaction que s'il agissait individuellement. Le problème est alors la recherche de ces coalitions et l'obtention d'accords de la part des agents membres. Nous soulignons le caractère central des dépendances des tâches et leur effet sur la qualité de la solution. Nous montrons en outre que la combinaison d'agents compétitifs, de buts atteignables par de multiples alternatives et de tâches interdépendantes peut soulever des questions telles que le problème des propositions cycliques, le problème de terminaison

de protocole et le problème des interblocages actifs, etc. En effet, notre solution doit surmonter ces défis.

Dans le premier contexte, les agents explorent les alternatives qui s'offrent à eux, sans divulguer leurs évaluations privées, en essayant de trouver une solution qui maximise leur gain individuel. Ainsi, nous introduisons un nouvel algorithme de formation de coalitions multi-agents (CFDEA) — *Coalition Formation with Dynamic Externalities Algorithm* — qui est approprié à la formation de coalitions dans le contexte distribué. CFDEA est basé sur l'idée que dans la description de CFDEP, les agents doivent maintenir une vue globale de leurs alternatives éventuelles (ou un ensemble préféré de celles-ci) tout au long du processus de formation de coalitions afin de négocier et de planifier un ensemble préféré de coalitions. Dans cette perspective, chaque agent vise à atteindre son but et à maximiser ses gains en formant les coalitions prévues. Cependant, la formation d'une coalition affecte le gain attendu des autres coalitions possibles. Les agents doivent naturellement tenir compte de l'évolution des dépendances dans leur solution et le faire en fonction de leurs fonctions d'utilité individuelles, éventuellement non linéaires. CFDEA permet à chaque agent d'exploiter les plans des autres agents et les dépendances entre ses tâches pour générer des structures de coalitions possibles. Pour faciliter cela, nous introduisons un *Protocole de Négociations Multilatérales* (MNP — *Multilateral Negotiations Protocol*) qui permet aux agents de proposer des coalitions, de répondre aux sollicitations et de s'engager dans une coalitions. Le protocole MNP permet donc de cadrer les comportements des agents tout en les laissant libres dans leur prise de décisions. En effet, il nécessite un partage minimaliste des informations privées. Plus précisément, les agents déterminent d'une manière implicite les préférences des autres à l'égard des leurs alternatives.

Au cours des négociations, les agents construisent les messages adéquats à envoyer suivant des stratégies de négociation, et examinent les propositions des autres qui expriment des suggestions de tâches et de coalitions pour les accomplir. En fonction de leur évaluation privée et des préférences de leurs pairs, ils forment leurs propres propositions ou en acceptent les autres. Les agents utilisent des stratégies de prise de décisions et d'ordonnement qui tiennent compte de

l'historique des échanges entre eux pour en déduire un classement basé sur une heuristique : *Iterative Search & Converge (IS&C)*. Avec l'heuristique *IS&C*, les agents tentent de minimiser la distance estimée entre l'alternative souhaitée d'un agent et les coalitions proposées par les autres. Nous avons traité le problème de la communication entre ces derniers par l'envoi direct de messages entre eux. Notre méthode impose, donc, une structure appropriée aux messages et fournit un protocole d'interaction imposant une sélection de réponses autorisées pendant l'échange des messages. Par ailleurs, il est important de noter que chaque agent cherche particulièrement comme solution une structure de coalitions propre à lui, qui peut être différente de celles des autres. Ceci constitue précisément une autre différence majeure avec les travaux existants dans la littérature, où la solution est une structure unique de coalitions qui englobe toutes les coalitions de tous les agents du système.

Notre évaluation expérimentale montre que d'une part, notre algorithme est capable de réduire significativement les coûts des agents, et d'autre part, l'heuristique *IS&C* est efficace pour faire converger les négociations vers une solution qui propose des coalitions à tous les agents, si elles existent.

Dans le deuxième contexte, nous présentons une version centralisée du problème de la formation de coalitions multi-agents dans un contexte à externalités dynamiques et nous présentons notre algorithme de résolution. Notre défi majeur dans ce contexte est d'éviter les calculs redondants. Par exemple, la génération multiple de toute coalition irréalisable en raison des contraintes du problème. De plus, il est aussi essentiel de considérer la façon dont les coalitions réalisables peuvent être combinées pour générer efficacement des structures de coalitions valides. En effet, il serait coûteux de passer par toutes les combinaisons de coalitions possibles et de vérifier si elles contiennent des coalitions conflictuelles.

Alors, nous proposons un algorithme de formation de coalitions (PSD) par une démarche de **P**arcours **S**électif de l'espace de recherche, guidé par les **D**épendances existantes entre les tâches. L'exploration se fait par itération, où à chaque itération l'ensemble des dépendances est mis à jour à chaque fois qu'une tâche est sélectionnée pour une réalisation conjointe entre plusieurs agents. Ainsi, implici-

tement, l'ensemble des tâches restantes qui peuvent être réalisées conjointement est aussi mis à jour en excluant celles qui ne le sont plus à cause des dépendances conflictuelles avec les tâches déjà sélectionnées.

L'algorithme PSD est dit *anytime* est peut retourner une solution à n'importe quel moment de son exécution et la qualité de sa solution s'améliore continuellement jusqu'à sa terminaison. L'évaluation expérimentale de l'algorithme PSD démontre son efficacité à réduire considérablement le nombre d'itérations nécessaires pour parcourir les espaces de recherche des agents.

Plan de la thèse

Le manuscrit est structuré comme suit : dans le chapitre 1, nous introduisons le domaine des systèmes multi-agents. Nous abordons le chapitre par la notion d'agent et nous présentons quelques concepts à base d'agents sous différentes représentations. Nous présentons également quelques usages que l'on fait avec des systèmes multi-agents, ainsi que quelques protocoles d'interaction et de négociation multi-agents.

Le chapitre 2 est consacré à la présentation du problème de la formation de coalitions multi-agents. Nous présentons une formalisation du problème pour les approches coopératives puis nous détaillons les approches compétitives. Nous décrivons quelques travaux fondamentaux, leurs problématiques et leurs solutions. Nous y définissons les concepts essentiels et nous exposons certains résultats.

Dans le chapitre 3, nous restons sur le problème de la formation de coalitions, où nous nous penchons plus précisément sur la formation de coalitions avec des externalités. Nous nous intéressons plus particulièrement aux situations avec des dépendances entre les tâches et nous introduisons un nouveau problème de formation de coalitions dans un contexte dynamique. Nous présentons deux cas d'études dans deux contextes, à savoir, le contexte distribué et le contexte centralisé. Nous montrons notamment que ces dépendances changent dynamiquement en fonction de l'évolution des considérations des agents dans leurs choix de coalitions. Par ailleurs, nous exposons quelques approches de formation de coalitions et nous pré-

sentons les différences avec quelques problèmes connus des systèmes multi-agents qui ont des points d'intersection avec CFDEP, en mettant en lumière les différences et particularités.

Dans le chapitre 4, nous proposons, une approche de formation de coalitions avec externalités dynamiques dans le contexte distribué. Nous introduisons quelques concepts fondamentaux nécessaires à la compréhension et à la résolution du problème et nous détaillons notre approche de résolution, à travers ces concepts.

Dans le chapitre 5, nous présentons la méthode poursuivie pour l'évaluation expérimentale et nous introduisons les métriques pour analyser et interpréter les résultats obtenus.

Dans le chapitre 6, nous proposons, une approche de formation de coalitions avec externalités dynamiques dans le contexte centralisé. Comme dans le contexte distribué, nous redéfinissons quelques concepts fondamentaux nécessaires à la compréhension et à la résolution du problème dans sa version centralisée et nous détaillons notre approche de résolution. Nous concluons le chapitre en présentant les résultats obtenus de l'évaluation expérimentale.

Enfin, dans le chapitre 7, nous concluons sur les travaux que nous avons menés et présentons des perspectives d'études.

Première partie

La coordination par formation de coalitions multi-agents

CHAPITRE 1

LA COORDINATION DANS LES SYSTÈMES MULTI-AGENTS

Les travaux de recherche sur les SMA ont permis d'obtenir des résultats remarquables tant du point de vue théorique que du point de vue pratique. Néanmoins, la coordination des agents demeure toujours une préoccupation d'actualité. L'enjeu est de permettre aux agents de décider de leurs propres actions sans conflits. L'objectif de ce chapitre n'est pas de présenter de manière détaillée les SMA et la coordination d'agents, mais de présenter les concepts nécessaires à la compréhension de nos travaux. Nous y exposons donc ce que nous entendons par les SMA et la coordination d'agents dans le contexte de nos travaux de recherche.

1.1 Introduction

L'intelligence artificielle distribuée (IAD) se définit comme une branche de l'intelligence artificielle (IA), qui s'intéresse à la modélisation du comportement intelligent d'un ensemble d'agents opérants collectivement, de façon décentralisée pour aboutir à un objectif global. Contrairement à l'approche classique de l'IA qui centralise le raisonnement et qui modélise le comportement intelligent d'une entité

unique, l'IAD propose la distribution du raisonnement sur un groupe d'agents et s'intéresse à des comportements intelligents qui sont le produit de la coordination ou la coopération de l'activité de plusieurs entités (agents).

1.2 Les systèmes multi-agents

Les systèmes multi-agents (SMA) font référence aux approches distribuées de l'IA fondées sur la définition de plusieurs entités appelées «agents». Ils ont la double vocation de fournir, d'une part, des principes pour le développement de systèmes complexes impliquant plusieurs agents et d'autre part, des mécanismes pour la coordination d'agents ayant des comportements indépendants les uns des autres [89].

Les SMA sont utilisés dans une variété de domaines. Dans l'industrie, ils assurent des fonctions d'automatisation des processus et de la production, la logistique et les robots coopératifs ou la maison intelligente. Dans le domaine de traitement de l'information, ils sont utilisés comme des assistants personnels ou pour la recherche d'information et la gestion du workflow. Dans le domaine de la santé, ils sont utilisés principalement pour la supervision des malades et les systèmes de support. Ils sont aussi utilisés dans le domaine de la communication pour effectuer la gestion de réseaux, le commerce électronique ou le calcul mobile [31]. Les domaines d'applications des SMA sont donc particulièrement riches. Nous en citons deux catégories fondamentales, à savoir : la résolution de problèmes et la simulation.

La résolution de problèmes : Consiste à utiliser des agents en groupe pour résoudre un problème qui requiert des efforts collectifs, soit, pour qu'un agent puisse accomplir ses tâches, soit, pour qu'il puisse les accomplir d'une meilleure façon (Plus de rapidité, plus de précision ...). Les approches de résolution sont d'ordinaire des approches coopératives [25].

La simulation : Concerne la modélisation et la reproduction des phénomènes du monde réel afin de comprendre ou d'expliquer leur comportement. La simulation

constitue une branche très active de l'informatique qui consiste à analyser les propriétés des modèles théoriques du monde environnant. La physique, la chimie, la biologie, l'écologie, la géographie et les sciences sociales notamment, font un grand usage des simulations pour essayer d'expliquer et de prévoir les phénomènes naturels. Les modèles les plus utilisés sont les équations différentielles, les matrices de transitions, etc. Ces simulations reposent sur la définition d'une relation de cause à effet entre des variables d'entrées et des variables de sorties [31].

Russell [67] présente un agent comme tout ce qui peut percevoir son environnement par des capteurs et agir sur cet environnement moyennant des effecteurs. Un agent est souvent vu comme étant une entité dotée de certaines caractéristiques (buts, actions, connaissances, préférences ... etc.) qui varient d'un domaine à un autre. Il est convenable de caractériser un agent comme une entité avec les caractéristiques suivantes :

- Capable d'agir dans son environnement ;
- Peut communiquer, directement ou indirectement, avec d'autres agents ;
- Animée par un ensemble de tendances (objectifs, satisfaction, etc.)
- Possède des ressources propres à elle et des compétences qu'elle peut offrir sous forme de services ;
- Dotée d'une capacité de perception de son environnement ;
- N'a qu'une connaissance partielle et locale de son environnement ;
- Peut éventuellement, se reproduire ;
- Se distingue par un comportement qui tend à satisfaire ses objectifs, en prenant en compte les ressources et compétences dont elle dispose, en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Un agent est donc défini comme une entité autonome évoluant en interaction avec son environnement, souvent dynamique et imprévisible. Il peut exister seul ou dans un groupe d'agents. Dans un groupe, de nouveaux concepts émergent, comme les notions d'interaction, coopération, collaboration ou coordination d'actions.

1.2.1 Interaction multi-agents

Dans un SMA, les agents peuvent interagir, donc, s'influencer mutuellement sous la forme d'action/réaction. Il en découle une propriété fondamentale : l'interaction multi-agents.

Nous entendons par interaction la mise en relation dynamique de plusieurs agents par un enchaînement d'actions, influant, les uns sur les autres. L'interaction multi-agents est centrale pour l'existence et l'évolution des agents où leur autonomie passe par leurs échanges avec l'environnement et où les échanges entre les agents peuvent mener à des comportements collectifs. Les interactions entre ces derniers prennent la forme d'actions réciproques qui peuvent impacter leurs comportements. Les agents interagissent le long d'une suite d'événements pendant lesquels ils sont d'une certaine manière en contact les uns avec les autres que ce contact soit direct ou qu'il s'effectue par un intermédiaire (un agent ou l'environnement) [31]. C'est donc par les échanges que les agents exercent leur influence les uns sur les autres. Pour qu'une interaction puisse avoir lieu, on suppose que certaines contraintes doivent être vérifiées :

- Les agents doivent être capables d'agir ou de communiquer ;
- Il existe des situations susceptibles de faire rencontrer les agents : collaboration, déplacement collectif d'objets, le partage contrôlé de ressources limitées, etc.
- Il existe un certain «jeu» dans les relations entre les agents leur permettant à la fois d'être en relation, mais aussi de pouvoir se séparer de cette relation, c'est-à-dire de disposer d'une certaine autonomie.

Deux niveaux d'interactions entre les agents peuvent avoir lieu (*cf.* Fig 1.1) : les interactions directes (agent-agent) et les interactions indirectes (via l'environnement). Au premier lieu, les interactions directes regroupent les formes de communication qui nécessitent la mise en place d'un protocole qui peut prendre la forme d'un langage. C'est ce type d'interaction que l'on retrouve dans les systèmes coopératifs où des véhicules connectés, par exemple, échangent des informations sous forme de messages. Au second lieu, les interactions indirectes sont les formes de communication qui s'appuient le plus souvent sur des modifications de l'en-

vironnement partagé par les agents. Elles sont le fait de la coopération entre les agents.



Fig. 1.1 – Interaction directe d’agents.

1.2.2 Agent coopératif vs agent compétitif

En considérant le comportement des agents au sein d’un SMA, un agent peut être soit coopératif, soit compétitif (égoïste). On définit un agent coopératif comme un agent qui agit dans l’intérêt de son groupe et peut réaliser des tâches selon le besoin du groupe sans espérer un retour particulier. Par conséquent, un groupe d’agents coopératifs est initialement défini avec des buts communs et œuvrent en équipe pour les satisfaire. Le rôle de chacun au sein du groupe est défini en fonction de ses capacités à participer à l’accomplissement de ces buts. Généralement, les agents coopératifs agissent en suivant un agent dirigeant et favorisent le partage de connaissances et d’informations pour l’élaboration d’une conjoncture commune.

Le deuxième type est l’agent compétitif qu’on définit comme un agent égoïste dont les actions sont guidées par ses intérêts propres, plutôt que les intérêts des autres agents. Contrairement aux SMA coopératifs (jeux coalitionnels) ou l’unité de modélisation est le *groupe d’agents*, dans les SMA compétitifs, l’unité de modélisation de base est l’agent comme individu seul. Dans ce type d’agents, à priori, il n’y a pas de buts communs aux agents et ils peuvent éprouver des intérêts divergents, mais cela ne signifie pas pour autant qu’ils ne cherchent qu’à satisfaire leurs propres intérêts. Ce sont des agents négociateurs à cause de leurs divergences ou de l’insuffisance de leurs ressources. Un agent compétitif évolue en parfaite cohérence avec ses préférences individuelles. Il ne cherche que la réalisation de ses propres buts et a priori ne partage pas ses connaissances avec les autres agents. S’il le fait,

c'est qu'il espère en tirer un bénéfice en retour ce qui peut engendrer des effets bienfaisants pour les autres.

En concevant un agent compétitif, le concepteur doit anticiper des situations dans lesquelles cet agent pourrait nécessiter la coopération avec d'autres agents [66]. Ainsi, des structures organisationnelles des agents émergent avec le temps lorsque ceux-ci coopèrent. Il est à noter que la compétition tend à rendre le système moins performant dans sa globalité comparativement à un système coopératif.

La fonction d'utilité d'un agent : La théorie de décision modélise le comportement d'un agent face à des situations de choix entre plusieurs alternatives (choix d'actions, sélection d'objets, évaluation d'une proposition, etc.). L'agent doit être capable d'évaluer ces alternatives et de décider s'il opte pour une alternative en particulier. L'utilité est une mesure de satisfaction, permettant à l'agent d'évaluer numériquement les alternatives qui sont à sa disposition. La fonction d'utilité, notée $u : \mathbb{A} \mapsto \mathbb{R}$ où \mathbb{A} est l'ensemble des alternatives de l'agent et \mathbb{R} est l'ensemble des nombres réels, lui permet d'établir un ordre de préférence sur ses alternatives [35].

1.3 La coordination dans les systèmes multi-agents

De nos jours, les progrès dans les SMA permettent de développer des systèmes complexes. Cependant, les agents sont parfois inefficaces et souvent leurs efforts individuels ne leur suffisent pas pour atteindre leurs objectifs. Cette inefficacité est parfois aussi présente même dans des situations où les agents disposent collectivement de toutes les ressources nécessaires. Le manque de coordination devient alors une des causes majeures de cette inefficacité. Par conséquent, dans les SMA, la coordination des agents est une préoccupation centrale [23]. Elle est étudiée dans diverses disciplines comme les sciences sociales, les sciences politiques, la psychologie sociale, ou la sociologie [37]. Par exemple, les théoriciens de l'organisation s'intéressent à la coordination entre les humains, allant des petits groupes aux grandes organisations formelles [56]. Les économistes s'intéressent à la coordina-

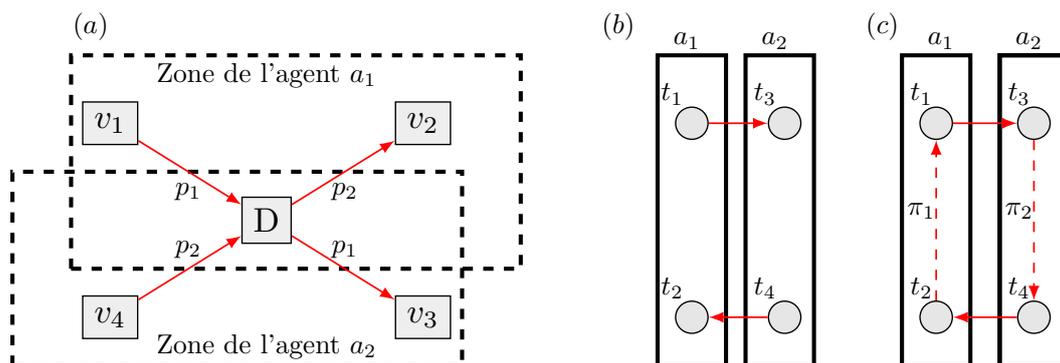


Fig. 1.2 – Illustration d'un problème de coordination d'agents.

tion pour étudier les marchés d'entreprises qui maximisent leurs profits [32].

1.3.1 Concepts et définitions

La coordination dans les SMA représente la capacité de chaque agent au sein d'un groupe d'agents à décider de ses propres actions sans qu'il y ait de conflits pour que tous les agents puissent accomplir leurs activités. Elle peut être vue comme un processus dans lequel les agents s'engagent afin d'assurer une communauté d'agents qui agissent de manière cohérente. La cohérence signifie que leurs actions s'articulent correctement les unes avec les autres sans créer de conflits. En d'autres termes, la cohérence fait référence à la façon dont un système d'agents se comporte en tant qu'unité [37].

Exemple 1.1. Soit deux agents a_1 et a_2 qui doivent livrer deux colis p_1 et p_2 depuis leurs sources v_1 et v_4 en passant par le dépôt D vers leurs destinations v_2 et v_3 (cf. Fig. 1.2 (a)). L'agent a_1 est capable de passer par v_1 , D et v_2 uniquement une seule fois, tandis que a_2 peut passer uniquement par v_3 , D et v_4 une seule fois. Les tâches de transport associées $T = \{t_1, t_2, t_3, t_4\}$ sont interdépendantes et a_1 , a_2 ont chacun des tâches individuelles (cf. Fig. 1.2 (b)). Si l'agent a_1 décide d'établir un plan où t_2 précède t_1 et a_2 établit un plan où t_3 précède t_4 (cf. Fig. 1.2 (c)), ces plans ne peuvent pas être combinés, car le résultat serait un plan global invalide.

Il n'existe pas de démarches universellement acceptées pour permettre aux

agents de se coordonner dans toutes les circonstances. En outre, dans la pratique, il existe plusieurs raisons pour lesquelles plusieurs agents doivent coordonner leurs actions :

Éviter l'anarchie ou le chaos : avec la décentralisation dans les systèmes multi-agents, des situations d'anarchie peuvent facilement émerger. En effet, dans de tels systèmes aucun agent n'a une vision globale sur l'ensemble du système et les agents doivent se coordonner pour pouvoir accomplir leurs tâches de manière cohérente. La coordination est donc nécessaire, car, l'anarchie ou le chaos ne sont pas tolérables dans les systèmes complexes.

Répondre à des contraintes globales : lorsqu'il s'agit de faire fonctionner un groupe d'agents ensemble, il existe généralement des contraintes globales qu'ils doivent vérifier. Par exemple, un groupe d'agents ayant l'objectif de construire une structure peut être amené à travailler en respectant des restrictions budgétaires préétablies. Les agents doivent alors coordonner leurs dépenses pour respecter cette contrainte globale.

Partage de ressources : Les agents peuvent avoir des ressources (moyens, expertises ...) individuelles ou collectives. Pour atteindre leurs objectifs, chacun peut avoir besoin de certaines ressources des autres agents. Alors, ils peuvent se coordonner pour partager leurs ressources et les mettre au profit des uns et des autres au moment opportun et gérer l'usage des ressources collectives.

Dépendances entre les tâches des agents : les tâches que les agents doivent accomplir sont parfois interdépendantes et la réalisation d'une tâche peut nécessiter les résultats d'une autre. À titre d'exemple, considérons deux agents qui veulent chacun empiler un bloc (*cf.* Fig. 1.3) pour avoir une pile dans l'ordre A-B-C. En supposant que chaque agent doit effectuer une action, la façon la plus simple serait que le premier agent procède à l'empilement de B sur C tandis que le deuxième procède à l'empilement de A sur B-C afin d'obtenir la pile A-B-C. De toute évidence, les tâches des agents sont interdépendantes : le second agent doit attendre que le premier ait empilé B sur C avant de pouvoir empiler A sur B. Lorsque de telles interdépendances existent, les activités des agents doivent être coordonnées.

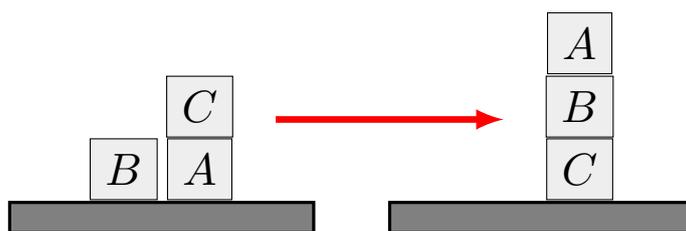


Fig. 1.3 – Le problème des blocs.

Efficacité : Même lorsque les agents peuvent atteindre leurs objectifs individuellement, sans avoir le besoin de coordonner leurs activités, les efforts fournis par les uns et les autres peuvent être utiles pour l'intérêt collectif pour que les agents réalisent leurs objectifs avec plus d'efficacité (réduire les coûts de réalisation, temps de réalisation, la qualité ...).

La coordination au sein des systèmes multi-agents est généralement réalisée selon 5 approches : (i) par structuration organisationnelle (ii) par contrat (iii) par planification (iv) par négociation et (v) par formation de coalitions. Pour assurer la coordination entre les agents, ces derniers peuvent être amenés à communiquer entre eux. Dans de telles situations, les agents doivent souvent faire connaître leurs objectifs, leurs intentions et leurs résultats aux autres agents (partiellement ou totalement). S'ils ne les font pas, il s'agirait alors d'une situation de compétition obligeant les agents à ne pas divulguer les informations qu'ils estiment privées. Cependant, il existe des situations où les agents peuvent parvenir à la coordination sans communication à condition qu'ils possèdent les modèles de comportement des autres [46]. Dans une telle situation, la coordination peut être assurée principalement par l'adoption d'une certaine structuration organisationnelle entre les agents.

1.3.2 Approches de coordination d'agents

Nwana, Lee et Jennings [37] ont classé les approches de coordination en quatre grandes catégories :

1. Par structuration organisationnelle

2. Par contrat
3. Par planification
4. Par négociation

Dans la suite de cette section, nous allons détailler chaque catégorie.

Coordination par structuration organisationnelle

Doter les agents d'une organisation structurelle où chacun a un rôle et des responsabilités prédéfinis est la façon la plus simple pour faire coordonner un groupe d'agents. Une organisation est un schéma de prise de décision et de communication appliqué à un ensemble d'agents qui réalisent un ensemble de tâches. Durfee et al [26] définissent ceci comme les relations à long terme prédéfinies entre les agents. Les organisations hiérarchiques de type maître/esclave ou client/serveur sont des approches de coordination, généralement utilisées pour la répartition des tâches ou des ressources entre des agents esclaves et un agent maître. Cette approche est mise en œuvre de plusieurs façons :

- L'agent maître a la responsabilité de planifier et distribuer des tâches aux esclaves. Les esclaves ont la responsabilité de les réaliser et communiquer leurs résultats à leur maître.
- L'agent maître a la responsabilité de planifier les lectures/écritures des agents dans un tableau noir partagé, selon une stratégie d'ordonnancement [33]. Le tableau est exploité par les agents comme une base de coordination entre eux. Cette approche est généralement utilisée pour la résolution des problèmes distribués. Ici, les tâches sont supposées déjà affectées aux agents[54, 33].

Dans [45], l'opération d'affectation des tâches concerne les tâches dites publiques. L'affectation se fait selon une stratégie dans laquelle on essaye de voir si un but est atteignable depuis un état initial et avec quel coût estimé. Dans cette stratégie les coûts des buts par rapport aux agents sont gardés dans une matrice et l'affectation se fait en fonction de ces coûts selon trois cas :

- All-achievable : chaque but peut être affecté à n'importe quel agent.

- Rest-achievable : l'affectation se fait de manière itérative. On affecte pour un agent les buts qu'il peut potentiellement atteindre puis on les supprime de l'ensemble des buts et on passe à l'agent suivant.
- Best-cost : chaque but est affecté pour un agent qui peut l'atteindre avec le moindre coût.

Cette dernière peut donner des résultats où des agents auront plus de buts à atteindre que d'autres.

L'approche maître/esclave impose beaucoup de contrôle sur les actions de l'esclave. Cependant, un excès d'un tel contrôle atténue tous les avantages de l'IAD - vitesse (due au parallélisme), fiabilité, robustesse, autonomie, etc [54]. Dans le scénario du tableau noir, la communication directe d'agent à agent peut entraîner des goulots d'étranglement et impose une contrainte sur le langage des agents participants [54, 33].

Coordination par contrat

Dans de nombreux domaines, certaines tâches sont intrinsèquement synergiques et doivent être gérées par un seul agent. Il devient donc nécessaire de procéder à une opération de répartition de tâches (ou ressources) entre les agents. Dans une telle situation, une des approches utilisées est le protocole Contract-net [83], qui suppose l'existence d'une structuration sous forme d'un marché avec un seul manager centralisé. Les agents sont supposés avoir des capacités et des ressources différentes. Par conséquent les coûts de réalisation des tâches sont différents d'un agent à un autre. Les agents peuvent assumer un des deux rôles suivants :

- Le rôle de manager : dont la responsabilité est d'une part, de diviser le problème en sous-tâches et de chercher des contractants pour les exécuter, et d'autre part, de superviser le déroulement de la résolution globale du problème.
- Le rôle de contractant : dont la tâche est la réalisation d'une sous-tâche. Notons qu'un contractant peut à son tour décomposer une tâche dont il a la responsabilité de réalisation, en sous-tâches et de jouer le rôle d'un manager pour sous-traiter les sous tâches à d'autres agents contractants.

Le protocole Contract-net (*cf.* Fig. 1.4), proposé par Smith [83], facilite le contrôle distribué de l'exécution de tâches coopératives avec une communication efficace entre les nœuds d'un solveur distribué de problèmes. Il fut l'une des premières solutions au problème d'allocation de tâches dans les systèmes multi-agents.

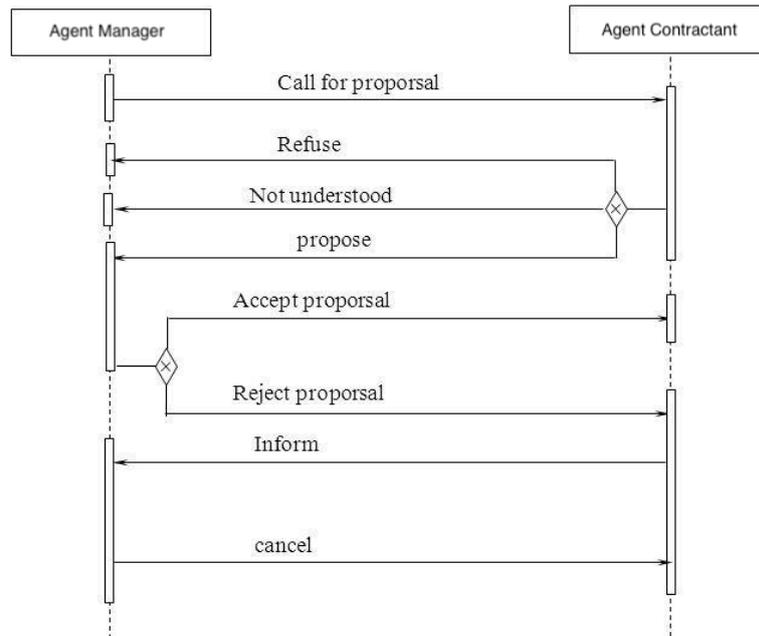


Fig. 1.4 – Le protocole Contract Net.

Dans ce protocole, les agents coordonnent leurs activités grâce à l'établissement de contrats afin d'atteindre des buts spécifiques. Pour qu'un manager arrive à trouver des contractants aux différentes sous-tâches dont il dispose, il suit le processus suivant :

1. Le manager annonce chaque tâche à réaliser aux contractants par diffusion d'un message ;
2. Les contractants évaluent la tâche en fonction de leurs compétences et de leurs engagements. Ceux qui possèdent les ressources nécessaires pour accomplir la tâche envoient au manager des soumissions qui indiquent leurs capacités à réaliser la tâche ;

3. Le manager, en rassemblant toutes les propositions qu'il a reçues, alloue la tâche à l'agent qui a formulé la meilleure proposition ;
4. Le manager et les contractants échangent les informations nécessaires durant l'accomplissement de la tâche. Par exemple, le contractant annoncera au manager le moment où l'exécution de la tâche sera terminée. Dans des cas exceptionnels, un manager peut annuler le contrat et annoncer au contractant qu'il faut abandonner l'exécution de la tâche.

En cas d'acceptation, chaque contractant devient responsable de l'exécution effective de la tâche qui lui a été attribuée. Le manager supervise l'exécution de la tâche et a la responsabilité de traiter les résultats obtenus de cette exécution. Procéder par une approche de coordination par contrat présente des avantages comme :

- Une attribution dynamique des tâches par le biais de l'auto-soumission, ce qui permet d'obtenir de meilleurs accords ;
- Permet un équilibrage de la charge ;
- Un mécanisme fiable pour le contrôle distribué et tolérant aux pannes [54].

Cette approche a été utilisée dans de nombreuses applications [72, 71, 70]. Cependant, le protocole Contract-net est un protocole de communication intensive et nécessite une décomposition appropriée des tâches.

Coordination par planification

Une autre approche de la coordination dans les SMA et de faire les agents œuvrer ensemble pour construire un plan qui guidera leur fonctionnement. Il s'agit d'un problème de recherche d'une séquence d'actions à réaliser de façon à permettre aux agents d'atteindre leurs objectifs en évitant les conflits qui peuvent apparaître lors de la réalisation de leurs tâches.

Un problème classique de planification multi-agents est défini par un quadruplet $\Pi = \langle P, A, I, G \rangle$, avec P : un ensemble de propositions, A : un ensemble d'agents ; I : un état initial et G : l'ensemble des buts. Généralement, le problème est abordé dans une des deux façons suivantes :

- Comment synthétiser les actions globales dans un plan individuel, local à un agent ?
- Comment coordonner les plans des agents dans un environnement partagé ?

Il existe deux grandes approches de résolution : la planification multi-agents centralisée et la planification multi-agents distribuée.

a) Planification multi-agents centralisée : La planification multi-agents centralisée consiste à élaborer un plan en utilisant un agent maître qui connaît tous des autres agents [45]. Il y a généralement un agent de coordination, dont le travail consiste à recevoir les plans partiels ou locaux de tous les autres agents, analyser ces plans afin de les identifier et les modifier en les combinant en un seul plan multi-agents qui est sans conflits [54]. Cammarata et al [13] ont adopté la planification centralisée dans un domaine de simulation de contrôle du trafic aérien. Dans ce démonstrateur, pour deux agents, en cas de conflit, l'un d'entre eux sera sélectionné pour agir comme agent de coordination. L'autre agent transmettra ses plans à l'agent de coordination qui révisera par la suite son propre plan pour éviter la collusion potentielle. Cette approche présente les mêmes inconvénients que la stratégie maître/esclave : le contrôle est central et sa complexité peut croître de façon exponentielle avec le nombre d'agents [45].

b) Planification multi-agents distribuée : La planification multi-agents distribuée se déroule en deux étapes. Dans la première, chacun construit son propre plan et dans la seconde les plans des agents sont, soit fusionnés avant le début de leur exécution, soit un plan de coordination est établi pendant leur exécution pour éviter que les actions d'un agent n'interfèrent négativement avec les objectifs des autres [45]. Il existe deux sous-classes : (i) Planification distribuée pour des plans centralisés et (ii) Planification distribuée pour des plans distribués.

Planification distribuée pour des plans centralisés : L'élaboration d'un plan complexe peut nécessiter le recours à plusieurs techniques de résolution de la planification coopérative, tout comme la recherche d'une solution à un problème complexe [33]. L'idée est de répartir la tâche d'élaboration du plan et de partager

les résultats des sous-tâches pour aboutir à un plan global. Si un des agents ne peut pas exécuter sa sous-tâche de planification à l'aide du plan partiellement construit, il peut trouver d'autres choix. Cette façon d'élaborer un plan exige ainsi systématiquement une véritable décomposition de la tâche d'élaboration du plan qui n'est pas le plus souvent adaptée à tous les domaines.

Planification distribuée pour des plans distribués : Il s'agit de la version la plus difficile de la planification distribuée; la planification et les résultats sont destinés à être distribués [33, 38, 99, 40]. L'idée est de doter chaque agent avec un modèle des plans des autres agents. Les agents communiquent pour construire et actualiser leurs plans individuels et leurs modèles de plans des autres jusqu'à ce que tous les conflits soient supprimés [33]. Un exemple de cette approche est l'approche de planification globale partielle (Partial Global Planning (PGP)) de Durfee [24]. Dans PGP, un agent peut :

1. Représenter ses propres activités prévues ;
2. Communiquer au sujet de ces plans avec d'autres agents ;
3. Modéliser les activités collectives de multiples agents ;
4. Proposer des changements à une ou plusieurs activités globales pour améliorer les performances du groupe ;
5. Modifier ses activités locales planifiées en fonction de l'évolution de la planification [33, 24].

En règle générale, la planification multi-agents distribuée est plus complexe que la planification centralisée. En effet, elle nécessite des moyens informatiques et des communications complexes. De plus, certaines approches de planification comme le PGP sont graduelles et, par conséquent, leur portée d'applicabilité peut être limitée.

Coordination par négociation

Dans la littérature, il existe plusieurs définitions de la négociation. Une des plus succinctes et plus basiques est celle de Bussman & Muller [11] où elle est

définie comme un processus de communication d'un groupe d'agents dans le but de parvenir à un accord mutuellement accepté. Une partie importante des travaux menés sur la coordination relève de la négociation [37].

Ainsi, en s'inspirant des interactions sociales entre les humains, plusieurs recherches dans le domaine des systèmes multi-agents ont proposé des modèles de négociation d'agents basés principalement sur la loi de l'offre et de la demande. Par exemple, les agents ayant des ressources et la capacité de réaliser des tâches avec ces ressources peuvent soumissionner auprès d'autres agents ayant besoin d'un service qu'eux-mêmes ne peuvent pas atteindre. Si les deux côtés arrivent à se mettre d'accord sur un contrat de réalisation des tâches, il y aura alors un contrat entre les deux parties. Les agents sans ressources doivent donc, payer pour sous-traiter leurs tâches.

Dans la réalité, le problème de la négociation peut s'avérer plus complexe. Notamment dans des situations où les agents ne sont pas bien intentionnés. Cela soulève de nombreux problèmes dans les SMA, ce qui se voit d'ailleurs, dans le nombre important de recherches menées sur le sujet. Sycara [90] souligne que pour négocier efficacement, les agents doivent raisonner sur les croyances, les désirs et les intentions des autres agents, ce qui a conduit à l'élaboration d'approches permettant aux agents : (i) la représentation et le maintien des modèles de croyance ; (ii) de raisonner sur les croyances des autres agents ; et (iii) d'influencer les intentions et les croyances des autres agents. Ces dernières passent nécessairement par l'utilisation de batterie d'approches d'IA et des mathématiques, y compris la logique, le raisonnement, la révision des croyances, la planification multiagent ... etc.

Les concepts clé dans la négociation sont les suivants : (i) la fonction d'utilité, définie comme la différence entre la valeur attendue d'un objectif et le prix nécessaire pour l'atteindre. (ii) L'espace de la recherche de la solution, défini comme l'ensemble des alternatives offertes à l'agent pour atteindre son objectif. (iii) Les stratégies et les protocoles, qui définissent les règles qui régissent la négociation, y compris comment et quand elle prend fin (p. ex. par accord ou sans accord). (iv) Un langage de communication ; (v) Un processus de décision par lequel un agent

détermine sa position.

Notons que la négociation en tant que technique est déjà utilisée dans de nombreuses approches. Le processus de négociation implique un processus interactif d'offres et de contre-offres dans lequel chaque agent choisit une transaction qui maximise sa valeur utilitaire attendue. Il y a une hypothèse implicite selon laquelle chacun des agents dans la négociation est un maximiseur d'utilité attendue. À chaque étape de la négociation, un agent évalue l'offre de l'autre en fonction de sa propre stratégie de négociation. Par ailleurs, nous distinguons trois formes de négociations multi-agents :

- Négociation un-à-un : C'est le cas ordinaire où chaque agent négocie avec un seul adversaire.
- Négociation un-à-plusieurs : Un agent négocie avec un groupe d'agents.
- Négociation plusieurs-à-plusieurs : C'est le cas le plus complexe où chaque groupe d'agents peut mener des négociations avec plusieurs agents.

À l'issue des négociations, un accord est défini comme un plan conjoint entre les agents qui répondraient à tous leurs objectifs. À chaque action est associée une valeur d'utilité et chacun des agents veut maximiser son utilité. Les agents discutent d'un ensemble de négociations qui est l'ensemble de tous les accords possibles qui ont une utilité positive pour chacun d'eux. Il y a une hypothèse implicite selon laquelle chaque agent veut maximiser son utilité.

Parce que les agents ne sont pas tenus d'être honnêtes en tout temps, le processus de négociation peut être considéré comme un processus en deux étapes [90] :

- La négociation proprement dite
- L'exécution de l'entente conjointe

Vu que l'interaction humaine nécessite une certaine forme de négociation, beaucoup de cas s'inspirent des stratégies de négociation humaine. L'un des documents les plus utiles sur la négociation est celui de Bussmann et Muller's qui proposent un framework de négociation pour des agents coopératifs [14]. S'inspirant des théories socio-psychologiques de la négociation, ils élaborent un modèle de négociation cyclique qui est à la fois général et simple. La stratégie générale est que la négociation commence avec un ou plusieurs agents qui font une proposition. D'autres agents évaluent la proposition et dressent la liste des préférences qui ont été insatisfaites

par la proposition. Ainsi, chacun actualise ses connaissances.

Coordination par formation de coalitions

La formation de coalitions demeure une question fondamentale dans la recherche sur les systèmes multi-agents. Les agents coopérants sont souvent plus efficaces que les agents qui agissent individuellement. La formation de coalitions offre des techniques permettant de tirer avantage des possibilités de coopération entre les agents en modélisant explicitement leur capacité à réaliser des actions conjointement. Nous détaillons davantage les concepts de la formation de coalitions ainsi que les principaux travaux existant dans le chapitre qui suit.

1.4 Conclusion du chapitre

Dans ce chapitre, nous avons introduit quelques notions préliminaires sur les fondements des systèmes multi-agents et de la coordination d'agents. Les travaux de recherche sur les SMA ont permis d'avoir des résultats remarquables tant du point de vue théorique que du point de vue pratique. Toutefois, les domaines de recherche sont très variés et les notions fondamentales peuvent être différentes d'un domaine à l'autre. L'objectif de ce chapitre n'est pas de présenter de manière détaillée les SMA et la coordination d'agents, mais de présenter les concepts nécessaires à la compréhension de nos travaux.

Nos travaux de thèse portent sur des agents compétitifs évoluant dans des contextes particuliers où, il leur est nécessaire de former des coalitions. Nos agents disposent de fonctions d'utilité afin de guider leurs choix. Le chapitre suivant présente les concepts préliminaires du problème de formation de coalitions, nécessaires pour comprendre les problèmes de formation de coalitions traités dans le cadre de nos travaux.

CHAPITRE 2

LA FORMATION DE COALITIONS MULTI-AGENTS

La formation de coalitions permet à un groupe d'agents de réaliser des tâches collectivement et de dépasser leurs limites individuelles. Plusieurs approches de formation de coalitions ont été proposées. Néanmoins, aucune ne traite le problème de la formation de coalitions avec des externalités dynamiques comme elles sont décrites dans nos travaux. En effet, les mécanismes proposés ne considèrent pas le cas d'existence de dépendances entre les tâches avec plusieurs alternatives possibles pour les agents, c.-à-d. l'ensemble des tâches est considéré commun entre tous les agents. De ce fait, l'aspect dynamique et changeant des dépendances n'a pas été abordé auparavant.

2.1 Introduction

LE problème de la formation de coalitions a suscité l'intérêt des chercheurs dans plusieurs disciplines. Il a été abordé sous différentes formulations. Par conséquent, cela a abouti à une diversité de formulations, analyses et approches de résolution. Cependant, ces recherches tendent toutes vers l'idée que la formation de coalitions est un moyen incitatif pour la coopération pour permettre à des

individus de se regrouper temporairement afin d'atteindre conjointement leurs objectifs. Dans ce chapitre, nous présentons les différentes approches du problème de la formation de coalitions relatives à des agents coopératifs puis celles relatives aux agents compétitifs. Nous y présentons les travaux fondamentaux et les notions clés à la fois nécessaires à la compréhension des modèles existants et à la formulation de nos travaux.

2.2 Concepts et définitions

Dans la littérature, la formation de coalitions multi-agents est définie comme un moyen incitatif pour la coopération et la coordination entre les agents. Elle permet de déterminer les groupes d'agents qui doivent œuvrer ensemble, selon un accord précis, pour accomplir des tâches en groupes [50]. Horling et al [39] définissent la formation de coalitions comme suit : *Les coalitions sont, en général, orientées vers un but de courte durée ; elles sont formées pour réaliser un objectif donné et dissoutes quand cet objectif n'existe plus, ou quand elles ne peuvent plus réaliser ce dernier, ou lorsqu'il n'y a plus d'intérêt à les former.*

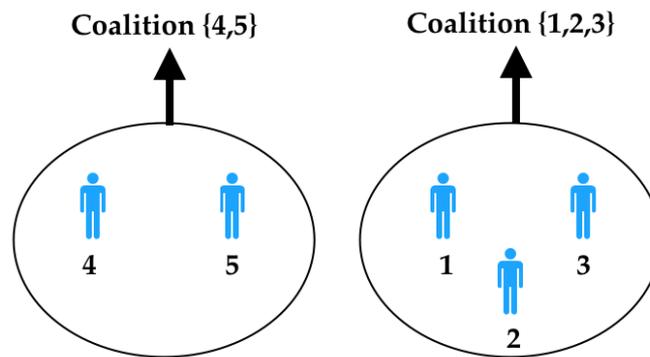


Fig. 2.1 – Exemple de formation de coalitions.

L'intérêt fondamental de faire partie d'une coalition pour les agents est de pouvoir coordonner leurs activités afin d'atteindre leurs objectifs, qu'ils soient collectifs ou individuels. La coordination est supposée s'effectuer entre les membres d'une même coalition sans imposer aucune autre forme de coordination avec les

agents faisant partie d'une autre coalition. Néanmoins, dans le cas particulier où des agents de différentes coalitions ont des buts qui interfèrent, des formes de coordinations entre différentes coalitions peuvent être envisagées.

Dès lors que la formation de coalitions est un des moyens les plus incitatifs aux agents pour coordonner leurs activités, elle devient de façon tout à fait naturelle le centre d'intérêt de plusieurs travaux de recherche en ces dernières années. À ce titre, dans le commerce électronique, il est parfaitement possible à un groupe d'agents de coopérer afin de lancer des achats groupés et de produits à des prix attractifs [97]. Dans le domaine de la recherche sur les réseaux de capteurs, des capteurs peuvent coopérer pour suivre des cibles [20]. Dans les systèmes de livraison, par la coopération de leurs activités, les entreprises deviennent plus compétitives avec des prix réduits, des destinations de livraison diverses à des temps raisonnables, etc. [69].

Malgré la diversité dans ces domaines applicatifs et les différences dans leurs objectifs, on retrouve quasiment dans tous ces domaines les mêmes questions fondamentales sur la formation de coalitions, à savoir :

1. Comment calculer la valeur de chaque coalition possible ?
2. Comment déterminer parmi l'ensemble des structures de coalitions possibles (chacune étant un ensemble de coalitions disjointes), la structure optimale (celle avec l'utilité maximale) ?
3. Comment déterminer l'influence des coalitions *dites externes* sur les coalitions *dites internes* lorsque des dépendances aux coalitions externes existent, et comment tenir compte de cette influence ?
4. Comment, après la formation d'une coalition, déterminer le degré d'importance ou d'implication de ses membres et d'en déterminer la distribution des gains en conséquence ?

Ainsi, pour former des coalitions, généralement, on a recours à différentes manières possibles, à savoir :

- Par ajustement dynamique ou non des comportements des agents. Cela se fait soit par planification soit par négociation ;

- Par calcul des valeurs de coalitions à former. Dans ce cas, le calcul peut être supervisé ou pas.

Dans la suite de ce chapitre, nous nous appuyons sur différents travaux de référence dans le domaine de la formation de coalitions [77, 78, 69, 16, 28] et nous établissons une synthèse sur les différents concepts utilisés dans ces travaux, nécessaires à la compréhension des bases de ce problème.

La notion de coalition

On considère un ensemble de n agents $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$, chacun d'entre eux étant motivé à former des coalitions avec d'autres agents.

Définition 2.1. - Coalition. Une coalition $c = \{a \mid a \in \mathcal{A}\}$ est un ensemble d'agents, tel que $c \subseteq \mathcal{A} \wedge c \neq \emptyset$.

Une coalition c est dite *la grande coalition* si elle est formée de tous les agents de \mathcal{A} (c.-à-d. $|c| = |\mathcal{A}|$), et dite *la coalition singleton* si elle est formée d'un seul agent (c.-à-d. $|c| = 1$). On dénote par \mathcal{C} , tel que $\mathcal{C} = \{c_i \mid \forall i \Rightarrow c_i \neq \emptyset\}$ l'ensemble de toutes les coalitions possibles sur l'ensemble \mathcal{A} , et par \mathcal{C}_i tel que $\mathcal{C}_i = \{c_{i'} \mid \forall i' \Rightarrow a_i \in c_{i'}\}$ l'ensemble de toutes les coalitions auxquelles l'agent a_i appartient.

Exemple 2.1. Soit l'ensemble d'agents suivant : $\mathcal{A} = \{a_1, a_2, a_3\}$.

Alors, les coalitions possibles sont :

$$\begin{aligned} c_1 &= \{\{a_1\}\}, c_2 = \{\{a_2\}\}, c_3 = \{\{a_3\}\}, \\ c_4 &= \{\{a_1, a_2\}\}, c_5 = \{\{a_1, a_3\}\}, c_6 = \{\{a_2, a_3\}\}, \\ c_7 &= \{\{a_1, a_2, a_3\}\} \end{aligned}$$

Ce qui nous donne l'ensemble \mathcal{C} suivant :

$$\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$$

Nous avons donc, c_1, c_2 et c_3 qui sont des coalitions *singleton* et c_7 qui est la *grande coalition* ($c_7 \equiv \mathcal{A}$). Les ensembles de coalitions possibles pour chacun des agents sont :

$$\begin{aligned}\mathcal{C}_1 &= \{\{\mathbf{a}_1\}, \{\mathbf{a}_1, \mathbf{a}_2\}, \{\mathbf{a}_1, \mathbf{a}_3\}, \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}\} \\ \mathcal{C}_2 &= \{\{\mathbf{a}_2\}, \{\mathbf{a}_1, \mathbf{a}_2\}, \{\mathbf{a}_2, \mathbf{a}_3\}, \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}\} \\ \mathcal{C}_3 &= \{\{\mathbf{a}_3\}, \{\mathbf{a}_1, \mathbf{a}_3\}, \{\mathbf{a}_2, \mathbf{a}_3\}, \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}\}\end{aligned}$$

Propriété 2.1. - Le cardinal de \mathcal{C} , est $|\mathcal{C}| = 2^n - 1$ avec n étant le nombre d'agents.

Alors, dans l'exemple 2.1, $|\mathcal{C}| = 2^3 - 1 = 7$.

Une structure de coalitions

Nous appelons *une structure de coalitions*, l'ensemble de tous les groupes disjoints d'agents possibles, recouvrant \mathcal{A} .

Définition 2.2. - Structure de coalitions. Une structure de coalitions sc est une partition de coalitions de \mathcal{A} , tel que :

$$sc = \{c_i | \forall c_i \Rightarrow c_i \neq \emptyset \wedge \forall i \neq j : (c_i \in \mathcal{A} \wedge c_j \in \mathcal{A}) \Rightarrow c_i \cap c_j = \emptyset \wedge \bigcup c_i = \mathcal{A}\}$$

Donc sc est définie comme étant un ensemble non vide, et les coalitions qui la composent sont des ensembles deux-à-deux disjoints. On dénote par \mathcal{SC} l'ensemble de toutes les structures de coalitions.

Exemple 2.2. Soit l'ensemble d'agents de l'exemple 2.1.

Des exemples de structures de coalitions sont :

$$\begin{aligned}\mathcal{SC}_1 &= \{\{c_1\}, \{c_2\}, \{c_3\}\}. \\ \mathcal{SC}_2 &= \{\{c_1\}, \{c_6\}\}. \\ \mathcal{SC}_3 &= \{\{c_4\}, \{c_3\}\}. \\ \mathcal{SC}_4 &= \{\{c_5\}, \{c_2\}\}. \\ \mathcal{SC}_5 &= \{c_7\}.\end{aligned}$$

Propriété 2.2. - Le cardinal de \mathcal{SC} est de l'ordre $\mathcal{O}(n^n)$ [73].

La fonction caractéristique

Dans un processus de formation de coalitions, et par le fait de l'existence de plusieurs coalitions possibles, pour répartir l'ensemble des agents en des groupes différents, il est nécessaire de différencier chaque coalition des autres par une valeur suffisamment représentative et facilement exploitable. Une *fonction caractéristique* v , appelée également *fonction de valuation*, est utilisée afin d'associer à chaque coalition une valeur numérique réelle et positive correspondant à la valeur ou l'utilité de la coalition.

$$\begin{aligned} v : \mathcal{C} &\mapsto \mathbb{R} \\ \mathbf{c} &\mapsto v(\mathbf{c}) \end{aligned}$$

Dans la pratique, pour une coalition \mathbf{c} , le calcul de sa valeur peut tenir compte de plusieurs critères, comme : la nature ou le nombre d'agents $\mathbf{a} \in \mathbf{c}$, les ressources nécessaires (si besoin) pour atteindre l'objectif de la coalition \mathbf{c} ou d'autres critères spécifiques au domaine d'application. Cependant, de manière générale, la valeur de \mathbf{c} est déterminée en fonction des agents qui la composent.

Définition 2.3. - Valeur d'une structure de coalitions. Soit une structure de coalitions $\mathbf{sc} \in \mathcal{SC}$. La valeur de la structure de coalitions \mathbf{sc} est la somme des valeurs des coalitions qui la composent :

$$v(\mathbf{sc}) = \sum_{\mathbf{c} \in \mathbf{sc}} v(\mathbf{c})$$

Exemple 2.3. Soit les deux structures de coalitions $\mathbf{sc}_1 = \{\{\mathbf{c}_1\}, \{\mathbf{c}_2\}, \{\mathbf{c}_3\}\}$ et $\mathbf{sc}_2 = \{\{\mathbf{c}_1\}, \{\mathbf{c}_6\}\}$, tel que : $v(\mathbf{c}_1) = 25$, $v(\mathbf{c}_2) = 30$, $v(\mathbf{c}_3) = 40$ et $v(\mathbf{c}_6) = 65$.

Alors, nous avons : $v(\mathbf{sc}_1) = 25 + 30 + 40 = 95$ et $v(\mathbf{sc}_2) = 25 + 65 = 90$. Avec ces valeurs de $v(\mathbf{sc}_1)$ et $v(\mathbf{sc}_2)$, il est plus avantageux pour les agents $\mathbf{a} \in \mathcal{A}$ de former les coalitions de \mathbf{sc}_1 que celles de \mathbf{sc}_2 .

2.3 Formation de coalitions avec des agents coopératifs

2.3.1 Jeux coalitionnels

Dans le cadre de la formation de coalitions avec des agents coopératifs, nous considérons que les agents ont comme objectif de résoudre un problème prédéfini. On distingue deux classes majeures, à savoir, i) les jeux coalitionnels et ii) la résolution collective de tâches. La première classe représente le problème de recherche de la structure de coalitions optimale. La deuxième classe considère que les agents sont motivés par la résolution collective par coopération de leurs tâches.

Pour décrire ce problème de manière formelle, nous rappelons dans ce qui suit les différents concepts existants dans la littérature et quelques approches de résolution.

Définition 2.4. - Jeu coalitionnel. Soit un ensemble d'agents \mathcal{A} et une fonction caractéristique v . Un jeu coalitionnel est défini comme un tuple (\mathcal{A}, v) .

Le couple formé de l'ensemble d'agents \mathcal{A} et la fonction caractéristique v qui permet de déterminer quelle structure de coalitions sc^* est plus avantageuse dans l'ensemble \mathcal{SC} , est appelé *jeu coalitionnel*. L'objectif du processus de formation de coalitions étant de diviser un ensemble d'agents en groupes, dans un jeu coalitionnel les agents sont coopératifs, dans des groupes disjoints, et sont censés se partager la recherche d'une structure de coalitions optimale et procéder à l'accomplissement collectif de tâches. De nombreux travaux ont été réalisés dans ce sens [57, 58, 59, 63, 94, 69, 73, 21, 43, 75]. L'objectif commun de tous les agents du système est de maximiser la somme des utilités des différents groupes qu'ils forment. Ainsi, dans un jeu coalitionnel (\mathcal{A}, v) , l'objectif des agents \mathcal{A} , est de trouver sc^* en fonction des valeurs des coalitions v , tel que :

$$sc^* = \underset{sc \in \mathcal{SC}}{\operatorname{argmax}} v(sc)$$

Pour atteindre cet objectif, la démarche naturelle à suivre serait alors de calculer toutes les valeurs des structures de coalitions et de sélectionner celle ayant la

valeur la plus élevée. Toutefois, l'espace de recherche des structures de coalitions est considérable d'après la propriété 2.2. Donc, effectuer une recherche exhaustive pour calculer les valeurs des structures de coalitions est très peu recommandé et ne peut se faire en un temps raisonnable. Ceci a suscité l'intérêt d'un bon nombre de recherches dans ce sens. En effet, plusieurs algorithmes ont été proposés pour que les agents puissent :

- soit se partager la recherche de la structure de coalitions optimale sc^* ;
- soit d'optimiser l'algorithme de recherche et de ne considérer que les structures de coalitions qui peuvent réellement être une solution optimale ;
- ou des algorithmes combinant à la fois le partage de l'espace de recherche et l'optimisation de l'ensemble des structures de coalitions parcourues.

Des résultats divers et variés ont été obtenus permettant d'avoir des temps limités ou du moins, d'atteindre une solution plus au moins proche de la solution optimale en fonction du temps alloué au processus de résolution. Dans le cadre des jeux coalitionnels, les agents étant collaboratifs et s'échangent leurs informations et résultats de calcul sans aucune contrainte relative à la divulgation d'information. Dans la pratique, il existe deux sous-classes à ce problème :

1. Les valeurs des coalitions ne sont pas connues par avance, alors les agents doivent les calculer et, par conséquent, se partager le calcul et trouver la solution optimale.
2. Les valeurs des coalitions sont déjà connues, alors les agents doivent se partager la recherche de la structure de coalitions optimale.

2.3.2 Calcul des valeurs des coalitions

Les travaux sur le calcul des valeurs des coalitions supposent que les agents se partagent le calcul de ces valeurs. De plus, ils supposent que les agents partagent tous la même fonction d'utilité et qu'ils se font confiance mutuellement. Par exemple, Rahwan et al. [57] proposent un algorithme DCVC de calcul des valeurs des coalitions en répartissant les calculs entre les agents en fonction de la taille des coalitions. DCVC assure un meilleur équilibre entre les calculs effectués par chaque agent tout en évitant la redondance des calculs. Avec DCVC, chaque

agent a connaissance de l'ensemble des coalitions dont il a la charge de calculer leur valeur. Les informations dont disposent les agents sont : le nombre total d'agents et leur numéro d'identification (qui doit être unique dans le système multi-agents).

2.3.3 Recherche de la structure de coalitions optimale

La recherche de la structure de coalitions optimale est un processus très coûteux en calcul si elle est effectuée de manière exhaustive et naïve [73]. Il est donc indispensable de proposer des algorithmes permettant de raccourcir les délais de la génération des structures de coalitions optimales, surtout lorsqu'il s'agit de systèmes avec un nombre important d'agents. Dans ce sens, plusieurs travaux ont été proposés pour la génération des structures de coalitions optimales (GSC) [73, 93, 60, 62, 17, 8, 36]. Tous sont basés sur des algorithmes qui explorent l'ensemble des structures de coalitions possibles en introduisant différentes techniques pour les représenter dans des structures appelées : Espaces de recherche.

Représentation de l'espace de recherche

La structuration de l'espace de recherche présente une double utilité. La première étant un moyen de garantir une régularité dans le processus de recherche sans maintenir toutes les structures de coalitions possibles en mémoire, surtout quand il s'agit d'un jeu coalitionnel avec un nombre important d'agents. La seconde est qu'elle permet de garantir une régularité dans le processus de calcul de la valeur des structures de coalitions et de réduire en conséquence le temps de recherche de la structure de coalitions optimale. Quelques travaux se sont intéressés à la représentation de l'espace de recherche pour ne tenir compte que des structures de coalitions susceptibles d'être une structure de coalitions optimale [73].

Définition 2.5. - Espace de recherche. Un espace de recherche S est un ensemble de structures de coalitions $S = \{sc | sc \text{ est potentiellement une structure de coalitions optimale } \}$, avec $S \subseteq \mathcal{SC}$.

Sandholm et al. [73] proposent un algorithme basé sur une représentation sous

la forme d'un graphe pour la recherche de la structure de coalitions optimale. L'algorithme développé possède la particularité de pouvoir sélectionner une structure de coalitions avec une certaine garantie qu'elle soit proche de la solution optimale même en cas d'interruption de l'algorithme, notamment dans des cas où les agents ont un temps limité de recherche.

Exemple 2.4. Soit l'ensemble composé de 4 agents : $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$. Il existe 15 structures de coalitions possibles.

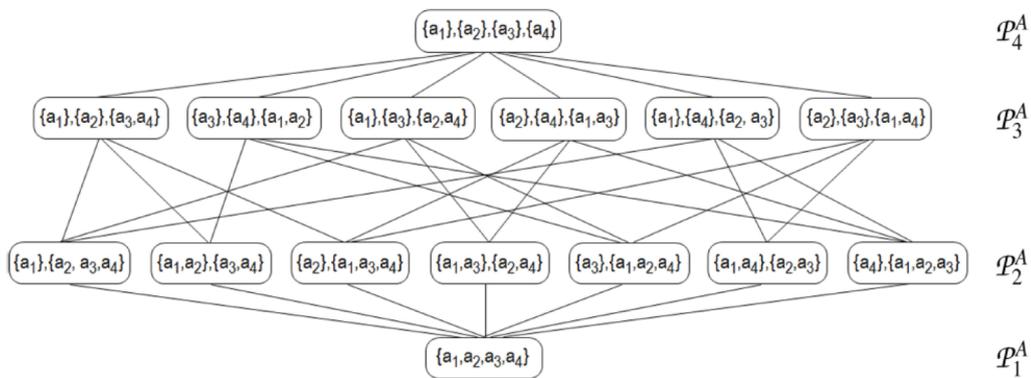


Fig. 2.2 – Représentation de l'espace de recherche pour un problème avec 4 agents.

Dans cet algorithme, l'espace de recherche représentant les structures de coalitions possibles est organisé sous la forme d'un graphe hiérarchique de différents niveaux, selon le nombre possible d'agents dans les coalitions de chaque niveau. La figure (*cf.* Fig. 2.2) illustre les 4 différents niveaux pour l'exemple 2.4.

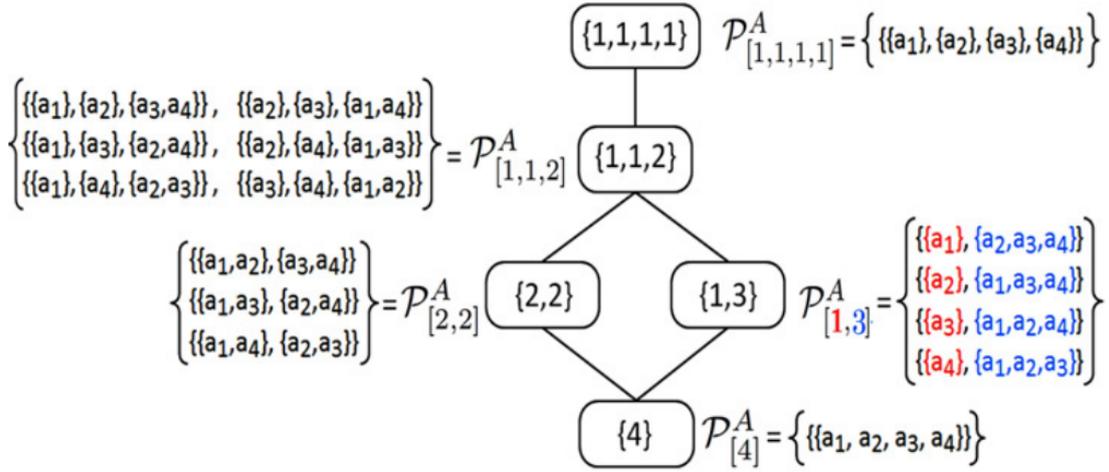
Dans Sandholm et al. [73, 69], chaque nœud correspond à une structure de coalitions et appartient à un niveau hiérarchique selon le cardinal de l'ensemble des structures de coalitions ; celles qui sont composées de i coalitions appartiennent au niveau P_i^A . En commençant du haut du graphe vers le bas, le sommet représente toutes les coalitions singleton et le bas du graphe représente la grande coalition. Deux structures de coalitions appartenant à deux niveaux successifs n et $n - 1$ sont reliées, s'il est possible d'obtenir une structure du niveau $n - 1$ à partir de l'union de deux coalitions du niveau n .

L'idée de l'algorithme est donc assez basique. Elle consiste à commencer la recherche de sc^* par le haut du graphe, avec 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} .

Dang et al. [21] se basent sur la représentation de l'espace de recherche proposé par [73] et proposent un algorithme avec une autre méthode de parcours qui explore le graphe en suivant des sous-ensembles de structures de coalitions dépendant du nombre de coalitions et du nombre d'agents par coalition. L'algorithme produit une amélioration significative de celui de [73] en termes de temps de recherche. Néanmoins, l'algorithme ne réduit pas la complexité de recherche de la solution optimale qui nécessite cependant un parcours de la totalité des structures de coalitions.

Rahwan et al. [93] proposent un des algorithmes les plus performants dans cette catégorie en utilisant une représentation basée sur la taille des coalitions. Ainsi, ils calculent une valeur appelée *Integer-Partition*, et notée I (comme illustré par la figure Fig. 2.3) qui représente la taille des coalitions. Cette valeur est par la suite utilisée pour former des configurations P_1^A . Une configuration étant une façon de représenter un ensemble de structures de coalitions selon les coalitions qu'elles contiennent. Par exemple, les structures de coalitions $\{\{a_1\}, \{a_2, a_3\}\}$ et $\{\{a_3\}, \{a_1, a_2\}\}$ ont la même configuration notée $P_{\{1,2\}}^A$ ce qui veut dire qu'elles ont toutes les deux une première coalition formée d'un seul agent et une deuxième formée de deux agents. Toutes les configurations sont illustrées par la figure Fig. 2.3. Les auteurs ont par la suite développé plusieurs autres améliorations dans [58, 59, 63].

Une des différences majeures avec l'algorithme de Sandholm et al [73], réside dans le fait que l'algorithme de Rahwan et al. [93] 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 P_1^A) ce qui permet d'éviter la redondance des calculs. Cela fait que l'algorithme améliore progressivement les bornes en éliminant les configurations qui n'ont pas une meilleure borne que la dernière meilleure borne trouvée. L'algorithme s'arrête à la fin de la liste des configurations.


 Fig. 2.3 – Le graphe des *Integer-Partitions*.

Pour un ensemble de n agents, la complexité de cet algorithme est de l'ordre $O(n^n)$. Cependant, il permet de retourner en cas d'interruption la meilleure solution retrouvée à l'instant où il a été interrompu, même si l'espace de recherche n'a pas été entièrement exploré.

Dans Rahwan et al. [60], les auteurs présentent la première étude computationnelle des jeux coalitionnels avec des dépendances inter-coalitions, ce qui revêt une importance lorsque les objectifs des agents sont contradictoires ou se chevauchent. L'objectif est de trouver à tout moment une division disjointe des agents en coalitions.

Bistaffa et al. [8] traitent le problème du covoiturage social en représentant l'ensemble des participants par un réseau social. Ils proposent un algorithme pour former des coalitions et organiser des trajets ponctuels à court terme. Deux aspects fondamentaux y sont abordés. Premièrement, le problème d'optimisation de la formation des coalitions qui minimisent le coût du système global. Deuxièmement, ils traitent l'aspect de la répartition des paiements.

[101] et [47] proposent deux algorithmes d'une complexité en $O(3^n)$, avec n étant le nombre d'agents. Ces algorithmes, basés sur la programmation dynamique, traitent le problème de partitionnement et le problème d'enchères combinatoires,

mais sont aisément adaptables au problème de la recherche de la structure de coalitions optimale. Ils se basent sur l'idée de la subdivision du problème d'optimisation en plusieurs sous-problèmes, pour ensuite les résoudre de façon récursive. À la fin de la résolution, les résultats sont combinés pour produire une solution au problème tel qu'il est défini initialement. Cependant, ces deux algorithmes ne produisant de résultat final que si une structure de coalitions optimale est trouvée. Cela constitue une faiblesse majeure puisque la recherche peut s'avérer très longue dans bon nombre de situations. Par ailleurs, ils nécessitent également un espace mémoire non-négligeable pour les calculs (trois tables, dont une qui contient les $(2^n - 1)$ entrées).

Les auteurs dans [52] utilisent les méthodes d'optimisation de contraintes distribuées (Distributed Constraint Optimization Problem (DCOP)) et proposent une approche où chaque agent a un gain à emporter, qui est déterminé par l'ensemble des actions qu'il va devoir réaliser. L'objectif de chacun des agents est alors, de choisir les actions qui peuvent maximiser ses gains. DCOP est aussi utilisé par Ueda et al. [68] pour résoudre le problème de la génération des structures de coalitions, et proposent une variante de DCOP, présentée comme une résolution du problème de calcul de la structure de coalitions optimale par la résolution d'un DCOP. Les auteurs proposent de combiner les calculs nécessaires des valeurs des coalitions avec la recherche de la structure de coalitions optimale en utilisant un paramètre σ . L'idée est d'éviter des calculs exhaustifs des valeurs des 2^n coalitions possibles pour n agents. Le paramètre σ étant spécifique à chaque domaine d'application, il représente le nombre maximum de coalitions qui peuvent être formées de plus d'un agent.

Dans leurs travaux [91], Rahwan et al. traitent le problème de la formation de coalitions avec des contraintes (Constrained Coalition Formation (CCF)) et proposent une modélisation sous la forme d'un ensemble de contraintes distribuées. Ces contraintes peuvent être de différentes natures, en fonction du domaine d'application, comme les préférences des agents, leurs compétences, ou des contraintes liées à leur environnement tout simplement. Ainsi, elles peuvent avoir une influence sur les groupes d'agents et limiter les coalitions dans lesquelles ils peuvent faire

partie. Avec l'hypothèse de l'existence de contraintes inhérentes au processus de formation de coalitions, les auteurs proposent de nouveaux concepts, à l'image de l'ensemble de structure de coalitions réalisable ($E_S C$) ce qui leur permet de réduire l'espace de recherche des coalitions optimales sans évaluer les 2^n coalitions possibles pour n agents. Formellement, le modèle de CCF est défini par la forme, $\{A, E_S C, v\}$ où A est un ensemble d'agents et v est une fonction caractéristique.

2.3.4 Allocation de tâches

Le problème de l'allocation de tâches est un tuple $\langle A, T, RA, RT \rangle$ avec $A = \{a_1, a_2, \dots, a_n\}$, $A \neq \emptyset$ est un ensemble de n agents, $T = \{t_1, t_2, \dots, t_m\}$, $T \neq \emptyset$, un ensemble de m tâches, $RA = [ra_a^1, \dots, ra_a^p]$ un vecteur de p types de ressources que possèdent les agents et $RT = [r_t^1; \dots; r_t^p]$ un vecteur de p types de ressources nécessaires pour la réalisation des tâches dans T [76, 77, 1].

Ainsi, ce problème représente également une des formes de la formation de coalitions pour la réalisation d'un ensemble de tâches par des groupes d'agents. D'un côté, la réalisation de chaque tâche $t \in T$ nécessite des quantités en chacun des p types de ressources de l'environnement $R_t = [r_t^1, \dots, r_t^p]$ et de l'autre côté, chaque agent $a \in A$ dispose aussi de quantités en chacun des p types de ces ressources $R_a = [r_a^1, \dots, r_a^p]$. Il est bien évidemment possible qu'un agent a réalise une tâche t seul s'il en a la capacité (a dispose de plus de ressources que les ressources nécessaires à la réalisation de t , pour tous les types p), c'est-à-dire : $\forall k \in [1, p], r_a^k \geq r_t^k$. Néanmoins, un agent n'est pas nécessairement capable de réaliser une tâche seul, ou de manière plus générale, il pourrait être plus important de réaliser les tâches avec une meilleure exploitation des ressources des agents. Dans ce cas de figure, l'utilité n'est plus une fonction du groupe d'agents, mais du groupe de tâches à réaliser. Ces dernières sont communes à tous les agents du système. L'objectif commun aux agents est alors de trouver les meilleurs couples (agent(s), tâche(s)) pour la réalisation des tâches du système.

Ainsi, pour la réalisation de l'ensemble des tâches, le problème consiste à trouver quel groupe d'agents est meilleur pour la réalisation de quel groupe de tâches. La réalisation d'une tâche par une coalition c n'est possible que si la totalité des

ressources que les agents dans c peuvent conjointement réunir est supérieure ou égale aux ressources nécessaires à la réalisation de la tâche, et cela, pour tout type de ressources. En d'autres termes, il faut que la condition suivante soit vérifiée :

$$\forall k \in [1, p] \Rightarrow \sum_{a \in c} r_a^k \geq r_t^k.$$

Alors, un groupe d'agent $g \subseteq A$ serait capable de réaliser un ensemble de tâches $b \subseteq T$ si et seulement si :

$$\forall k \in [1, p], \sum_{a \in g} r_a^k \geq \sum_{t \in b} r_t^k$$

C'est-à-dire, l'ensemble des ressources mises en commun par les agents, est supérieur à la somme des ressources de toutes les tâches dans b , pour tous les types de ressources.

Une coalition d'agent c reçoit un paiement de chaque réalisation d'une tâche t à la base d'une fonction de paiement qui dépend du domaine étudié. Alors, la résolution du problème de l'allocation de tâches par la formation de coalitions consiste à partitionner l'ensemble des agents en m coalitions c_i disjointes et d'assigner une tâche t à chaque coalition, de telle manière à ce que la somme des paiements $\sum_{c_i} v(c_i, t_k)$ soit maximale.

Les domaines d'application de la formation de coalitions par allocation de tâches sont très variés. Sherief et al. [1], proposent un algorithme sur un ensemble d'agents où il y a une structuration hiérarchique. En effet, il existe deux types d'agents : i) Les agents managers et ii) Les agents subordonnés. Un agent manager connaît les ressources de ses subordonnés et un agent subordonné peut à son tour être un agent manager d'autres agents subordonnés, etc. La fonction de paiement dépend directement de la tâche à réaliser $v : T \mapsto \mathbb{R}^+$. Ainsi, l'algorithme consiste à déterminer les agents capables de réaliser une tâche t en effectuant une recherche guidée par la hiérarchie définie sur les agents, du subordonné au manager. Plus précisément, à la récupération d'une tâche à réaliser, par un agent manager, ce dernier prend soit la décision de sa réalisation par son équipe (lui et ses subordonnés), soit, s'ils sont dans l'incapacité à la réaliser, il fait passer la tâche à son supérieur hiérarchique.

Dans le domaine des réseaux de capteurs, Horling et al. [10] formalisent le problème de surveillance avec des capteurs comme un problème d'allocation de tâches où une tâche correspond à la surveillance d'une cible. Dans le même domaine [49] proposent des algorithmes par négociations directes pour la formation de coalitions pour accomplir des tâches de surveillance. Soh et al. proposent plusieurs travaux [87, 88, 85, 86], basés sur des stratégies de négociation utilisant une base de cas, dans l'objectif de sélectionner des capteurs voisins motivés pour la formation de coalitions.

Dans le domaine de la coopération de satellites, Bonnet et al. [9] proposent un algorithme permettant aux satellites de former des coalitions pour la prise de photographies de la Terre. Chaque requête d'observation est considérée comme une tâche à traiter. Les satellites n'étant pas capables de réaliser toutes les tâches seuls, ils forment des coalitions pour mettre en commun leurs capacités à les réaliser.

Dans le domaine de la recherche d'information dans des documents électroniques, Aknine et al. [3] proposent un système où il existe des agents spécialisés dans la recherche d'une information précise qui effectuent une recherche coopérative sur plusieurs documents.

2.4 Formation de coalitions avec des agents compétitifs

2.4.1 Jeux non coopératifs

Dans le cadre des jeux non coopératifs, les agents sont considérés comme compétitifs et rationnels dans leur prise de décision. C'est-à-dire qu'ils agissent pour satisfaire leurs propres objectifs et maximiser leur utilité, sans se soucier de l'utilité des autres agents. Dans ce cas de figure, on distingue deux catégories de jeux, à savoir, les jeux à utilité transférable et les jeux à utilité non transférable :

- Les jeux à utilité transférable : Lorsque des agents forment une coalition, ils reçoivent une utilité commune qu'ils doivent ensuite se partager.
- Les jeux à utilité non transférable : Chaque agent possède sa propre fonction

d'utilité sur les coalitions auxquelles il peut appartenir.

Dans cette section nous allons introduire le problème de la formation de coalitions dans le cas d'agents compétitifs. Nous présentons les différents concepts nécessaires à la compréhension de ce problème. Notons que nous nous appuyons sur les travaux [41, 55, 80] et reprenons les définitions qui y sont exposées dans le cadre des jeux coalitionnels pour des agents coopératifs. 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 grande coalition est une coalition équivalente à A , qui est formée par tous les agents et une coalition avec un seul agent est appelée coalition singleton. La fonction $v : 2^A \mapsto \mathbb{R}^+$ appelée fonction caractéristique, attribue une utilité pour chaque coalition.

2.4.2 Jeux à utilité transférable

Un jeu à utilité transférable est défini par un tuple (A, v) avec A un ensemble de n agents et v une fonction caractéristique. Une coalition c permet de générer un gain $v(c)$ que les agents $a \in c$ doivent se partager entre eux. Ainsi, chacun des agents a_i reçoit un paiement x_i .

La valeur d'une coalition n'est pas forcément suffisante pour permettre aux agents de se fixer sur la coalition la plus bénéfique. Ils recourent également à une distribution de paiement, qui elle aussi est sujette à différents paramètres dont il faut tenir compte.

Distribution des paiements

Une distribution de paiement, notée \vec{x}_c , d'une coalition c tel que $|c| = m$, est définie par un vecteur $\vec{x}_c = (x_1, x_2, \dots, x_m) \in (\mathbb{R}^+)^m$, avec $\forall a_i \in c, x_i$ est l'utilité que reçoit a_i .

Exemple 2.5. Soit 3 voisins (a_1, a_2, a_3) résidants dans le même quartier. Chaque voisin a une pompe à eau fonctionnant par gasoil. Les trois voisins ont chacun un budget limité. Par exemple, a_1 a 400 euros, a_2 a 300 euros et a_3 a 500 euros. L'objectif de chaque voisin est de maximiser le volume de gasoil qu'il peut se faire

livrer avec son budget. Si la citerne de 500l coûte 700 euro, celle de 750l coûte 900 euro et celle de 1000l coûte 1100 euro, alors aucun des voisins n'est capable de se faire livrer une citerne complète à lui seul. Cependant, il leur est tout à fait possible s'ils coopèrent les uns avec les autres d'augmenter leur pouvoir d'achat.

Dans le cadre des jeux à utilité transférable, l'utilité des agents peut être vue comme de la monnaie qu'ils peuvent gagner, économiser ou s'échanger. Le but de chacun est d'accumuler le plus de bénéfices possibles. Dans l'exemple des trois voisins (Exemple 2.5), le gain à gagner (la récompense) de chaque groupe de voisins est le volume de gasoil à acheter. Par ailleurs, le partage de la citerne peut se faire selon une configuration de paiement convenue entre les voisins.

Définition 2.6. - Configuration de paiement. Soit (A, v) un jeu coalitionnel. Une configuration de paiement est définie par le couple (sc, \vec{x}) où $sc \in \mathcal{SC}$ est une structure de coalitions, $\vec{x} = (x_1, x_2, \dots, x_n) \in (\mathbb{R}^+)^n$ est un vecteur indiquant les paiements que reçoivent tous les agents du système (L'agent a_i recevant le paiement x_i).

Exemple 2.6. Soit les trois voisins $\{a_1, a_2, a_3\}$ de l'exemple 2.5, avec les valeurs des coalitions suivantes : $v(\{a_1\}) = 40, v(\{a_2\}) = 30, v(\{a_3\}) = 50, v(\{a_1, a_2\}) = 100$ et $v(\{a_1, a_2, a_3\}) = 200$. Si les trois agents forment la grande coalition $(\{a_1, a_2, a_3\})$, ils doivent se partager $v(\{a_1, a_2, a_3\}) = 200$. Cependant, il existe différentes façons d'établir ce partage. Par exemple, la distribution $\{400, 400, 200\}$ signifie que l'agent a_1 reçoit $x_1 = 400$, l'agent a_2 reçoit $x_2 = 400$ et l'agent a_3 reçoit $x_3 = 200$.

Nous notons par CP l'ensemble de toutes les configurations de paiement.

Définition 2.7. - Distribution de paiement faisable. Une distribution de paiement qui vaut $\vec{x}^c = (x_1, x_2, \dots, x_m) \in (\mathbb{R}^+)^m$ est dite *faisable*, si :

$$\sum_{i=1}^m x_i \leq v(c)$$

Définition 2.8. - Distribution de paiement efficace. Une distribution de paiement qui vaut $\vec{x}^c = (x_1, x_2, \dots, x_m) \in (\mathbb{R}^+)^m$ est dite *efficace*, si :

$$\sum_{i=1}^m x_i = v(c)$$

Dans le reste de cette section, nous considérons uniquement les distributions de paiement efficaces. C'est-à-dire, pour toutes les coalitions, les agents membres se partageront exactement la valeur de la coalition en question.

Définition 2.9. - Dominance de Pareto. Soit $(sc, \vec{x}), (sc', \vec{x}') \in CP^2$ deux configurations de paiement de deux structures de coalitions qui peuvent être 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 :

$$\forall i \in [1, n], x_i \geq y_i$$

- sc domine strictement sc' 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}') et qu'au moins un agent reçoit strictement un meilleur paiement strictement, c'est-à-dire :

$$(\forall i \in [1, n], x_i \geq x'_i) \wedge (\exists j \in [1, n], x_j > x'_j)$$

Un des principaux critères qui a une influence directe sur l'acceptation d'un agent a_i d'une distribution de paiement est ce qu'on appelle la rationalité individuelle.

Propriété 2.3. - Rationalité individuelle. Pour tout agent compétitif et rationnel a_i , pour qu'il accepte de former une coalition c pour un paiement x_i , il faudrait que x_i soit supérieur à ce qu'il pourrait obtenir en formant la coalition singleton : $x_i \geq v(\{a_i\})$.

Alors, pour qu'une coalition c soit acceptée par tous les agents $a_i \in c$, la rationalité individuelle est l'une des premières conditions à vérifier concernant une distribution de paiement. Cependant, cette condition ne garantit pas une distribution équitable des bénéfices de c .

Exemple 2.7. Soit le jeu (A, v) tel que :

- $A = \{a_1, a_2\}$,
- $v(\{a_1\}) = v(\{a_2\}) = 10$ et $v(\{a_1, a_2\}) = 30$.

Si avec le jeu (A, v) , les agents ont une distribution de paiement $(\{A\}, (5, 25))$, l'agent a_1 n'acceptera pas de former la coalition c . Parce que la distribution de paiement adoptée ne lui garantit pas la rationalité individuelle. Dans cette situation, a_1 préférera de former la coalition singleton $\{a_1\}$ pour obtenir $x_1 = 10$ qui est meilleur que 5 avec la distribution de paiement associée à la coalition $\{a_1, a_2\}$. Au final, l'agent a_2 serait alors contraint de former lui aussi la coalition singleton $\{a_2\}$, qui lui permet d'obtenir $x_2 = 10$. Cela nous donnera une autre distribution de paiement qui est : $(\{\{a_1\}, \{a_2\}\}, (10, 10))$.

On voit bien dans l'exemple ci-dessus que la rationalité individuelle ne garantit pas une configuration de paiement équitable et elle peut être plus avantageuse pour certains agents au détriment des autres. En effet, sur l'exemple précédent, la configuration $(\{A\}, (10, 20))$ est plus avantageuse pour a_2 qui récupère entièrement le surplus gagné par la formation de $\{\{a_1, a_2\}\}$.

2.4.3 Jeux à utilité non transférable

Dans les systèmes multi-agents compétitifs l'utilité d'une coalition n'est pas toujours transférable. C-a-d, il peut y avoir des systèmes avec des agents qui ont leur propre fonction d'utilité ou leur propre relation de préférence sur les coalitions possibles auxquelles ils peuvent appartenir. On appelle de tels jeux coalitionnels : *jeux à utilité non transférable*.

Définition 2.10. - Jeu à utilité non transférable. Un jeu coalitionnel à utilité non transférable est un couple (A, u) tel que :

- A est un ensemble fini de n agents. $A = \{a_1, a_2, \dots, a_n\}$.
- $u = \{u_1, u_2, \dots, u_n\}$ est un ensemble de n fonctions d'utilité u_i , correspondant à chaque agent a_i , sur l'ensemble des coalitions auxquelles a_i appartient.

$$\begin{aligned} u_i : C_i &\mapsto \mathbb{R}^+ \\ c &\mapsto u_i(c) \end{aligned}$$

Dans un jeu à utilité non transférable, la fonction d'utilité d'un agent dépend de ses actions dans sa coalition et n'est pas connue par les autres agents. Il y a

donc une différence majeure avec les jeux à utilité transférable. D'une part, chaque agent dispose de sa propre fonction d'utilité sur les coalitions dont il est membre et d'autre part, les utilités entre agents ne sont pas comparables.

Exemple 2.8. Un des exemples classiques avec une utilité non transférable est celui des chercheurs universitaires. Différents chercheurs issus de différentes universités peuvent former des associations (coalitions) dans le cadre des projets de recherche et œuvrer ensemble pour la publication d'articles scientifiques. Publier un article nécessite qu'un groupe de chercheurs coopèrent. Mais la récompense que peut espérer un chercheur suite à la publication d'un article dépend de son université d'affiliation (promotion, bonus, etc.). Ici, il est clair que cette récompense n'est pas transférable entre les chercheurs.

Dans le cadre de cette thèse, nous considérons des jeux coalitionnels où les agents ont des fonctions d'utilité privées, différentes les unes des autres, et qui induisent des relations de préférence sur les coalitions possibles pour un agent donné.

Deux cas ont été traités dans la littérature :

1. Une coalition qui peut être influencée par d'autres coalitions, toutes formées simultanément. Cela est valable dans les cas de recherche de la structure de coalitions optimale par exemple.
2. Une coalition qui peut être influencée par d'autres coalitions déjà formées avant elle.

Dans nos travaux, nous présentons un troisième cas qui est celui d'une coalition qui peut être influencée par des coalitions de la même structure de coalitions, et même d'autres coalitions issues d'autres structures de coalitions. Ici, chaque agent peut apparaître dans différentes structures de coalitions. De plus, une coalition peut être influencée par, à la fois, des coalitions déjà formées ou des coalitions qui peuvent être formées dans le futur.

Ainsi, on voit bien que le problème de la formation de coalitions avec des dépendances aux coalitions externes est différent de celui de la formation de coalitions sans dépendances aux coalitions externes. De ce fait, dans les chapitres suivants,

nous allons présenter de façon approfondie nos travaux de recherche tout en introduisant de nouvelles définitions des concepts liés à la fois à ce nouveau cadre et à sa résolution.

2.5 Conclusion du chapitre

Quand il y a besoin de faire réaliser des tâches d'une manière collective entre plusieurs agents compétitifs, les techniques de formation de coalitions sont particulièrement intéressantes. L'idée est de permettre à des groupes d'agents d'agir collectivement. Nous avons exposé dans ce chapitre un panel des travaux existants sur la formation de coalitions.

Les méthodes existantes dans la littérature permettent de traiter un nombre considérable de problèmes de formation de coalitions. Cependant, nous avons constaté qu'aucune de ces méthodes ne traite le problème de la formation de coalitions avec des externalités dynamiques tel qu'il est décrit dans nos travaux. Généralement, les mécanismes de formation de coalitions ne considèrent pas le cas d'existence de dépendances entre les tâches. De plus, l'ensemble des tâches est considéré comme commun entre tous les agents. Par ailleurs, l'aspect dynamique et changeant des dépendances n'a jamais été abordé. En revanche, dans le cadre de nos travaux, nous traitons ces cas particuliers et nous apportons des solutions dans les contextes distribué et centralisé.

Deuxième partie

CFDEP : un problème de formation
de coalitions avec externalités
dynamiques (Analyse et solutions)

CHAPITRE 3

ÉTAT DE L'ART SUR LA FORMATION DE COALITIONS AVEC EXTERNALITÉS

Nous introduisons dans ce chapitre un problème de formation de coalitions d'agents compétitifs avec des buts individuels à satisfaire. Les agents possèdent chacun un jeu d'alternatives formées de tâches interdépendantes et ont pour objectif de déterminer la meilleure alternative pour atteindre leurs buts. Nous appelons ce problème : le problème de formation de coalitions avec externalités dynamiques (CFDEP – Coalition Formation with Dynamic Externalities). Nous montrons que l'ensemble (dépendances, alternatives) donne lieu à une nouvelle classe d'externalités, que nous appelons Externalités d'Existence. L'illustration applicative de ce problème est donnée à travers l'exemple, MATP – Multi-alternative Agent Traveling Problem – dans deux contextes distincts : distribué et centralisé.

3.1 Introduction

LE champ des travaux de recherche sur la formation de coalitions multi-agents est très vaste. Cependant, il en demeure que la plupart des travaux [74, 69, 73, 77, 78, 8, 4, 60, 62, 92, 57, 63, 64, 61] sont fondés sur l'hypothèse que les coalitions formées sont (et restent) stables et que la valeur d'une coalition ne

dépend que d’elle-même. Néanmoins, certains travaux [77, 44, 98, 64, 6, 14, 102] étendent ces recherches et présentent des applications où une coalition peut être directement influencée par différentes coalitions dans le système. Ainsi, dans ce contexte, la valeur d’une coalition ne dépend plus que d’elle-même, mais aussi d’autres coalitions formées [92]. À titre d’exemple, dans le cadre du problème d’allocation de ressources, une coalition qui peut engendrer une augmentation dans la consommation des ressources exploitées par d’autres agents, peut avoir une influence directe sur la capacité d’autres agents au sein d’autres coalitions à réaliser leurs tâches, et cela en raison de l’exploitation des ressources dont ils ont besoin par les agents d’autres coalitions. Cette influence que peut avoir une coalition sur une autre est appelée dans la littérature : *Externalité*.

Dans ce chapitre, nous introduisons un nouveau problème, appelé *Formation de Coalition avec des Externalités Dynamiques* (CFDEP – *Coalition Formation with Dynamic Externalities* –) et nous illustrons ses challenges à travers un exemple applicatif appelé, *Multi-alternative Agent Traveling Problem* (MATP) que nous traitons dans deux contextes, à savoir : le contexte distribué et le contexte centralisé. Nous expliquons par la suite comment MATP est un problème de formation de coalitions multi-agents avec des externalités et nous présentons pour chacun des contextes un modèle du problème. Nous utilisons ces modèles pour présenter deux algorithmes de résolution.

3.2 Externalités de coalitions

La formation de coalitions avec externalités a été définie comme un problème de formation de coalitions où dans l’ensemble du système l’intérêt de former une coalition peut être affecté par d’autres coalitions concurrentes [51, 61, 81]. Les influences entre les coalitions constituent des dépendances inter-coalitions qui jouent un rôle prépondérant dans de nombreuses applications multi-agents du monde réel [61]. Nous proposons dans ce qui suit de nouveaux concepts fondés sur une nouvelle notion d’externalités et nous étudions les particularités du problème CFDEP. Notons qu’une partie de ces travaux est publiée dans le journal *Engineering Ap-*

plications of Artificial Intelligence [82].

3.2.1 Concepts et définitions

Traditionnellement, en théorie des jeux, les externalités sont liées à la fusion de deux coalitions dans un système. Plus précisément, il s'agit d'un changement de valeur d'une coalition donnée, causée par la fusion de deux autres coalitions distinctes. Plus formellement, une définition conventionnelle d'une externalité est la suivante [95] :

Définition 3.1. - Externalité de coalition. Soit cs et cs' deux structures de coalitions avec $cs = \{c_1, c_2, c_3\}$ et $cs' = \{c_1, c_4\}$. cs' est formée à partir de cs par fusion des deux coalitions c_2 et c_3 ($c_4 = c_2 \cup c_3$). L'externalité de la nouvelle coalition c_4 sur la coalition c_1 est mesurée comme suit : $v(c_1, cs') - v(c_1, cs)$ avec $v(c_1, cs)$ est la valeur de la coalition c_1 dans cs .

La valeur de l'externalité d'une coalition dépend alors des coalitions qui forment la fusion. La figure Fig. 3.1 illustre un exemple où la fusion de $\{a_3\}$ avec $\{a_4, a_5\}$ induit une externalité différente sur $\{a_1, a_2\}$ que celle de la fusion de $\{a_4\}$ avec $\{a_3, a_5\}$. En particulier, $v(\{a_1, a_2\}, cs) - v(\{a_1, a_2\}, cs') = 9 - 7 = 2$, tandis que $v(\{a_1, a_2\}, cs) - v(\{a_1, a_2\}, cs'') = 9 - 10 = -1$. En d'autres termes, l'externalité est une fonction à la fois de la coalition intégrée dans cs , c'est-à-dire (c, cs) , et de la structure de coalitions à partir de laquelle cs a été créée.

Dans la littérature, l'existence d'externalités entre les coalitions peut s'effectuer selon deux cas de figures [95] :

1. Des externalités vers l'extérieur (*cf.* Fig. 3.2)
2. Des externalités vers l'intérieur (*cf.* Fig. 3.3)

Exemple 3.1. Soit l'ensemble d'agents $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5\}$. Une des structures de coalitions possibles est $cs = \{c_1 = \{a_1, a_2\}, c_2 = \{a_3\}, c_3 = \{a_4, a_5\}\}$.

Dans le premier cas (*cf.* Fig. 3.2), on s'intéresse à l'influence d'une coalition c sur les autres coalitions dans une structure de coalitions cs sans les influences des coalitions les unes sur les autres. La fonction caractéristique des coalitions donne

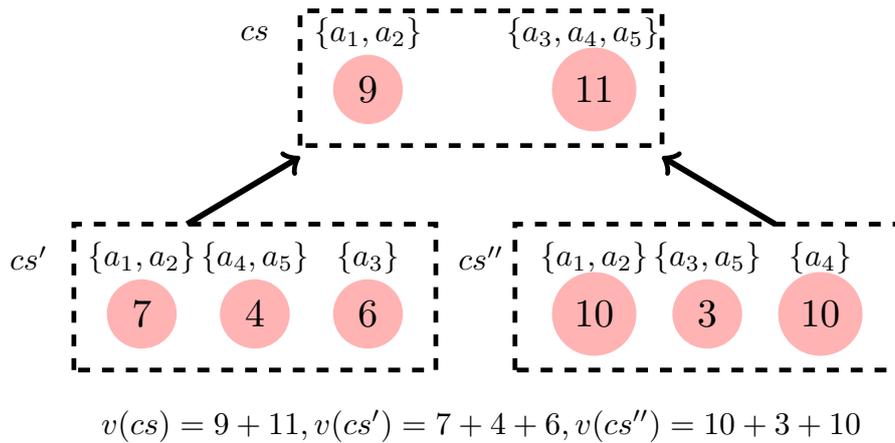


Fig. 3.1 – Des exemples d'externalités.

les valeurs suivantes : $v(c_1) = 5$, $v(c_2) = 9$ et $v(c_3) = 2$. Dans le cas d'externalités vers l'extérieur, la coalition c_1 fait augmenter la valeur de $v(c_3)$ de 4 et fait diminuer la valeur de $v(c_2)$ de 1, et $v(c_2)$ fait diminuer $v(c_1)$ de 1 et fait augmenter $v(c_3)$ de 2. Enfin, $v(c_3)$ fait augmenter $v(c_1)$ et $v(c_1)$ de 2.

Dans le deuxième cas (*cf.* Fig. 3.3), on s'intéresse à l'influence des autres coalitions sur une coalition donnée. En l'absence d'externalités, les valeurs de la fonction caractéristique des coalitions sont : $v(c_1) = 5$, $v(c_2) = 9$ et $v(c_3) = 2$. Avec les externalités, nous avons : $v(c_1)$ est affectée par une augmentation par 4, $v(c_2)$ est affectée par une diminution de 3 et $v(c_3)$ est affectée par une augmentation de 5.

3.2.2 Les externalités dans CFDEP

Dans les travaux existants, la notion d'externalité se limite à l'effet que peut avoir une coalition formée par la fusion d'autres coalitions sur une coalition existante. Dans nos travaux, nous introduisons un nouveau type d'externalités qui s'intéresse à la raison d'être d'une coalition si d'autres coalitions existent. En d'autres termes, nous nous intéressons à l'effet que peut avoir la formation d'une coalition c sur la possibilité de former une autre coalition c' ou pas. Ainsi, une coalition peut devenir inutile ou impossible à former à cause d'une autre coalition. Nous appelons ce type d'externalités : *Externalités d'existence*.

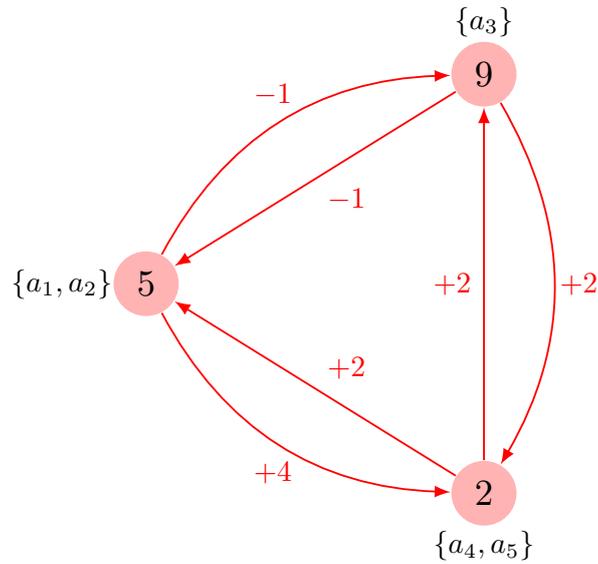


Fig. 3.2 – Externalités vers l'extérieur.

Définition 3.2. - Externalité d'existence. Soit c et c' deux coalitions. On dit qu'il existe une externalité d'existence entre c et c' (ou entre c' et c), si la réalisation de c rend impossible la réalisation de c' et la réalisation de c' rend impossible la réalisation de c .

Dans CFDEP, les agents compétitifs doivent prendre en compte les dépendances entre leurs tâches, qui peuvent être différentes d'un agent à un autre. Du fait que chacun d'eux peut avoir différentes alternatives, il en résulte que les externalités d'une coalition peuvent affecter celles des autres coalitions. On dit alors que les externalités sont dynamiques et changeantes. En effet, l'ensemble des externalités change à chaque choix d'alternative souhaitée par un agent. C'est-à-dire, à un instant t , l'ensemble des externalités valables pour une coalition c donnée dépend des autres coalitions considérées par les autres agents.

Exemple 3.2. Soit l'ensemble d'agents $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ et les coalitions $\{c_1, c_2, c_3, c_4, c_5, c_6\}$, comme illustré par la figure Fig. 3.4. Supposons que l'agent a_1 peut faire partie de toutes les coalitions possibles et qu'il veut atteindre un objectif qui ne peut se réaliser qu'à travers la coalition c_5 . La coalition c_3 nécessite

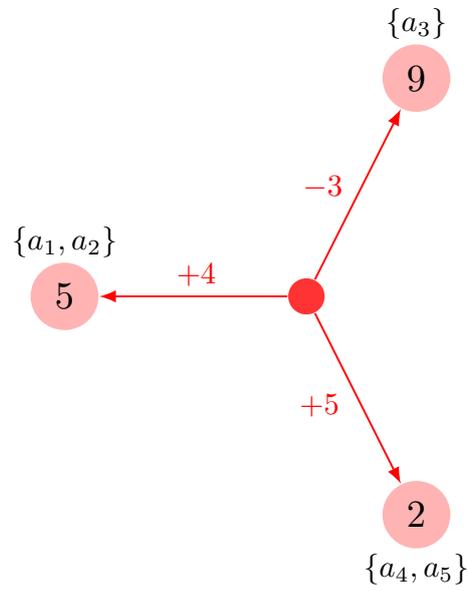


Fig. 3.3 – Externalités vers l'intérieur.

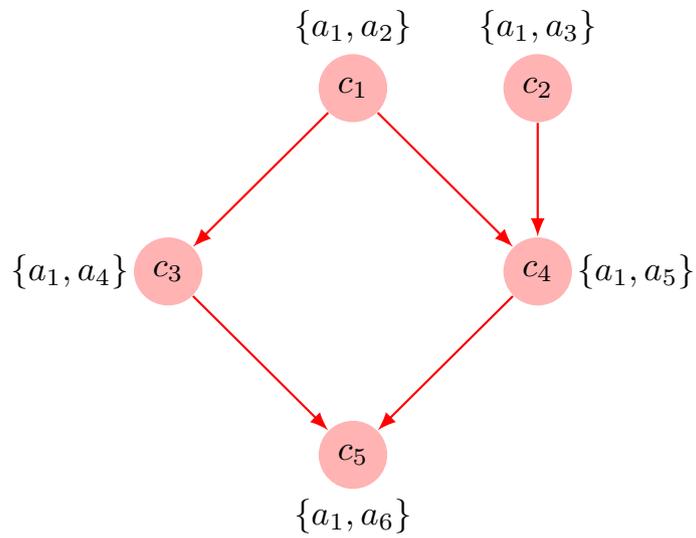


Fig. 3.4 – Externalités d'existence.

le résultat de la coalition c_1 et c_5 nécessite soit le résultat de c_3 ou celui de c_4 . À son tour, c_4 nécessite soit celui de c_1 ou celui de c_2 . Alors, nous sommes devant une situation où il existe des dépendances entre les coalitions possibles. Si l’agent a_1 participe dans une des coalitions c_1 ou c_3 , la coalition c_2 devient impossible à réaliser. On dit qu’il y a une externalité d’existence entre c_3 et c_2 (et entre c_1 et c_2).

Il existe plusieurs scénarios réels où la formation de coalitions multi-agents peut être soumise à des contraintes de type externalités d’existence comme décrit dans l’exemple 3.2. CFDEP se distingue des travaux existants dans la littérature sur plusieurs aspects. Nous présentons dans la suite de ce chapitre, d’une part, les spécificités de CFDEP à travers deux cas d’étude et d’autre part, une comparaison et un positionnement comparatif avec les approches existantes.

3.3 Formation de coalitions avec externalités dynamiques : cas d’étude

Nous consacrons cette section à l’introduction d’un cas d’application que nous appelons le *Multi-alternative Agent Traveling Problem* (MATP) pour illustrer le problème de formation de coalitions avec externalités dynamiques. MATP est une abstraction d’une série de problèmes dans le monde réel, où il existe un ensemble d’agents compétitifs qui possèdent, chacun, un jeu d’alternatives formées de tâches interdépendantes et dont l’objectif est de déterminer quelle alternative est la mieux indiquée parmi l’ensemble de toutes les alternatives en fonction des autres agents. Nous étudions ce cas d’application dans deux contextes, à savoir, le contexte distribué (MATP) et le contexte centralisé (CMATP – *Centralized Multi-alternative Agent Traveling Problem* –). Nous développons ces deux cas d’étude tout au long du reste de cette thèse pour présenter nos approches de résolution.

3.3.1 Contexte distribué : MATP

Considérons le scénario suivant : supposons qu'il existe quatre individus représentés par quatre agents compétitifs $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$: un conférencier appelé *Tom* (a_1), une étudiante *Lisa* (a_2), un homme d'affaires *Mark* (a_3) et une touriste *Alice* (a_4). Chaque individu (agent) a_i veut voyager à partir d'une ville de départ $v_i^s \in \{a, b, c\}$ vers une ville de destination $v_i^d \in \{j, k\}$ en visitant différentes villes intermédiaires $v_i \in \{d, e, f, g, m, i, h\}$ une fois chacune, au plus (*cf.* Fig. 3.5). Les individus peuvent avoir différentes villes de départ, de destination et intermédiaires. En effet, chaque individu peut avoir différents chemins possibles (**path**) pour atteindre sa destination et peut voyager en utilisant différents moyens de transport collectifs/individuels (location de voiture, taxi, covoiturage, bus, train, avion ...etc.). La figure Fig. 3.5 illustre la carte des possibilités de déplacement de chaque individu. Le tableau Tab. 3.3 énumère les chemins alternatifs que chaque individu peut choisir pour atteindre son objectif.

Nous supposons que chaque individu a_i encourt un coût $\phi_i(\mathbf{path})$ lorsqu'il se déplace selon un chemin **path**. Nous supposons aussi que les déplacements effectués en commun par les individus en utilisant des moyens de transport partagés peuvent être plus avantageux que les déplacements en solitaire. Par conséquent, les individus peuvent être intéressés par des déplacements en groupe sur certains segments de leur parcours (**path**). Nous appelons un déplacement d'une ville v à une ville v' , représenté par un segment $v \rightarrow v'$, une tâche t . Pour exécuter t (c'est-à-dire se déplacer de v à v'), a_i encourt un coût $\phi_i(t) = \frac{\phi^{\text{pub}}(t)}{n} + \phi_i^{\text{priv}}(t)$, avec $\phi^{\text{pub}}(t)$ qui représente le *coût public* de la tâche t qui est connu de tous les individus et $\phi_i^{\text{priv}}(t)$ est le *coût privé* et qui n'est connu que par l'individu a_i . Nous supposons que, si n individus se déplacent conjointement sur un segment commun, ils en partageront le coût public, mais pas le coût privé. Le tableau Tab. 3.2 illustre les coûts des tâches pour chaque individu.

Prenons le cas de Tom (a_1) dont le but est de se déplacer d'une ville a à une ville j . Tom vise à trouver un chemin qui minimise ses coûts de déplacement. En considérant les deux tableaux Tab. 3.1 et Tab. 3.2, il peut choisir le chemin $\mathbf{path}_{1,1}$ pour un coût global de 115, le chemin $\mathbf{path}_{1,2}$ pour un coût de 103,33 et le chemin

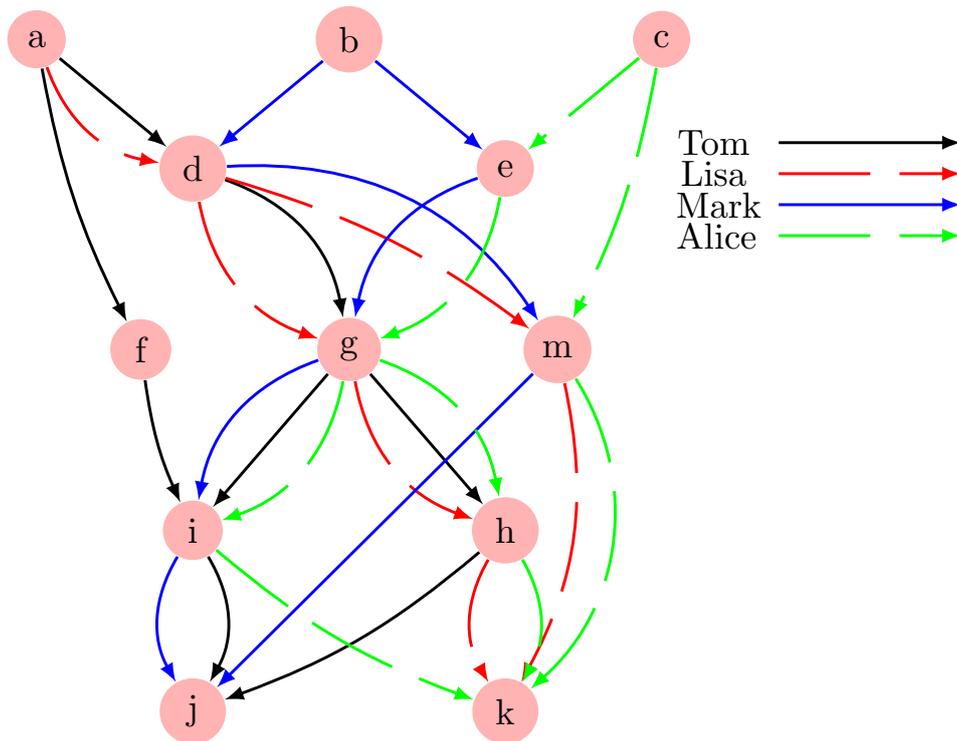


Fig. 3.5 – La carte des chemins possibles de chaque individu.

$\text{path}_{1,3}$ pour un coût de 116,66. Les déplacements sur les segments de chaque path peuvent se faire en groupe. Par exemple, Tom peut se déplacer suivant le schéma : $a \rightarrow f \rightarrow i$ seul, puis $i \rightarrow j$ avec Mark ; ou : $a \rightarrow d \rightarrow g$ conjointement avec Lisa, puis $g \rightarrow h$ avec Lisa et Alice, puis $h \rightarrow j$ seul ; ou : $a \rightarrow d \rightarrow g$ avec Lisa, puis $g \rightarrow i$ avec Mark et Alice, et $i \rightarrow j$ avec Mark. Une question centrale à considérer par Tom est de savoir quels segments il devrait partager et avec qui il devrait le faire en tenant compte des coûts des tâches et des interdépendances ?

Agent	Chemin
Tom : a_1	$path_{1,1} : a \rightarrow f \rightarrow i \rightarrow j$
	$path_{1,2} : a \rightarrow d \rightarrow g \rightarrow i \rightarrow j$
	$path_{1,3} : a \rightarrow d \rightarrow g \rightarrow h \rightarrow j$
Lisa : a_2	$path_{2,1} : a \rightarrow d \rightarrow g \rightarrow h \rightarrow k$
	$path_{2,2} : a \rightarrow d \rightarrow m \rightarrow k$
Mark : a_3	$path_{3,1} : b \rightarrow d \rightarrow m \rightarrow j$
	$path_{3,2} : b \rightarrow e \rightarrow g \rightarrow i \rightarrow j$
Alice : a_4	$path_{4,1} : c \rightarrow e \rightarrow g \rightarrow i \rightarrow k$
	$path_{4,2} : c \rightarrow e \rightarrow g \rightarrow h \rightarrow k$
	$path_{4,3} : c \rightarrow m \rightarrow k$

Tab. 3.1 – Les chemins possibles pour chaque individu.

Il est tout à fait naturel que les agents adoptent une approche individualiste pour atteindre leurs destinations. Cela serait dicté par leur nature d'agents compétitifs qui fait qu'ils voudront suivre le chemin le moins coûteux. Cependant, si ces agents agissent seuls dans leurs prises de décisions, sans interactions et sans prise en compte des interdépendances qui existent entre les segments constituant leurs chemins, selon les méthodes de formation de coalitions existantes, ils voudront certainement (chacun d'entre eux) commencer par le premier segment le moins coûteux, puis à chaque fin de segment, décider à nouveau sur le prochain segment à faire, et répéter cette démarche jusqu'au dernier segment. Les approches classiques de formation de coalitions ne tiennent pas compte du fait que les tâches sont séquentielles et qu'elles font partie d'un jeu d'alternatives possibles [74, 69, 73, 77, 78, 8, 4, 60, 62, 92, 57, 63, 64, 61].

En adoptant une telle approche, au moment de la prise de décision sur un segment, les individus ne possèdent pas de projection sur le reste des chemins et leurs segments. Ainsi, suivant cette approche, la première décision de Tom (quand il est à la ville \mathbf{a}) serait de sélectionner le segment $\mathbf{a} \rightarrow \mathbf{f}$ (avec un coût de 20) plutôt que $\mathbf{a} \rightarrow \mathbf{d}$ (avec un coût d'au moins 45). Cependant, cette approche conduirait Tom à un cheminement coûteux et sous-optimal, à savoir $path_{1,1}$, dont le coût

total est d'au moins $20 + 80 + 15 + 15 = 115$. Ainsi, Tom pourra rater des chemins plus attractifs (par exemple, $\text{path}_{1,2}$).

Pour remédier à cela, les individus doivent adopter une approche qui ne se focalise pas uniquement sur le coût du segment en cours (ou d'un seul segment), mais qui tient compte de la totalité des segments possibles de tous les chemins possibles. Les décisions doivent être prises avec une projection sur le restant des segments possibles. Suivant cette approche, Tom préférerait $\text{path}_{1,2}$, ce qui pourrait être plus avantageux (avec un coût de 103,33, comparativement à 115 pour $\text{path}_{1,1}$). Cependant, ce coût inférieur n'est possible que s'il obtient les accords de Lisa, Mark et Alice pour se déplacer conjointement sur les segments partagés du chemin $\text{path}_{1,2}$. Néanmoins, eux aussi peuvent avoir d'autres chemins qu'ils préféreraient. Par exemple, Lisa peut préférer $\text{path}_{2,2}$ qui peut lui coûter 92,5 (moins de $\text{path}_{2,1}$ qui peut lui coûter au moins 94,16) si tous les individus concernés acceptent ce chemin. Ce dernier cheminement ne peut être possible que si Mark l'accepte. Mark peut cependant préférer $\text{path}_{3,2}$ qui lui coûte au moins 88,33 plutôt que 117,5 pour le chemin que Lisa préfère. Alice peut préférer $\text{path}_{4,2}$ qui lui coûte 99,16, moins que le coût des chemins $\text{path}_{4,1}$ et $\text{path}_{4,3}$ ¹.

Ce scénario démontre que les individus ont plusieurs chemins pour atteindre leurs destinations et leurs préférences sur ces chemins peuvent être conflictuelles avec celles des individus avec lesquels ils coopèrent. Par conséquent, il est difficile de satisfaire les préférences de tous les individus. Par exemple, concentrons-nous à nouveau sur Tom ; s'il décide de se déplacer suivant le schéma : $\mathbf{a} \rightarrow \mathbf{d} \rightarrow \mathbf{g}$ avec Lisa et $\mathbf{g} \rightarrow \mathbf{h}$ avec Lisa et Alice ensemble, le segment $\mathbf{g} \rightarrow \mathbf{i} \rightarrow \mathbf{j}$ devient impossible, laissant $\mathbf{h} \rightarrow \mathbf{j}$ comme l'unique segment pour Tom d'atteindre son but. Cependant, $\mathbf{h} \rightarrow \mathbf{j}$ est un segment coûteux. Ainsi, étant donné les chemins d'autres individus, Tom peut préférer un chemin qui ne le force pas à choisir $\mathbf{h} \rightarrow \mathbf{j}$ ou $\mathbf{f} \rightarrow \mathbf{i}$, qui sont des segments coûteux. Par conséquent, avant de décider de $\mathbf{g} \rightarrow \mathbf{h}$, Tom peut chercher à obtenir l'engagement de Mark et Alice de se déplacer conjointement sur le segment $\mathbf{g} \rightarrow \mathbf{i}$ (un segment moins coûteux). Le problème est que Mark et Alice peuvent avoir d'autres possibilités pour lesquelles

1. Le coût de $\text{path}_{4,1}$ est de 103,33 et celui de $\text{path}_{4,3}$ est de 110.

–	Coût public	Coût privé			
		Tom	Lisa	Mark	Alice
$t_1 : a \rightarrow d$	20	35	35	–	–
$t_2 : a \rightarrow f$	15	5	–	–	–
$t_3 : d \rightarrow g$	20	10	5	–	–
$t_4 : g \rightarrow h$	20	10	10	–	10
$t_5 : g \rightarrow i$	40	10	–	15	15
$t_6 : h \rightarrow j$	30	5	–	–	–
$t_7 : i \rightarrow j$	30	0	–	5	–
$t_8 : h \rightarrow k$	35	–	0	–	5
$t_9 : d \rightarrow m$	45	–	10	25	–
$t_{10} : m \rightarrow k$	30	–	0	–	65
$t_{11} : b \rightarrow e$	10	–	–	5	–
$t_{12} : e \rightarrow g$	20	–	–	15	15
$t_{13} : b \rightarrow d$	30	–	–	5	–
$t_{14} : m \rightarrow j$	30	–	–	5	–
$t_{15} : c \rightarrow e$	30	–	–	–	5
$t_{16} : c \rightarrow m$	25	–	–	–	5
$t_{17} : i \rightarrow k$	10	–	–	–	5
$t_{18} : f \rightarrow i$	55	25	–	–	–

Tab. 3.2 – Les coûts des segments des chemins.

ils cherchent les engagements des autres individus et ainsi de suite. En fait, chaque décision de parcourir collectivement un segment de trajet peut avoir un impact sur le reste des segments des trajets possibles. En effet, une approche de décision par segment unique est inappropriée dans de telles situations. Une approche avec une vue d'ensemble sur tous les chemins possibles peut être plus appropriée pour tous.

Donc, MATP dans le contexte distribué peut être formulé comme suit : étant donné un ensemble d'agents, chacun cherchant à atteindre sa destination tout en réduisant son coût total, quels comportements ces agents devraient adopter

pour parvenir à une solution communément acceptée (ou éventuellement plusieurs solutions disjointes), tout en respectant d'une part, les dépendances qui existent entre les tâches dans leurs alternatives respectives et d'autre part, les préférences des agents. De plus, les agents disposent d'informations incomplètes les uns sur les autres et des fonctions privées d'évaluations des tâches, qui sont inconnus par les autres aussi. Ici, les agents recherchent non seulement la coordination entre eux pour accomplir leurs tâches, mais aussi l'accord sur la réalisation conjointe de ces tâches. Néanmoins, il n'y a aucune garantie que des accords soient conclus.

3.3.2 Contexte centralisé : CMATP

Dans cette section, nous considérons une version centralisée du problème MATP (*c.f.* 3.3.1), qu'on appelle CMATP (Centralized MATP), et qu'on décrit par le scénario présenté dans le chapitre 2 avec des adaptations qu'on présente ci-dessous.

Soit 4 agents compétitifs, représentant 4 différents individus (Tom, Lisa, Mark et Alice), qui veulent chacun se déplacer depuis leur ville de départ $\mathbf{v}^s \in \{\mathbf{a}, \mathbf{g}\}$ vers une ville d'arrivée $\mathbf{v}^d \in \{\mathbf{j}, \mathbf{h}, \mathbf{k}\}$ en visitant différentes villes intermédiaires possibles $\mathbf{v} \in \{\mathbf{g}, \mathbf{i}, \mathbf{h}, \mathbf{m}\}$, une fois au plus pour chacune d'elles. La figure Fig. 3.6 illustre les plans possibles des déplacements des 4 individus. Les nœuds représentent les villes et les arcs représentent les déplacements possibles.

Les déplacements qui mènent un individu d'une ville de départ vers une ville de destination forment un chemin. Chaque individu peut avoir différents chemins possibles pour atteindre sa destination. Un déplacement d'une ville vers une autre peut se faire par différents moyens de transport collectifs/individuels (location de voiture, taxi, covoiturage, bus, train, avion...etc.). Le tableau Tab. 3.3 illustre les chemins possibles de chaque individu. Chaque déplacement d'un individu est possible individuellement et engendre un coût public (commun aux individus) et un coût privé (propre à chaque individu). Il est donc, plus avantageux pour chacun de partager les coûts publics avec les autres quand il est convenable de le faire.

Le problème CMATP peut être formulé comme suit : comment identifier les meilleurs déplacements permettant aux individus d'atteindre leur ville de destination avec un minimum de coût, en identifiant les coalitions (groupes d'individus)

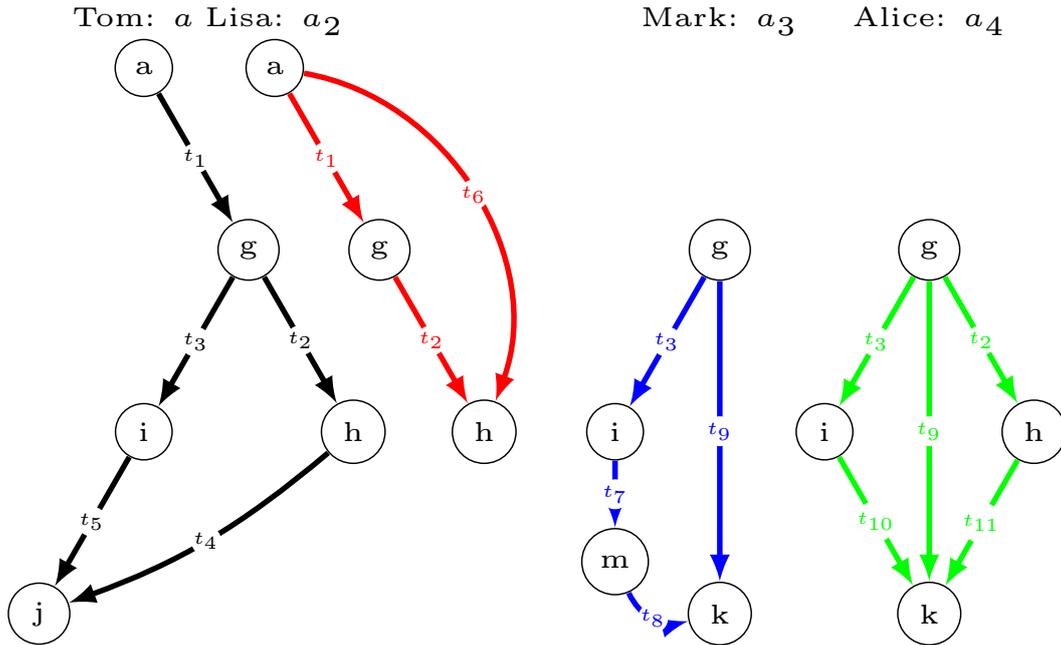


Fig. 3.6 – Les possibilités de déplacements des individus.

Agent	Chemin
Tom : a_1	$a \rightarrow g \rightarrow i \rightarrow j$
	$a \rightarrow g \rightarrow h \rightarrow j$
Lisa : a_2	$a \rightarrow g \rightarrow h$
	$a \rightarrow h$
Mark : a_3	$g \rightarrow i \rightarrow m \rightarrow k$
	$g \rightarrow k$
Alice : a_4	$g \rightarrow i \rightarrow k$
	$g \rightarrow k$
	$g \rightarrow h \rightarrow k$

Tab. 3.3 – Les chemins individuels des agents.

pour partager le coût public de chaque déplacement entre deux villes? Le défi ici est d'éviter les calculs redondants. Par exemple, la génération multiple de toute coalition irréalisable à cause des contraintes. De plus, il est essentiel de considérer

la façon dont les coalitions réalisables peuvent être combinées pour générer efficacement des structures de coalitions valides. En effet, il peut être coûteux de passer par toutes les combinaisons possibles de coalitions et de vérifier si elles contiennent des coalitions conflictuelles.

3.4 CFDEP et les approches de formation de coalitions

Dans CFDEP, l'objectif de chaque agent est d'atteindre son but à un coût minimum. En adoptant une attitude coopérative, les agents peuvent tirer des profits en se groupant dans des coalitions. Notre étude considère un problème spécifique de formation de coalitions multi-agents avec des externalités d'existence qui changent dynamiquement. Les agents visent à former une série de coalitions choisies parmi plusieurs alternatives sur des tâches séquentielles et interdépendantes, afin d'atteindre leurs objectifs individuels et de maximiser leurs gains. Les agents sont rationnels avec des limitations sur leurs coûts prévus, *i.e.*, ils ont chacun des coûts minimaux et maximaux qu'ils ne peuvent pas dépasser, se référant respectivement aux coûts de l'alternative la plus bénéfique et celui de la moins bénéfique. Ces deux bornes limitent le gain qu'un agent peut en tirer de la réalisation de ses tâches. Notons que le fait qu'un agent puisse former plusieurs coalitions séquentielles est une différence majeure entre CFDEP et les problèmes de formation de coalitions existants dans la littérature.

3.4.1 Formation de coalitions avec externalités

La première étude computationnelle des jeux coalitionnels avec externalités dans un contexte multi-agents a été présentée par Rahwan et al. [61]. Par ailleurs, dans la littérature, très peu d'attention a été accordée à ce type de problèmes [51, 61, 81, 77, 4, 44, 98, 64, 6, 14, 102]. Dans des configurations de formation de coalitions avec des agents compétitifs, utiliser des informations dérivées pour en déduire les intentions des agents s'avère efficace. Aknine et al. [4] examinent les

relations entre les tâches pour parvenir à des accords en dérivant les intentions des agents. L'objectif étant de réduire le temps de calcul, les auteurs proposent un mécanisme de formation de coalitions basé sur 4 principes :

- L'utilisation des informations sur les relations entre les tâches afin de réduire le temps de calcul des coalitions ;
- L'exploitation des propositions de coalitions formulées par les agents afin de déduire leurs intentions. Cela passe tout d'abord par une représentation expressive des informations véhiculées par ces propositions, qu'elles soient implicites ou explicites ;
- Faire propager les propositions des agents en utilisant plusieurs stratégies afin de renforcer leurs choix ;
- La réorganisation des coalitions précédentes lorsque des changements se produisent dans le contexte de l'exécution ou lorsque des événements inattendus (liés aux coalitions précédentes) se produisent.

Ainsi, le processus de formation de coalitions se déroule en deux phases :

- Une phase d'initialisation où les agents s'échangent leurs tâches et leurs priorités ;
- Une phase de négociation où les agents formulent et s'échangent leurs propositions. Ces échanges peuvent se faire selon trois stratégies : i) en envoyant les propositions à tous les agents, d'une manière séquentielle, ii) en envoyant la même proposition à différents agents au même moment, et iii) en envoyant différentes propositions à différents agents au même moment.

La formation de coalitions commence quand un agent initiateur envoie à un autre agent ses propositions préférées (une proposition est considérée comme préférée si l'agent a un degré de préférence concernant cette proposition, supérieure à une valeur de référence). L'agent initiateur peut ensuite envoyer de manière itérative, par ordre décroissant de préférence, ses autres propositions, jusqu'à ce qu'il n'y ait plus de propositions au moins équivalentes à celle de référence. Cependant, avant d'envoyer une proposition de préférence inférieure à la préférence de la dernière proposition envoyée, l'agent initiateur peut attendre de recevoir un message de l'agent destinataire de sa proposition : soit une réponse à sa proposition pré-

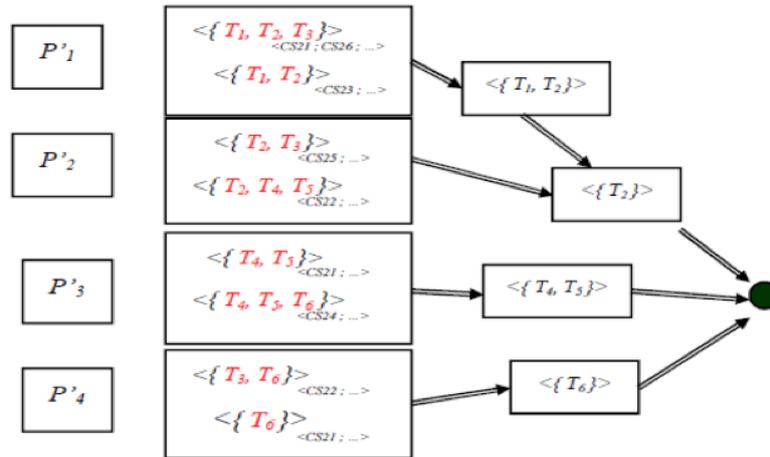


Fig. 3.7 – Représentation graphique des intentions des agents.

cédente, soit une nouvelle proposition de cet agent. Les propositions reçues sont utilisées pour construire le graphe d'intentions de ces agents (*cf.* Fig. 3.7). Chacun des autres agents calcule également ses propositions préférées. L'initiateur calcule ensuite la deuxième proposition préférée et ainsi de suite.

Lorsqu'un agent reçoit un groupe de propositions d'un agent émetteur, il ne considère que les propositions au moins équivalentes à sa proposition de référence. S'il existe au moins une proposition qui est préférable ou équivalente à sa meilleure proposition, il transmet cette proposition à l'agent suivant qu'il souhaite inclure dans la négociation. Si un agent trouve des propositions, dans le groupe reçu, inacceptables pour lui, il les déclare aux autres agents. Lorsqu'un agent reçoit un groupe de propositions approuvées par tous les agents concernés dans ce groupe, il peut mettre fin à la négociation et envoyer aux concernés cette proposition, dite la *solution*.

Dans certains cas d'applications où les agents doivent intervenir dans un environnement réel, il est possible qu'ils soient confrontés à des contraintes spatiales et temporelles pour la réalisation de leurs tâches. Dans ce contexte, Ramchurn et al. [64], traitent le problème de la formation de coalitions sous des contraintes spatiales et temporelles, dénoté CFSTP. Ils proposent une approche de formation de coalitions pour la coordination d'un ensemble d'intervenants et des robots pour

la réalisation des tâches dans des scénarios de gestion de crise. Dans le problème exposé, les auteurs supposent l'existence d'un ensemble de contraintes de type spatiales et temporelles auxquelles les agents doivent faire attention pour garantir le bon déroulement d'une opération de gestion de crise. Dans de telles situations, il y a généralement plus de tâches à réaliser que d'agents. Les auteurs considèrent un ensemble de tâches distribuées géographiquement sur le périmètre de la zone d'une crise et un ensemble d'agents qui ont des positions différentes sur cette zone et qui doivent tenir compte du temps nécessaire pour passer d'une tâche à une autre (d'une position à une autre). Les coalitions formées doivent être dissoutes au bout d'un certain temps et peuvent être reformées dans d'autres zones du périmètre de la gestion de crise. Les auteurs traitent ici des contraintes spatiales. Les tâches peuvent être très complexes ce qui fait que la période de temps que les agents peuvent leur consacrer peut varier d'un agent à un autre. De plus, un agent est tenu de respecter le temps d'intervention dans une tâche. Cela donne lieu à des contraintes temporelles. Par ailleurs, chaque tâche est associée à une charge de travail et à une date limite de sorte que la tâche est considérée comme terminée si et seulement si la quantité de travail requise est effectuée avant le seuil de temps fixé. Si plusieurs agents travaillent simultanément sur une tâche, leur contribution par intervalle de temps peut ne pas être nécessairement additive. Ainsi, alors que chacun des agents i et j , à eux deux, peuvent réaliser une quantité de travail x dans un intervalle de temps, ils peuvent produire $y > 2x$ ($y < 2x$ dans le cas des domaines sous-additifs), s'ils collaborent et travaillent sur la tâche en même temps. Dans [64], les auteurs proposent une formalisation du problème CFSTP et montrent qu'il est NP-difficile. Ils présentent comment le résoudre d'une manière optimale ou approximative en utilisant l'heuristique MIP (Mixed Integer Programming) pour l'allocation de groupes de robots à des tâches nécessitant des équipes ayant des capacités spécifiques et une heuristique d'ordonnement.

Rahwan et al. [91] présentent un framework CCF (Constrained Coalition Formation) dans lequel ils traitent le problème de formation de coalitions avec des contraintes inhérentes aux coalitions réalisables. Les contraintes présentées précisent quels groupes d'agents devraient ou ne devraient pas travailler ensemble.

En soulignant l'importance de ne tenir compte que des coalitions réalisables, les auteurs proposent une procédure qui transforme ces contraintes en une structure d'entrée qui permet aux algorithmes de formation de coalitions d'identifier toutes les coalitions qui respectent les contraintes pour éviter les calculs redondants. Ils utilisent ensuite cette procédure pour développer un algorithme permettant de générer une structure de coalitions optimale (maximisant le bien-être) sous contraintes.

L'idée de base de l'approche est la suivante : pour une optimisation de la recherche d'une structure de coalitions optimale, il faut se focaliser sur les coalitions faisables uniquement et éviter de calculer les coalitions qui ne sont pas faisables. Les auteurs définissent des contraintes positives et des contraintes négatives, formulées avec la logique propositionnelle. Alors, pour qu'une coalition soit faisable, il faut qu'elle satisfasse toutes les contraintes négatives, et au moins une seule contrainte positive. En effet, on ne peut pas imposer la satisfaction de toutes les contraintes positives, sinon, ça reviendrait à former la grande coalition. Contrairement aux contraintes négatives (qui doivent être toutes satisfaites). L'algorithme proposé pour la génération des coalitions faisables est un algorithme basé sur le principe de diviser pour régner (Divide and Conquer). Il procède par division récursive de l'ensemble des coalitions en deux sous-ensembles : i) le sous-ensemble des coalitions qui contiennent un agent a_i et ii) le sous-ensemble des coalitions qui ne contiennent pas un agent a_i . Ce processus se répète de manière récursive jusqu'à obtention des sous-ensembles indivisibles. Ces sous-ensembles sont ensuite combinés pour former à la fin la solution au problème globale.

Arib et al. [6] s'intéressent aux situations où les agents planifient leurs activités de façon dynamique et utilisent des plans pour coordonner leurs actions et rechercher les coalitions à former. Les auteurs utilisent les plans des agents pour guider la recherche des coalitions à former. Pour cela, ils présentent de nouveaux concepts, dont l'un est la désirabilité d'action qui donne une mesure sur le degré de l'importance d'une action dans le plan des agents. Ils soulignent la pertinence de ne pas prendre en compte seulement les actions immédiates des agents dans le processus de formation de coalitions et que l'analyse des relations entre les actions

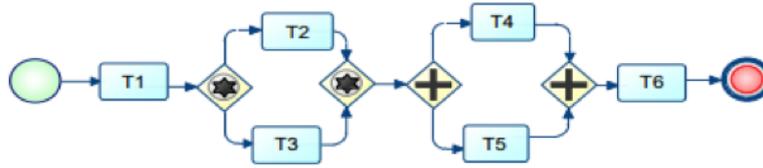


Fig. 3.8 – Modèle BPMN de tâches.

peut améliorer la formation de coalitions. En effet, l'existence de relations entre les actions (par exemple des dépendances) engendre des dépendances entre les plans des agents qui les exécutent. Dans ces situations, il est important que les agents identifient ces dépendances afin de faciliter l'obtention de compromis entre eux.

Dans un contexte distribué, Jian et al. [14] considèrent des agents compétitifs qui doivent collaborer pour atteindre un seul but global réalisable à travers un ensemble de tâches indépendantes. Les agents disposent d'un plan global commun à tous et les tâches peuvent être complexes et composées de plusieurs sous-tâches. Les coûts de réalisation des tâches sont composés d'une partie publique connue par tous les agents et d'une partie privée propre à chaque agent et inconnue par les autres. En adoptant une représentation des relations entre les tâches par un modèle de workflow (BPMN) (*cf.* Fig. 3.8), les auteurs proposent un algorithme où, dans un premier temps, chaque agent forme une proposition (un plan) de la manière optimale pour lui. Ensuite, un agent *publisher* est sélectionné aléatoirement et diffuse sa proposition. Les agents alors, envoient un *feedback* concernant cette proposition et pour chaque tâche du plan et qui peut être soit une acceptation, un refus, ou une concession. La concession étant une divulgation graduelle et contrôlée par un facteur de divulgation, sur le coût privé des tâches. Après réception du *feedback*, l'agent *publisher* met à jour sa table des coûts des agents. Au prochain rond, un autre agent est sélectionné aléatoirement pour jouer le rôle de *publisher* et publier son plan candidat.

Dans Zhang et al. [102], les auteurs considèrent un problème d'allocation de tâches, avec des robots qui peuvent participer dans l'exécution d'une tâche à la fois, avec un système d'affectation de tâches instantané. La réalisation d'une tâche donnée nécessite des ressources possédant des dépendances inter-ressources. Le

processus d’allocation de tâches doit donc tenir compte de ces dépendances qui engendrent d’une manière indirecte des dépendances entre les tâches. Les auteurs proposent une solution à base d’une heuristique qui tient compte à la fois du gain et de la perte d’utilité. Ainsi, pour la prise en considération des dépendances et relations de précédence entre les tâches, ils proposent un algorithme très semblable à celui de Shehory et al. [77]. La différence réside dans les paramètres à prendre en considération, comme la mesure de l’utilité et les vecteurs de capacités nécessaires pour la réalisation des tâches.

Michalak et al. [51] ne considèrent que la question de la représentation des jeux coalitionnels dans les systèmes multi-agents qui présentent des externalités dues à la formation de coalitions. Ils proposent une nouvelle représentation basée sur des expressions booléennes. L’objectif est de construire des expressions plus riches qui permettent de capturer les externalités induites sur les coalitions.

3.4.2 Formation de coalitions avec chevauchements

Quelques travaux de recherche ont porté sur le problème de formation de coalitions avec chevauchements (*OCF – Overlapping Coalitions Formation –*) [76, 79, 20, 15, 2, 103, 100]. Dans ce type de problèmes, les agents impliqués dans les coalitions peuvent participer simultanément dans plusieurs coalitions [103]. Fondamentalement, CFDEP est basé sur la même idée. Néanmoins, il y a une différence majeure entre CFDEP et le problème de formation de coalitions avec chevauchement. Dans des travaux comme [76, 79, 20, 15, 2, 103, 100], les agents impliqués dans les coalitions exécutent leurs plans de manière indépendante. Alors que, dans CFDEP, nous nous concentrons sur l’exécution de plans de coalitions interdépendantes. De plus, même si dans CFDEP les coalitions d’un agent sont exécutées de manière séquentielle, leur formation est simultanée. C’est-à-dire, aucune coalition n’est exécutée avant la fin du processus de la formation de coalitions, contrairement aux travaux existants pour le problème OCF.

Dans un cadre coopératif distribué, Shehory et al. [76, 79] proposent des algorithmes à base d’heuristiques qui conduisent à la création de structures de coalitions avec chevauchement pour la réalisation de tâches avec un ordre de précé-

dence. Dang et al. [20] utilisent également des algorithmes à base d'heuristiques, mais dans le domaine des réseaux de surveillance multi-capteurs.

Dans [15], les auteurs introduisent un modèle de jeux coopératifs avec des coalitions avec chevauchement et ont exploré la question de la stabilité dans ce contexte. Ils ont introduit un concept d'équilibre pour les jeux coalitionnels avec chevauchement et l'ont utilisé pour caractériser les structures de coalitions qui peuvent être étendues aux éléments du noyau (core).

Yair et al. [103] s'intéressent également à la notion de stabilité et proposent un framework unifié pour l'étude de la stabilité dans le cadre de la formation de coalitions avec chevauchement. Leur approche est basée sur la notion de fonctions d'arbitrage. Ces fonctions déterminent le gain obtenu par les déviateurs (les agents qui souhaitent former une autre coalition que celle dans laquelle ils sont déjà impliqués) compte tenu de leur déviation et de l'allocation actuelle des ressources.

Les travaux de [100] introduisent un nouveau framework mathématique à partir des jeux coopératifs pour modéliser et résoudre des scénarios coopératifs où les périphériques réseaux doivent devenir plus autonomes et coopérer entre eux dans deux classes principales de jeux OCF, à savoir : la gestion des interférences et la détection coopérative du spectre. De tels mécanismes de coopération impliquent le partage et la distribution simultanée des ressources entre un certain nombre de groupes ou coalitions coopératifs avec chevauchements.

3.4.3 Formation de coalitions à base de graphes

Dans une tout autre approche, plusieurs algorithmes de formation de coalitions à base de graphes ont été proposés [17, 36, 29]. Dans Georgios et al. [17], les auteurs traitent le problème de la formation de coalitions dans un environnement à contraintes, en particulier, dans le cas où les agents sont munis de graphes d'interaction. Un graphe d'interaction détermine l'ensemble de toutes les coalitions possibles. Ainsi, une coalition c ne peut être formée que si le sous-graphe induit sur les noeuds/agents dans c est connecté.

Dans la même perspective, Gianluigi et al. [36] proposent un modèle de génération de structures de coalitions où des contraintes de deux types différents

peuvent être exprimées simultanément. Le modèle est basé sur le concept d’évaluation d’une structure qui consiste en un ensemble d’agents pivots incompatibles par paires plus un graphe d’interaction prescrivant qu’une coalition c ne peut être formée que si le sous-graphe induit sur les noeuds/agents en c est connecté.

Dans Alessandro et al. [29], les auteurs proposent une nouvelle heuristique basée sur les idées et les outils utilisés dans le domaine des clusters de données. Ils présentent un algorithme de formation de coalitions inspiré des techniques d’agglomération hiérarchique, dans différentes variantes de l’algorithme, qu’ils appellent Coalition Linkage (C-Link). Ils y démontrent comment un tel algorithme peut être adapté aux problèmes de formation de coalitions restreintes aux graphes. Un graphe d’interaction défini entre les agents limite l’ensemble des coalitions réalisables.

3.4.4 Formation de coalitions multi-robots

Certains travaux ont traité le problème de formation de coalitions avec des relations de précédence entre les tâches dans différents domaines autres que celui des systèmes multi-agents. Par exemple, dans des environnements multi-robots, Lovekesh et al. [98] prennent en compte l’ordre de précédence entre les tâches qu’un groupe de robots coopératifs doit réaliser. Yu et al. [102] ont adopté la même approche que celle décrite dans [77] avec quelques adaptations pour gérer l’attribution des tâches pour les robots.

Par ailleurs, il existe des travaux où le problème de la formation de coalitions est traité comme un problème d’optimisation de la fonction d’utilité pour un ensemble d’agents indépendants. Ce type de problèmes est plus généralement connu sous l’appellation : problèmes d’optimisation de contraintes distribuées (DCOP – *distributed constraint optimization problems*). Cependant, les problèmes dans DCOP sont fondamentalement différents de CFDEP. Les DCOPs supposent que les agents sont coopératifs et tentent de trouver conjointement la solution qui maximise la fonction d’utilité globale [53]. En fait, il n’est pas possible d’appliquer les algorithmes DCOP dans le cas de CFDEP où les agents sont compétitifs avec différentes fonctions d’utilités privées et non linéaires.

Le tableau 3.4 illustre une synthèse comparative entre CFDEP et les principaux travaux exposés dans cette thèse, selon différents critères, qui sont :

- **Dépendances** : qui indique l'existence de dépendances entre les tâches (✓) ou pas (-) ;
- **Tâches par coalition** : qui indique le nombre de tâches d'une même coalition ;
- **Alternatives** : qui indique si un agent dispose d'une seule alternative (1), ou de plusieurs (**n**). En d'autres termes, si l'ensemble des tâches pour lesquelles un agent souhaite former des coalitions est fixe tout au long du processus de formation de coalitions ou variable ;
- **Externalités** : qui indique l'existence d'externalités (✓) ou pas (-) (quelle que soit leur nature) ;
- **Négociations** : qui indique si les agents utilisent (✓) des mécanismes de négociations pour la formation de coalitions.
- **Négociations disjointes** : qui indique, si les agents procèdent par négociations, si ces négociations se déroulent entre les agents au sein d'un seul groupe (1), ou en plusieurs groupes disjoints (**n**). Donc un agent peut avoir différentes négociations avec différents groupes d'agents.
- **Chevauchement** : qui indique s'il existe (✓) des chevauchements entre les coalitions ou pas (-). C'est-à-dire, si un agent donné, peut être présent dans plus d'une coalition à la fois.
- **GSC** : Si les agents utilisent des techniques de GSC – Génération de Structures de Coalitions – (✓) dans leur processus de formation de coalitions ou pas (-) ;
- **Algorithme Any-time** : Si l'algorithme de résolution a la caractéristique dite *Any-time* signifiant qu'il peut retourner une solution à tout moment du processus de formation de coalitions et que cette solution s'améliore continuellement dans le temps ;
- **Distribué / Centralisé** : qui indique si l'algorithme proposé est distribué (D) ou centralisé (C).

On remarque que toutes ces études ne traitent pas l'existence de dépendances

entre coalitions dans le cas où plusieurs alternatives sont possibles pour les agents ; elles ne conviennent donc pas au CFDEP. Comme on peut le constater sur le tableau Tab. 3.4, CFDEP est différent des autres problèmes sur plusieurs aspects. Néanmoins, il existe quelques aspects qui peuvent être similaires sous certains points de vue. Dans ce qui suit nous présentons quelques travaux et les différences/ressemblances qui existent par rapport à CFDEP.

3.5 CFDEP et quelques problèmes multi-agents

3.5.1 Problème de négociation multi-agents

Le problème de négociation multi-agents a reçu beaucoup d'attention tant dans le domaine de la formation de coalitions que dans d'autres domaines [5, 65, 84, 42]. En effet, il est utile de rappeler que la négociation est un mécanisme d'interaction qui permet à un groupe d'agents d'interagir afin d'atteindre un certain objectif, en particulier dans des situations où les agents ont plusieurs possibilités à explorer.

Bo et al. [5] considèrent des négociations inter-reliées où des agents compétitifs doivent coordonner efficacement leurs négociations avec de multiples fournisseurs de ressources pour acquérir de nouvelles ressources nécessaires à l'accomplissement d'une tâche de haut niveau. Israël et al. [84] utilisent la négociation comme un processus d'exploration sur un ensemble d'alternatives possibles. Les auteurs considèrent des situations où les négociateurs n'ont aucune information sur la valeur des alternatives. Ainsi, leur objectif est de permettre à deux parties prenantes dans la négociation de choisir une alternative parmi plusieurs possibles. Par ailleurs, les auteurs supposent que la négociation est limitée dans le temps et qu'un facteur d'actualisation est utilisé pour faire converger la négociation vers un accord.

Un autre travail dans lequel les agents doivent explorer différentes alternatives est celui de Rochlin et al. [65]. Les agents peuvent identifier plusieurs alternatives (opportunités) qui s'offrent à eux, à la suite d'un processus d'exploration coûteux, sans révéler l'utilité de ces alternatives. Ainsi, l'objectif des agents n'est pas de maximiser le gain global, mais plutôt de maximiser leur propre gain.

Travaux	Dépendances	Tâches par coalition	Alternatives	Externalités	Négociations	Négociations disjointes	Chevauchement	GSC	Algorithme Any-time	Distribué / Centralisé (C)
[76]	✓	1	1	-	-	1	-	✓	✓	D
[79]	✓	1	1	-	-	1	✓	✓	✓	D
[77]	✓	1	1	-	-	1	✓	✓	✓	D
[4]	✓	$n \geq 1$	1	-	✓	1	-	✓	-	D
[20]	-	1	1	-	-	1	✓	✓	-	C
[44]	✓	1	1	-	-	1	✓	-	-	C
[98]	✓	1	1	-	-	1	✓	✓	-	D
[64]	-	1	1	✓	-	1	-	✓	-	C
[51]	✓	1	1	✓	-	1	-	✓	-	C
[2]	-	1	1	-	✓	1	✓	-	-	D
[15]	-	1	1	-	-	1	✓	✓	-	C
[6]	✓	1	1	-	✓	1	-	-	-	D
[61]	-	1	1	✓	-	1	-	✓	✓	C
[14]	✓	1	1	-	✓	1	-	-	-	D
[102]	✓	1	1	-	-	1	-	-	-	D
[103]	-	1	1	-	-	1	✓	✓	-	C
[100]	-	-	1	-	-	1	✓	-	-	C
[81]	-	1	1	✓	-	1	-	✓	-	C
[17]	-	-	1	-	-	1	-	✓	-	C
[36]	-	-	1	✓	-	1	-	✓	-	C
[29]	-	-	1	-	-	1	-	✓	✓	C
[82] CFDEP	✓	$n \geq 1$	$(n \geq 1)$	✓	✓	$n \geq 1$	✓	✓	-	D

Tab. 3.4 – Positionnement de CFDEP.

Dave et al. [22] proposent un nouvel algorithme de négociation à base d'une heuristique. Les agents sont considérés comme compétitifs avec des fonctions d'utilités non-linéaires. L'espace des accords possibles étant très large, une recherche exhaustive des meilleures propositions n'est pas envisageable. C'est pourquoi les auteurs proposent une heuristique et considèrent que les négociations sont limitées dans le temps. Byung-Kook [42] considère un problème d'arrêt optimal (*optimal stopping problem*) avec une probabilité de rappel incertaine où un coût de rappel doit être payé pour accepter une offre passée. Ainsi, le premier intérêt de cette approche est de trouver une règle de décision optimale pour arrêter la négociation, en permettant aux parties de décider quand arrêter la recherche de nouvelles offres et en acceptant l'une des offres apparues à l'instant t .

Compte tenu des travaux sur les négociations multi-agents, sur plusieurs alternatives, CFDEP peut être considéré comme un problème d'exploration avec un rappel incertain, comme proposé par [42]. Cependant, la différence avec les travaux de Byung-Kook [42] réside dans le fait qu'il y a un coût de rappel qui doit être assumé pour accepter une offre passée. Dans CFDEP, il n'y a pas de coût de rappel à assumer, mais la valeur peut diminuer ou augmenter en fonction des négociations d'autres agents disjoints.

3.5.2 Problème de planification multi-agents

L'utilisation de plans prédéfinis par nos agents rend la description de notre problème proche du problème communément connu : *la Planification Multi-Agent (PMA)*². Surtout dans le cas particulier où les agents s'engagent dans le processus de planification avec un plan initial individuel à exécuter. Les deux problèmes sont proches dans le sens où ils supposent que les agents ont en entrée un plan individuel et en sortie une exécution non conflictuelle de leurs tâches. Par exemple, Hans et al. [96] traitent le problème de la coordination comme un processus itératif de révision pour coordonner les plans des agents selon deux approches : i) par la fusion des plans et ii) par la collaboration entre agents. Cox et al. [19] abordent la coordination comme une forme de réparation itérative des défauts du plan des

2. Multi-agents Planing (MAP).

agents. Dans Buzing et al. [12], les auteurs commencent par déterminer l'ensemble des contraintes, puis les agents sont tenus de les respecter avant de commencer la planification individuelle. Hadad et al. [38] traitent la planification distribuée continue avec des contraintes temporelles. Les agents sont supposés avoir des recettes prédéfinies à partir desquelles un agent sélectionné prend une décision sur la recette à utiliser pour effectuer une action conjointe. Dans [30], les agents utilisent la négociation pour construire un plan de transport.

Tout compte fait, la différence majeure entre la PMA tel que présenté dans ces travaux et notre problème est que, dans CFDEP, nous nous intéressons à l'exécution conjointe des tâches par le biais de la formation de coalitions multi-agents. Dans ces conditions, les différences fondamentales entre PMA et CFDEP peuvent se résumer comme suit :

- *Coordination* : Dans PMA, des tâches interdépendantes sont assignées à différents agents dans le but de trouver une exécution non-confliktuelle de leurs plans. Dans CFDEP, les plans des agents sont présumés non-confliktuels s'ils sont exécutés individuellement. Il s'agit donc d'identifier les exécutions possibles qui peuvent être plus bénéfiques.
- *Tâches conjointes* : Dans PMA, les tâches communes sont prédéfinies et connues par tous les agents. L'objectif est de coordonner en conséquence l'exécution du plan des agents. Dans CFDEP, nous supposons que toutes les tâches sont réalisables conjointement selon qu'elles soient partagées ou pas entre plusieurs plans.
- *Négociation* : Dans PMA, la négociation porte surtout sur les ressources nécessaires à l'exécution des tâches. Dans CFDEP, la négociation porte sur des groupes de tâches et des agents pour les exécuter. De plus, dans la planification par négociation, on suppose que les agents négocient sur le même ensemble de tâches ; ce qui est totalement différent dans CFDEP.

3.5.3 Problèmes liés aux déplacements d'individus

Dans les problèmes liés aux déplacements d'individus (parfois appelé voyages des individus), comme dans le TSP – *Traveling Salesman Problem* – [27], le *multiple*

Traveling Salesman Problem. (**mTSP**) [7] et le *problème des vendeurs négociateurs*. (**NSP** – *Negotiating Salesmen Problem* –) [22], ils diffèrent de **MATP** dans leurs configurations. Dans le **mTSP**, tous les agents commencent et finissent au même point (ville), toutes les autres villes intermédiaires sont visitées exactement une fois, et le coût total de la visite de tous les nœuds est minimisé. Alors que dans le **mTSP**, les agents visent conjointement à minimiser la longueur totale des trajets de tous les agents. Dans le **NSP**, chaque agent n'est intéressé que par la minimisation de son trajet individuel. De plus, les modèles de [27, 7, 22] diffèrent de **MATP** à plusieurs égards. Par exemple, dans ces problèmes, les agents recherchent la performance des tâches individuelles (*c.-à-d.* voyagent chaque segment de chemin seul), alors que dans **MATP** les agents recherchent la performance collaborative de (ensembles de) tâches (*c.-à-d.* partagent les segments de chemin) lorsque cela est applicable et bénéfique. De plus, dans **MATP**, les agents peuvent avoir différents ensembles de tâches interdépendantes qui leur permettent d'atteindre leurs objectifs individuels. Une autre différence notable est que, dans **MATP**, les agents sont compétitifs et ne disposent que d'informations partielles sur les autres.

3.6 Domaines d'application de CFDEP

CFDEP s'applique à différents domaines d'application où il est question pour un ensemble d'agents d'explorer un ensemble d'alternatives pour en trouver la meilleure, qui permette par collaboration de réduire les coûts nécessaires aux agents pour atteindre leurs buts. Par exemple, il peut être appliqué dans le domaine de la mobilité intelligente (*Smart Mobility*) où l'objectif est d'optimiser les déplacements des agents et de réduire leurs coûts tout en respectant certaines contraintes et préférences. Il peut également s'appliquer au domaine de la logistique (*Logistics*) où l'objectif est d'optimiser la planification entre plusieurs acteurs de transport ayant différents moyens de transport (comme les camions) pour effectuer des livraisons de colis sur un réseau routier, où le déplacement sur les chemins et les livraisons de colis entraînent des coûts et des gains [22].

3.7 Conclusion du chapitre

Nous avons exposé dans ce chapitre le problème de la formation de coalitions avec des externalités et avons présenté une nouvelle classe d’externalités que nous appelons *Externalités d’Existence*. Par ailleurs, nous avons présenté un problème pratique, nommé MATP – *Traveling Optimisation Problem* – qui est une abstraction de plusieurs problèmes réels de la formation de coalitions avec des externalités dynamiques (CFDEP). Nous avons montré les différences et particularités de CFDEP par rapport aux problèmes de formation de coalitions existants. Dans la suite de ce manuscrit, nous présentons deux algorithmes de résolution de ce problème : un algorithme pour le contexte distribué et un algorithme pour le contexte centralisé.

CHAPITRE 4

UNE APPROCHE DISTRIBUÉE POUR LA RÉOLUTION DE CFDEP

La solution au problème CFDEP dans sa version distribuée nécessite de doter les agents de mécanismes leur permettant, d'une part, de s'engager dans des négociations afin d'échanger des propositions et des contre-propositions et, d'autre part, d'évaluer et de dériver les intentions des autres agents afin de converger vers une solution. Ainsi, l'idée est de tenir compte de l'historique des négociations pour en déduire les intentions des agents et d'en tenir compte. Pour cela, il faut des stratégies de dérivation d'intention et d'ordonnancement des alternatives possibles. Dans ce chapitre, nous présentons un algorithme distribué de formation de coalitions répondant aux spécificités de CFDEP et permettant d'améliorer les performances du système multi-agents.

4.1 Introduction

LE problème de la formation de coalitions avec des externalités dans sa version distribuée est d'une difficulté élevée en raison de la nature compétitive des agents et des dépendances entre les tâches. Pour doter les agents de mécanismes leur permettant d'interagir pour aboutir à une solution, la négociation est

sans doute une des approches les plus indiquées. Dans ce chapitre, nous présentons un algorithme fondé sur un protocole de négociations multilatérales, guidé par un mécanisme d'ordonnement des alternatives, mettant en relation l'historique des négociations et les préférences des agents afin de les faire converger vers une solution. Cette convergence est rendue possible moyennant une heuristique de recherche itérative et convergente.

4.2 CFDEP dans le contexte distribué

Dans ce chapitre, nous présentons en détail notre approche de formation de coalitions dans le contexte distribué. Pour ce faire, nous reprenons le problème MATP de la section (Sec.3.3.1) du chapitre 3. La figure Fig. 4.1 montre les possibilités individuelles pour chaque agent \mathbf{a}_i , que nous appelons *un plan* π_i .

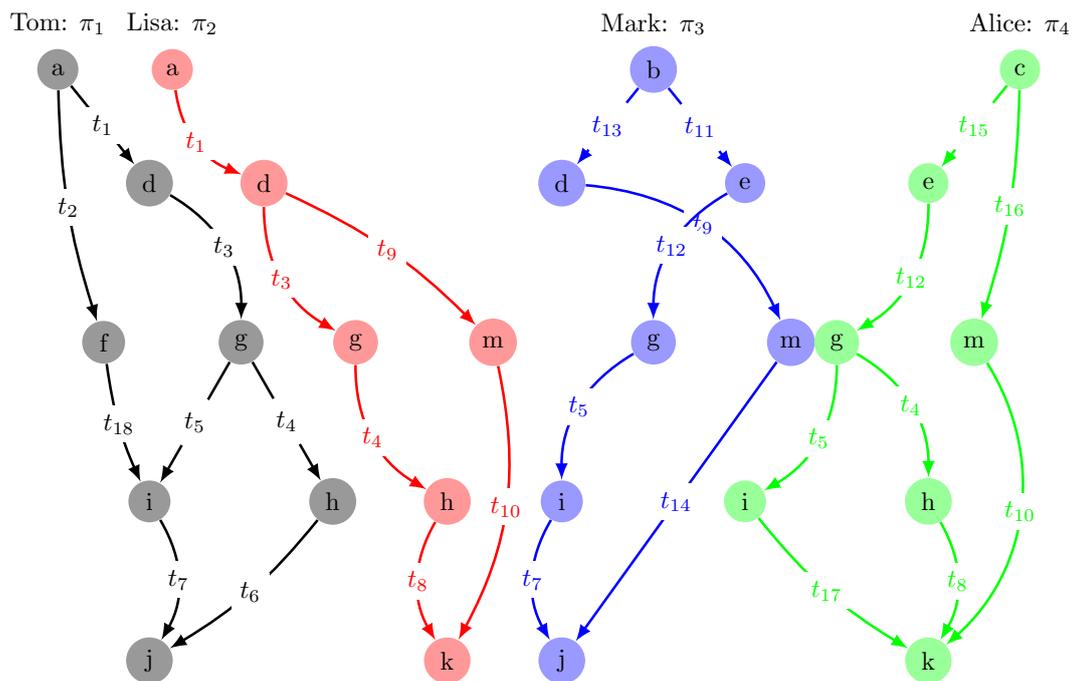


Fig. 4.1 – Les plans des agents.

Définition 4.1. - Plan. Un *plan* π_i d'un agent \mathbf{a}_i est un graphe dirigé acyclique fini : $\pi_i = (\mathfrak{V}_i, \mathcal{T}_i, \prec_i, \mathbf{u}_i)$, où $\mathfrak{V}_i = \{\mathbf{v}_i^j\}$ est l'ensemble des nœuds (les villes) et \mathcal{T}_i est l'ensemble des arcs $\mathcal{T}_i : \mathfrak{V}_i \times \mathfrak{V}_i$.

Les arcs représentent les segments des chemins dans un plan, chaque arc $(\mathbf{v}_i^1, \mathbf{v}_i^2)$ représente une tâche \mathbf{t} qui signifie un déplacement depuis \mathbf{v}_i^1 vers \mathbf{v}_i^2 . \prec_i définit un ordre de précedence sur les tâches $\mathbf{t} \in \mathcal{T}_i$. $\mathbf{t} \prec_i \mathbf{t}'$ veut dire que dans le plan π_i , \mathbf{t} précède \mathbf{t}' . \mathbf{u}_i est une fonction d'utilité non linéaire, qui associe une valeur à chaque arc : $\mathbf{u}_i : \mathcal{T}_i \rightarrow \mathbb{R}^+$.

Dans MATP, chaque agent $\mathbf{a}_i \in \mathcal{A}$ veut se déplacer depuis un nœud de départ \mathbf{v}_i^s vers un nœud de destination (son but) \mathbf{v}_i^d à travers une succession de déplacements (c.-à-d., réalisation de tâches) entre des nœuds intermédiaires $\mathbf{v}_i \in \mathfrak{V}_i$. Parmi les déplacements sur quelques segments (réalisation d'une tâche), il existe ceux qui peuvent affecter la possibilité de se déplacer sur d'autres segments. Ainsi, ils peuvent les rendre possibles ou impossibles à réaliser (une réalisation d'une tâche peut affecter l'ensemble des tâches possibles à réaliser). Dans le jargon des systèmes multi-agents, cela peut être considéré comme un ensemble de N agents $\mathcal{A} = \{\mathbf{a}_i | i = 1, \dots, N\}$ qui veulent réaliser un ensemble de tâches indépendantes avec un ordre de précedence partiel (cf. Fig. 4.1). Nous dénotons par $\mathcal{PL} = \bigcup_{i=1, \dots, N} \pi_i$ l'ensemble de tous les plans de tous les agents et par $\hat{\mathcal{T}} = \bigcup_{i=1, \dots, N} \mathcal{T}_i$ l'union de tous les ensembles de tâches des plans de tous les agents.

Donc, comme nous l'avons présenté dans la figure Fig. 4.1, chaque agent \mathbf{a}_i peut avoir plusieurs sous-ensembles de tâches dans \mathcal{T}_i qui le mènent vers son but \mathbf{v}_i^d . Nous supposons que chaque plan π_i a un nœud de départ \mathbf{v}_i^s et un nœud d'arrivée \mathbf{v}_i^d uniques. De plus, chacun des agents peut avoir plusieurs alternatives pour atteindre \mathbf{v}_i^d .

Définition 4.2. - Alternative. Une *alternative* d'un plan π_i est un tuple $\alpha_{i,k} = \langle \mathcal{T}_i^{\alpha_{i,k}}, \prec_i \rangle$ où i est l'indice de l'agent \mathbf{a}_i , k est l'indice des alternatives de \mathbf{a}_i et $\mathcal{T}_i^{\alpha_{i,k}} \subseteq \mathcal{T}_i$. $\mathcal{T}_i^{\alpha_{i,k}}$ est un ensemble de tâches séquentielles qui mènent depuis \mathbf{v}_i^s vers \mathbf{v}_i^d .

Nous dénotons l'ensemble de toutes les alternatives d'un plan π_i par $\mathcal{AL}_i =$

$\{\alpha_{i,k} | k = 1, \dots, K_i\}$ d'une taille K_i . Le tableau Tab. 4.1 illustre les alternatives des plans présentés dans la figure Fig. 4.1.

Individus	Alternatives
Tom	$\alpha_{1,1} = \langle \{t_2, t_{18}, t_7\}, \{t_2 \prec t_{18}, t_{18} \prec t_7\} \rangle$
	$\alpha_{1,2} = \langle \{t_1, t_3, t_5, t_7\}, \{t_1 \prec t_3, t_3 \prec t_5, t_5 \prec t_7\} \rangle$
	$\alpha_{1,3} = \langle \{t_1, t_3, t_4, t_6\}, \{t_1 \prec t_3, t_3 \prec t_4, t_4 \prec t_6\} \rangle$
Lisa	$\alpha_{2,1} = \langle \{t_1, t_3, t_4, t_8\}, \{t_1 \prec t_3, t_3 \prec t_4, t_4 \prec t_8\} \rangle$
	$\alpha_{2,2} = \langle \{t_1, t_9, t_{10}\}, \{t_1 \prec t_9, t_9 \prec t_{10}\} \rangle$
Mark	$\alpha_{3,1} = \langle \{t_{13}, t_9, t_{14}\}, \{t_{13} \prec t_9, t_9 \prec t_{14}\} \rangle$
	$\alpha_{3,2} = \langle \{t_{11}, t_{12}, t_5, t_7\}, \{t_{11} \prec t_{12}, t_{12} \prec t_5, t_5 \prec t_7\} \rangle$
Alice	$\alpha_{4,1} = \langle \{t_{15}, t_{12}, t_5, t_7\}, \{t_{15} \prec t_{12}, t_{12} \prec t_5, t_5 \prec t_7\} \rangle$
	$\alpha_{4,2} = \langle \{t_{15}, t_{12}, t_4, t_8\}, \{t_{15} \prec t_{12}, t_{12} \prec t_4, t_4 \prec t_8\} \rangle$
	$\alpha_{4,3} = \langle \{t_{16}, t_{10}\}, \{t_{16} \prec t_{10}\} \rangle$

Tab. 4.1 – Les alternatives des 4 individus du problème MATP.

Dans MATP, chaque tâche $t \in \hat{\mathcal{T}}$ peut être réalisée par différents moyens de transport. Nous supposons qu'il y a suffisamment de ressources pour se déplacer de \mathbf{v}^s à \mathbf{v}^d . Par conséquent, il n'y a pas de conflits inter-plans lorsqu'ils sont accomplis séparément¹. Rappelons que dans l'exemple ci-dessus, chaque tâche a un coût public ϕ^{pub} et un coût privé ϕ^{priv} . Sans coopération, lorsque α_i exécute la tâche t , il encourt un coût $\phi_i(t) = \phi^{\text{pub}}(t) + \phi_i^{\text{priv}}(t)$. Ainsi, son coût individuel d'une alternative $\alpha_{i,k}$ est $\phi_i^{\text{ind}}(\alpha_{i,k}) = \sum_{t \in \mathcal{T}_i^{\alpha_{i,k}}} \phi_i(t)$. Ce dernier représente le coût qu'il encourt lorsqu'il réalise seul toutes les tâches $t \in \mathcal{T}_i^{\alpha_{i,k}}$. Nous dénotons par $\phi_i^- = \min_{\alpha_{i,k=1, \dots, K_i}} (\phi_i^{\text{ind}}(\alpha_{i,k}))$ son coût individuel minimum de tous les coûts de ses alternatives, et $\phi_i^+ = \max_{\alpha_{i,k=1, \dots, K_i}} (\phi_i^{\text{ind}}(\alpha_{i,k}))$ son coût individuel maximum de tous les coûts de ses alternatives.

Deux plans distincts $\pi_i, \pi_{i'} \in \mathcal{PL}$, peuvent avoir des tâches communes $t^c \in (\mathcal{T}_i \cap \mathcal{T}_{i'})$. Dans le cas où t^c est réalisée conjointement par n agents, nous supposons que $\phi_i(t^c) = \frac{\phi^{\text{pub}}(t^c)}{n} + \phi_i^{\text{priv}}(t^c)$. Dans MATP, ϕ^{pub} peut être considéré comme

1. Chaque individu réalise ses tâches seul sans coordination avec les autres.

le coût résultant de la consommation de carburant ou du temps nécessaire pour se déplacer d'une ville à l'autre et ϕ_i^{priv} peut être la valeur de préférence que l'agent α_i associe à la visite de chaque ville ou une préférence sur le chemin permettant à α_i d'atteindre sa destination. Par exemple, un agent peut préférer se déplacer sur une route le long d'un littoral ou se déplacer en passant par des villes contenant des sites archéologiques ou des événements culturels, etc. Nous dénotons par $\mathcal{T}_i^c = \{t \mid t \in \mathcal{T}_i, \exists \alpha_{i'} \in \mathcal{A} : \alpha_{i'} \neq \alpha_i \wedge t \in \mathcal{T}_{i'}\}$ l'ensemble des *tâches communes* d'un agent α_i .

Pour réduire ses coûts, l'agent α_i peut réaliser conjointement ses tâches communes avec d'autres agents. Nous appelons une possible réalisation conjointe et séquentielle d'un ensemble de tâches communes : une *séquence commune directe*.

Définition 4.3. - Séquence Commune Directe. Un ensemble de tâches \mathcal{T} est une *Séquence Commune Directe* ($\vec{\mathcal{T}}$) si : $\vec{\mathcal{T}} = \{t_{h=1..m} \mid t_h \prec t_{h+1} \prec \dots \prec t_m, m \geq 1 \wedge \forall t_h, \exists \alpha_i, \alpha_{i'} \in \mathcal{A} : t_h \in (\mathcal{T}_i^c \cap \mathcal{T}_{i'}^c)\}$.

En d'autres termes, une *séquence commune directe* est un ensemble de segments de chemin, représentant des tâches séquentielles où chacune peut être réalisée par au moins deux agents. Des exemples de *séquence commune directe* de la figure Fig. 3.6 sont : $\{t_1 \prec t_3 \prec t_4\}$ avec les agents α_1 et α_2 , et $\{t_4\}$ avec les agents $\{\alpha_1, \alpha_3, \alpha_4\}$.

Un agent α_i qui a un ensemble de tâches communes \mathcal{T}_i^c peut être en mesure de réaliser conjointement différents sous-ensembles de ces tâches avec différents groupes d'agents. Nous dénotons un sous-ensemble de l'ensemble des tâches communes de l'agent α_i par $\mathcal{T}_{\mathcal{A}^{\bar{c}}}^c$ et $\mathcal{A}^{\bar{c}}$ le groupe d'agents qui peuvent réaliser conjointement $\mathcal{T}_{\mathcal{A}^{\bar{c}}}^c$. Nous appelons l'ensemble des tuples $\langle \mathcal{A}^{\bar{c}}, \mathcal{T}_{\mathcal{A}^{\bar{c}}}^c \rangle$ de l'agent α_i , un *ensemble commun*. Ainsi, α_i peut avoir plusieurs ensembles de tâches communes (dénové $\mathcal{T}_i^{\bar{c}}$) avec différents agents. Un *ensemble commun* de tâches communes $\mathcal{T}_i^{\bar{c}}$ est un ensemble de tuple $\langle \mathcal{A}^{\bar{c}}, \mathcal{T}_{\mathcal{A}^{\bar{c}}}^c \rangle$ d'un ensemble de tâches $t \in \mathcal{T}_{\mathcal{A}^{\bar{c}}}^c$ qui forment une *séquence commune directe* ($\vec{\mathcal{T}}_{\mathcal{A}^{\bar{c}}}^c$). On dit que $\vec{\mathcal{T}}_{\mathcal{A}^{\bar{c}}}^c$ est défini sur un tuple $\langle \mathcal{A}^{\bar{c}}, \mathcal{T}_{\mathcal{A}^{\bar{c}}}^c \rangle$ et peut être réalisé conjointement par un groupe d'agents $\mathcal{A}^{\bar{c}} \subseteq \mathcal{A}$ où $|\mathcal{A}^{\bar{c}}| \geq 2$ en respectant l'ordre de précedence \prec_i .

Formellement, nous avons : $\mathcal{T}_i^{\bar{c}} = \{ \langle \mathcal{A}^{\bar{c}}, \mathcal{T}_{\mathcal{A}^{\bar{c}}}^c \rangle \mid \forall \alpha_i, \alpha_{i'} \in \mathcal{A}^{\bar{c}}, \alpha_i \neq \alpha_{i'}, \exists \alpha_{i,k} \in \mathcal{A}_{\mathcal{L}_i}, \alpha_{i',k'} \in \mathcal{A}_{\mathcal{L}_{i'}} \Rightarrow \forall t \in \mathcal{T}_{\mathcal{A}^{\bar{c}}}^c : t \in \left((\mathcal{T}_i^c \cap \mathcal{T}_i^{\alpha_{i,k}}) \cap (\mathcal{T}_{i'}^c \cap \mathcal{T}_{i'}^{\alpha_{i',k'}}) \right) \}$. Le tableau

Tab. 4.2 illustre les tâches communes et les ensembles communs de chaque agent de la figure Fig. 3.5.

Agents	Tâches communes	Ensembles communs
\mathbf{a}_1	$\mathcal{T}_1^c = \{\mathbf{t}_1, \mathbf{t}_3, \mathbf{t}_4, \mathbf{t}_5, \mathbf{t}_7\}$	$\mathcal{T}_1^{\bar{c}} = \{\langle\{\mathbf{a}_1, \mathbf{a}_2\}, \{\mathbf{t}_1, \mathbf{t}_3\}\rangle, \langle\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_4\}, \{\mathbf{t}_4\}\rangle, \langle\{\mathbf{a}_1, \mathbf{a}_3, \mathbf{a}_4\}, \{\mathbf{t}_5\}\rangle, \langle\{\mathbf{a}_1, \mathbf{a}_3\}, \{\mathbf{t}_7\}\rangle\}$
\mathbf{a}_2	$\mathcal{T}_2^c = \{\mathbf{t}_1, \mathbf{t}_3, \mathbf{t}_4, \mathbf{t}_8, \mathbf{t}_9, \mathbf{t}_{10}\}$	$\mathcal{T}_2^{\bar{c}} = \{\langle\{\mathbf{a}_2, \mathbf{a}_1\}, \{\mathbf{t}_1, \mathbf{t}_3\}\rangle, \langle\{\mathbf{a}_2, \mathbf{a}_1, \mathbf{a}_4\}, \{\mathbf{t}_4\}\rangle, \langle\{\mathbf{a}_2, \mathbf{a}_3\}, \{\mathbf{t}_9\}\rangle, \langle\{\mathbf{a}_2, \mathbf{a}_4\}, \{\mathbf{t}_{10}\}\rangle\}$
\mathbf{a}_3	$\mathcal{T}_3^c = \{\mathbf{t}_5, \mathbf{t}_7, \mathbf{t}_9, \mathbf{t}_{12}\}$	$\mathcal{T}_3^{\bar{c}} = \{\langle\{\mathbf{a}_3, \mathbf{a}_1, \mathbf{a}_4\}, \{\mathbf{t}_5\}\rangle, \langle\{\mathbf{a}_3, \mathbf{a}_1\}, \{\mathbf{t}_7\}\rangle, \langle\{\mathbf{a}_3, \mathbf{a}_2\}, \{\mathbf{t}_9\}\rangle, \langle\{\mathbf{a}_3, \mathbf{a}_4\}, \{\mathbf{t}_{12}\}\rangle\}$
\mathbf{a}_4	$\mathcal{T}_4^c = \{\mathbf{t}_4, \mathbf{t}_5, \mathbf{t}_8, \mathbf{t}_{10}, \mathbf{t}_{12}\}$	$\mathcal{T}_4^{\bar{c}} = \{\langle\{\mathbf{a}_4, \mathbf{a}_2, \mathbf{a}_1\}, \{\mathbf{t}_4\}\rangle, \langle\{\mathbf{a}_4, \mathbf{a}_3, \mathbf{a}_1\}, \{\mathbf{t}_5\}\rangle, \langle\{\mathbf{a}_4, \mathbf{a}_2\}, \{\mathbf{t}_8\}\rangle, \langle\{\mathbf{a}_4, \mathbf{a}_2\}, \{\mathbf{t}_{10}\}\rangle, \langle\{\mathbf{a}_4, \mathbf{a}_3\}, \{\mathbf{t}_{12}\}\rangle\}$

Tab. 4.2 – Les tâches communes et les ensembles communs des agents.

4.3 CFDEA : un algorithme distribué de formation de coalitions

MATP comme décrit dans la section (Section 3.3.1) fait émerger de nouveaux défis et soulève plusieurs questions. Nous l'utilisons dans cette section pour illustrer une approche distribuée pour la formation de coalitions multi-agents dans un contexte à externalités dynamiques. L'algorithme CFDEA commence par l'identification des ensembles des coalitions possibles (et leurs ensembles de structures de coalitions possibles). Ensuite, les agents communiquent et interagissent pour trouver une solution acceptée par tous ; c'est ce que nous appelons un processus de négociation. La théorie de la négociation dans le domaine de la formation de coalitions multi-agents a fait l'objet d'un corpus croissant en recherche [34]². Dans

2. Ici, la négociation est considérée comme un processus d'interaction entre des agents, impliquant des actions (échange de messages de proposition) et des réactions (échange de messages

CFDEP, les négociations se font entre des groupes disjoints d'agents et concernent des sous-ensembles différents de tâches. Ainsi, elles sont particulièrement difficiles à concevoir. En effet, les agents sont compétitifs, ce qui peut engendrer des situations où ils font des propositions cycliques, et par conséquent, se retrouver dans des cas d'interblocage. Deux questions se posent dans ce contexte :

- Comment éviter les propositions cycliques ? Rappelons que pour chaque agent, il peut y avoir plusieurs solutions avec des degrés de préférences variables. Si chaque agent propose et continue à proposer sa *solution optimale* (mais qui peut être insatisfaisante pour les autres agents), cela pourrait conduire à un cycle de propositions refusées. Il est donc nécessaire de mettre en place des stratégies de concession permettant aux agents de faire des concessions progressives afin d'explorer d'autres possibilités et d'éviter les interblocages.
- Quand arrêter les négociations ? Au cours des négociations, un agent est confronté à des points de décision dans lesquels il doit décider s'il accepte la proposition reçue et mettre fin aux négociations, ou les poursuivre dans l'espoir d'une meilleure proposition dans le futur³. Le problème central dans la prise de décision dans ce contexte est que, même si les propositions arrivent séquentiellement, leur temps d'arrivée et leur valeur d'utilité pour l'agent réceptionnaire sont inconnus à l'avance. Par conséquent, si un agent met fin aux négociations trop tôt, il risque de rater des propositions futures plus favorables. En revanche, s'il attend trop longtemps, les propositions antérieures favorables peuvent ne plus réapparaître.

Les négociations dans ce cas ont les propriétés suivantes :

1. Les agents ont des informations partielles les uns sur les autres. Il s'agit d'une hypothèse réaliste, car dans la pratique, les agents sont compétitifs et ont des intérêts conflictuels. Pour des raisons stratégiques, les agents ne révèlent pas leurs stratégies, leurs contraintes, leurs préférences et les coûts

de réponse : rejet, acceptation, contre-propositions, etc.).

3. Si la durée limite des négociations est atteinte sans accord, l'agent doit soit accepter/confirmer une des propositions apparues à la fin de la durée, soit quitter les négociations sans bénéficier de coalitions

réels que leur engendrent les tâches. Nous supposons naturellement que les agents connaissent tous les autres agents disposés à négocier ainsi que leurs plans respectifs.

2. Un agent peut négocier simultanément sur de multiples ensembles de tâches avec différents groupes disjoints d'agents. Étant donné qu'aucun d'eux ne sait laquelle de ses propositions sera acceptée par les autres, et que chacun peut quitter les négociations à tout moment, un agent devrait entamer des négociations sur toutes ses possibilités d'une manière itérative et parallèle (*c.-à-d.* il y a plusieurs possibilités sur lesquelles il faut négocier en parallèle, d'une manière itérative alternée).
3. La stratégie de négociation d'un agent est une fonction de son historique de négociation vers ses actions à chaque cycle. C'est-à-dire, chaque agent tient compte à la fois des propositions reçues des autres et de ses propres propositions.
4. Le processus des négociations est limité dans le temps par une valeur τ fixée par l'utilisateur pour chaque domaine d'application.

En tenant compte de la configuration particulière de CFDEP, nous définissons un protocole de négociation, appelé *Protocole de Négociations Multilatérales* (MNP) qui permet aux agents d'identifier les coalitions bénéfiques. Ceci est réalisé par notre algorithme CFDEA (*cf.* Algorithme 1) qui, tout au long du processus de la formation de coalitions, permet aux agents de maintenir une vue d'ensemble sur leurs alternatives. Plus précisément, les agents peuvent s'engager dans plusieurs processus de négociations avec différents groupes d'agents disjoints sur différents ensembles de tâches et d'observer tout au long de ces négociations l'évolution de la désirabilité de chaque ensemble de tâches par les groupes d'agents avec lesquels ils négocient.

En effet, nous définissons un processus de formation de coalitions qui se déroule en plusieurs étapes distinctes, à savoir *échange des plans*, *génération des propositions* (SC), *propositions*, *acceptations* et *confirmations* comme l'illustre la figure Fig. 4.2.

L'algorithme CFDEA consiste en un processus itératif cyclique entre ces étapes.

Algorithme 1 CFDEA(π_i)

Require: τ , $\text{time}_{\text{current}}$, π_i ;

- 1: $Q_i \leftarrow$ calculer l'ensemble des alternatives dans π_i ;
 - 2: calculer l'utilité de référence u_i^δ
 - 3: $\text{time}_{\text{start}} \leftarrow \text{current_time}$;
 - 4: $\text{time}_{\text{current}} \leftarrow 0$;
 - 5: $\text{nextStep} \leftarrow$ Échange de plans;
 - 6: $\text{problemSolved} \leftarrow \text{false}$;
 - 7: **while** ($(\text{time}_{\text{current}} < \tau)$ **and** (**not** problemSolved)) **do**
 - 8: $\text{runStep}(\text{nextStep}, Q_i, r)$;
 - 9: $\mathcal{M}_i \leftarrow \mathcal{M}_i \cup$ les messages reçus dans r
 - 10: $\text{nextStep} \leftarrow \text{selectNextStep}(\mathcal{M}_i, \text{nextStep}, r)$ // *déterminer la prochaine étape*
 - 11: $\text{time}_{\text{current}} \leftarrow \text{current_time} - \text{time}_{\text{start}}$;
 - 12: **end while**
 - 13: $\hat{c}s_i \leftarrow \text{bestRemained}(Q_i)$
-

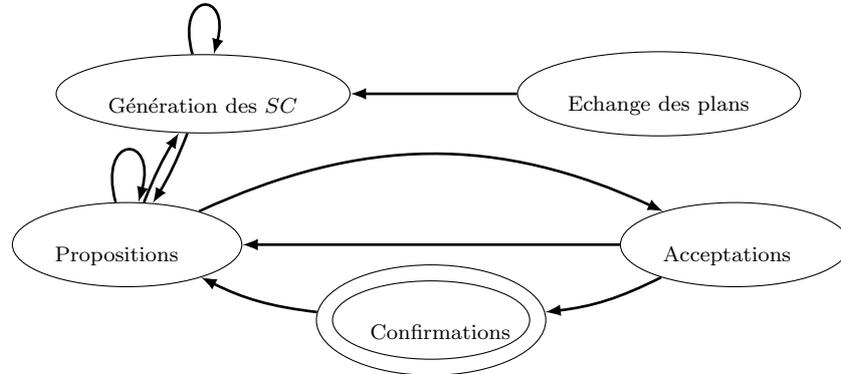


Fig. 4.2 – Les étapes des négociations.

Il commence par le calcul de toutes les alternatives du plan d'un agent, puis, initialise la variable nextStep à l'étape *échange de plans* dans laquelle les agents échangent leurs plans respectifs. Tant qu'aucune solution n'est trouvée et que le temps limite des négociations n'est pas atteint, l'algorithme boucle et fait des ité-

rations sur les étapes : *propositions*, *acceptations* et *confirmations* qui représentent un cycle des négociations. À la fin de celle-ci, chaque agent choisit sa meilleure alternative parmi les alternatives qui contiennent les tâches pour lesquelles il s'est engagé à les réaliser conjointement.

4.3.1 Génération des structures de coalitions

D'une manière générale, l'objectif de la génération des structures de coalitions (GSC) est de trouver la structure qui maximise le bien-être social ou minimise l'écart entre les coalitions formées et celles initialement souhaitées par les agents. En effet, le problème de GSC consiste à partitionner les agents en groupes disjoints afin d'améliorer leur performance [62]. Dans CFDEP, la GSC concerne l'identification des groupes de tâches ainsi que des groupes d'agents — pas nécessairement disjoints — pour les réaliser. Dans Rahwan et al. [62], des coalitions mutuellement disjointes sont formées en divisant l'ensemble des agents où chaque tâche est attribuée à un groupe d'agents. Dans CFDEP, nous n'allouons pas nécessairement toutes les tâches et les agents peuvent être impliqués dans plusieurs coalitions à des instants différents. L'existence de dépendances entre les tâches et plusieurs sous-ensembles possibles de tâches qui permettent d'atteindre un but de différentes manières, ont une influence dans CFDEP. Par ailleurs, nous nous concentrons sur un contexte distribué avec des agents compétitifs détenant des informations partielles sur les autres (par exemple, les fonctions d'utilité des agents sont privées).

Définition 4.4. - Coalition. Une *coalition* $c_{i,q}$ où q est l'indice des coalitions de l'agent a_i , est un tuple $c_{i,q} = \langle \mathcal{A}_{c_{i,q}}, \overrightarrow{\mathcal{T}_{c_{i,q}}} \rangle$ où $\mathcal{T}_{c_{i,q}} \subseteq \mathcal{T}_i$, avec $|\mathcal{T}_{c_{i,q}}| \geq 1$ et $\mathcal{T}_{c_{i,q}}$ forme une *Séquence commune directe* $(\overrightarrow{\mathcal{T}_{c_{i,q}}})$, est l'ensemble des tâches à réaliser par le groupe d'agents $\mathcal{A}_{c_{i,q}} \subseteq \mathcal{A}$, avec $|\mathcal{A}_{c_{i,q}}| \geq 2$.

Nous dénotons l'ensemble de toutes les coalitions d'un agent par $\mathcal{C}_i = \{c_{i,q=1,\dots,Q_i} \mid \forall q, \exists \mathcal{T}_i^c = \{\langle \mathcal{A}^c, \mathcal{T}_{\mathcal{A}^c}^c \rangle\} : \mathcal{A}_{c_{i,q}} \subseteq \mathcal{A}^c \wedge \mathcal{T}_{c_{i,q}} \subseteq \mathcal{T}_{\mathcal{A}^c}^c\}$. Un exemple de coalition d'agent a_1 peut être $c_{1,1} = \langle \{a_1, a_2\}, \{t_1, t_3\} \rangle$.

Pour former les coalitions possibles et bénéfiques, les agents doivent négocier sur les ensembles de tâches à accomplir en s'échangeant des propositions et contre-

propositions. Ils explorent ainsi l'ensemble des coalitions possibles. Un agent \mathbf{a}_i qui veut négocier sur une alternative particulière doit considérer les coalitions pertinentes à cette alternative. En effet, l'agent calcule ses coalitions pour générer une structure de coalitions $\mathbf{cs}_{i,s}$ où s est l'indice de la structure de coalitions de l'agent \mathbf{a}_i .

Définition 4.5. - Structure de coalitions. Une *structure de coalitions* $\mathbf{cs}_{i,s} = \{\mathbf{c}_{i,q} \mid \exists \alpha_{i,k} : (\cup_{\mathbf{c}_{i,q} \in \mathbf{cs}_{i,s}} \mathcal{T}_{\mathbf{c}_{i,q}}) \subseteq \mathcal{T}_i^{\alpha_{i,k}}\}$ est un ensemble de coalitions pour réaliser les tâches d'une alternative de plan $\alpha_{i,k}$ où $\cup_{\mathbf{c}_{i,q} \in \mathbf{cs}_{i,s}} \mathcal{A}_{\mathbf{c}_{i,q}} \subseteq \mathcal{A}$ et $\forall \mathbf{c}_{i,q}, \mathbf{c}_{i,q'} \in \mathbf{cs}_{i,s}, \mathcal{T}_{\mathbf{c}_{i,q}} \cap \mathcal{T}_{\mathbf{c}_{i,q'}} = \emptyset$.

Nous dénotons par $\Lambda(\mathbf{cs}_{i,s}) = \cup_{\mathbf{c}_{i,q} \in \mathbf{cs}_{i,s}} \mathcal{A}_{\mathbf{c}_{i,q}}$ l'ensemble des agents dans $\mathbf{cs}_{i,s}$ et par $\Gamma(\mathbf{cs}_{i,s}) = \cup_{\mathbf{c}_{i,q} \in \mathbf{cs}_{i,s}} \mathcal{T}_{\mathbf{c}_{i,q}}$ l'ensemble des tâches dans $\mathbf{cs}_{i,s}$. Chaque alternative $\alpha_{i,k}$ est associée à une structure de coalitions $\mathbf{cs}_{i,s}$ que nous dénotons $\alpha_{i,k}^{\mathbf{cs}_{i,s}}$. D'où, nous avons $\Gamma(\alpha_{i,k}^{\mathbf{cs}_{i,s}}) \subseteq \mathcal{T}_i^{\alpha_{i,k}}$.

Considérons $\alpha_{1,3}$, la troisième alternative du plan de l'agent \mathbf{a}_1 (cf. Table 4.1), des exemples de coalitions possibles sont alors :

$$\mathcal{C}_1 = \left\{ \begin{array}{l} \mathbf{c}_{1,1} = \langle \{\mathbf{a}_1, \mathbf{a}_2\}, \{\mathbf{t}_1\} \rangle, \\ \mathbf{c}_{1,2} = \langle \{\mathbf{a}_1, \mathbf{a}_2\}, \{\mathbf{t}_1, \mathbf{t}_3\} \rangle, \\ \mathbf{c}_{1,3} = \langle \{\mathbf{a}_1, \mathbf{a}_2\}, \{\mathbf{t}_1, \mathbf{t}_3, \mathbf{t}_4\} \rangle, \\ \mathbf{c}_{1,4} = \langle \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_4\}, \{\mathbf{t}_4\} \rangle, \\ \mathbf{c}_{1,5} = \langle \{\mathbf{a}_1, \mathbf{a}_3\}, \{\mathbf{t}_7\} \rangle, \\ \dots \end{array} \right\}$$

Un exemple de structure de coalitions associée à $\alpha_{1,3}$ ($\mathbf{cs}_{1,3}$ ou $\alpha_{1,3}^{\mathbf{cs}_{1,3}}$) est : $\alpha_{1,3}^{\mathbf{cs}_{1,3}} = \{\mathbf{c}_{1,2}, \mathbf{c}_{1,4}\}$.

Pour la génération des propositions, nous utilisons l'algorithme de recherche en profondeur (*depth-first search*) [18] pour l'exploration des plans des agents. Sa complexité est de $\mathbf{N} \times \mathbf{O}(\mathbf{V}_i + \mathcal{T}_i)$ où \mathbf{N} est le nombre d'agents, \mathbf{V}_i et \mathcal{T}_i sont le nombre de nœuds et d'arcs dans π_i .

4.3.2 Représentation de l'espace de recherche

Quand un agent a_i cherche à trouver une solution conjointe (une structure de coalitions), il doit explorer l'ensemble des structures de coalitions possibles \mathcal{S}_i , que nous appelons *espace de recherche*. Ainsi, $|\mathcal{S}_i| = \sum_{k=1}^{K_i} \prod_{h=1}^{|\mathcal{T}_i^{\alpha_i, k}|} (2^{\Omega(t_h)} - \Omega(t_h))$ où $\Omega(t_h)$ représente le nombre de plans $\pi \in \mathcal{PL}$ qui contiennent la tâche t . Par exemple, le cardinal de l'espace de recherche de a_1 , comme illustré dans la figure Fig. 4.3 est $|\mathcal{S}_1| = ((2^1-1) \times (2^1-1) \times (2^2-2)) + ((2^2-2) \times (2^2-2) \times (2^3-3) \times (2^2-2)) + ((2^2-2) \times (2^2-2) \times (2^3-3) \times (2^1-1)) = 62$. Alors, le cardinal de l'espace de recherche $\bar{\mathcal{S}}$ de tous les agents est $|\bar{\mathcal{S}}| \leq \prod_{i=1}^N |\mathcal{S}_i|$.

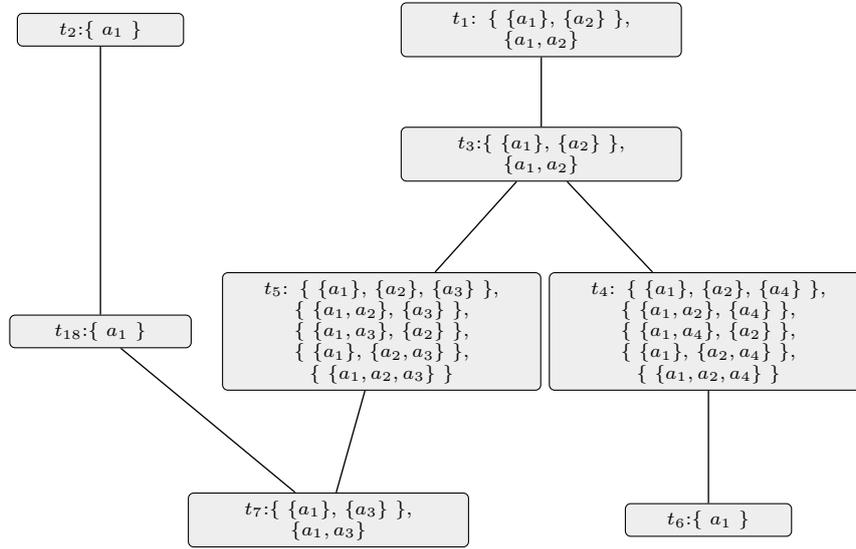


Fig. 4.3 – L'espace de recherche de l'agent a_1 .

En utilisant les définitions présentées ci-dessus, un CFDEP est un tuple $\langle \mathcal{A}, \mathcal{PL}, \tau \rangle$ avec $\tau \in \mathbb{R}^+$, le temps limite (*deadline*) pour le processus de formation de coalitions. Dans CFDEP, l'objectif de chaque agent $a_i \in \mathcal{A}$, associé à un plan $\pi_i = (V_i, \mathcal{T}_i, \prec_i, u_i)$ est de trouver une structure de coalitions possible $cs_{i,s}$ en conformité avec \prec_i pour atteindre son but v_i^d avec une valeur acceptable de sa fonction d'utilité u_i . Pour cela, dans notre approche, le processus de formation de coalitions commence par l'échange des plans entre les agents concernés et ensuite, chacun

d'eux procède à l'analyse des tâches de chaque plan reçu et de leurs dépendances correspondantes pour reconstituer les alternatives des autres agents. Ces alternatives sont utilisées pour identifier les tâches communes et les ensembles communs pour ensuite générer les structures de coalitions possibles.

4.3.3 Protocole de négociations multilatérales

Le MNP (*Multilateral Negotiation Protocol*) est un protocole de négociation itératif (exécution cyclique) sur des étapes séquentielles. Chaque cycle comprend trois étapes consécutives : (1) *Propositions* ; (2) *Acceptations* ; (3) *Confirmations*. Chaque cycle a un temps limité. Le protocole laisse une liberté aux agents dans le sens où ils ne sont pas obligés de suivre toutes les étapes et qu'ils sont libres de se retirer à tout moment d'un cycle pour repasser de nouveau sur la première étape d'un nouveau cycle, à savoir l'étape *Propositions*. Les propositions échangées sont ensuite utilisées par les agents pour dériver les préférences des autres. Par ailleurs, aucun partage complet d'information n'est nécessaire.

Les agents peuvent proposer des coalitions de leurs structures de coalitions à partir de leur espace de recherche et peuvent accepter, confirmer ou ignorer les propositions faites par les autres. De cette manière, les agents sont en mesure de faire des évaluations et de mettre à jour leurs visions sur leurs structures de coalitions possibles à la fin de chaque cycle. Essentiellement, l'idée est que les agents s'informent mutuellement de leurs dispositions, mais sans être obligés de les respecter tant que des confirmations ne sont pas échangées⁴. Par exemple, après avoir envoyé des propositions à différents groupes d'agents, l'agent émetteur n'est pas tenu de respecter ses propositions et de continuer les négociations sur celles-ci. De plus, il n'est pas tenu de leur envoyer (aux agents émetteurs) des acceptations même s'il reçoit des acceptations de la plupart d'entre eux (mais pas de tous). En effet, comme nous l'avons indiqué précédemment, chaque coalition peut impacter

4. Chaque agent étant seulement intéressé par l'optimisation de sa propre utilité, il doit donc s'assurer que les autres agents reçoivent aussi assez d'utilité pour assurer leur coopération, *i.e.* chaque agent essaie d'exploiter ses adversaires autant que possible et n'a aucun intérêt à atteindre un optimum social.

Algorithme 2 RunStep(nextStep, \mathcal{Q}_i, r)

```

1: if (nextStep == Échange de plans) then
2:   for all  $a_j, j \neq i, \in \mathcal{A}$  do
3:     envoyer  $\pi_i$  à  $a_j$ 
4:   end for
5: else
6:   if (nextStep == Échange de propositions) then
7:      $cs_i \leftarrow$  Ordonnanceur( $\mathcal{Q}_i, r$ )
8:     if décision( $cs_i, \text{Proposal}$ ) then
9:        $\forall a_{i'} \in \mathcal{A}_{c_{i'}, q'}$ , envoyer  $m = (\text{Proposal}, a_i, a_{i'}, c_{i, q})$  à  $a_{i'}$ .
10:    end if
11:  else
12:    if (nextStep == Échange d'acceptations) then
13:      if décision( $cs_i, \text{Accept}$ ) then
14:         $\forall a_{i'} \in \mathcal{A}_{c_{i'}, q'}$ , si (Règle 2) alors, envoyer  $m = (\text{Accept}, a_i, a_{i'}, c_{i, q'})$ 
15:          à  $a_{i'}$ .
16:      end if
17:    else
18:      if (nextStep == Échange de confirmations) then
19:        if décision( $cs_i, \text{Confirm}$ ) then
20:           $\forall a_{i'} \in \mathcal{A}_{c_{i'}, q'}$ , si (Règle 4) alors, envoyer  $m =$ 
21:            ( $\text{Confirm}, a_i, a_{i'}, c_{i, q'}$ ) à  $a_{i'}$ .
22:          end if
23:        end if
24:      end if

```

la possibilité de réalisation d'autres coalitions. Par conséquent, s'il arrive qu'un des agents n'a pas accepté une proposition, cela pourrait rendre toutes les autres propositions impossibles à réaliser.

Par ailleurs, si une proposition est mutuellement acceptée et confirmée, cela signifie que chacun des agents impliqués dans la proposition s'est engagé à exécuter conjointement l'ensemble de ses tâches avec les agents concernés.

Formulation des propositions

Chaque agent \mathbf{a}_i a une fonction d'utilité $\mathbf{u}_i : \mathcal{Q}_i \rightarrow \mathbb{R}$ qui associe une valeur privée à chaque alternative $\alpha_{i,k}$, induisant un ordre de préférence sur \mathcal{Q}_i . Nous dénotons par \mathbf{u}_i^e l'utilité estimée d'une alternative $\alpha_{i,k}$ durant les négociations et par \mathbf{u}_i l'utilité finale⁵ d'une alternative $\alpha_{i,k}$ à la fin des négociations. Rappelons que $\alpha_{i,k}^{cs_{i,s}}$ est la structure de coalitions $cs_{i,s}$ associée avec l'alternative $\alpha_{i,k}$. En commençant par son alternative la plus bénéfique α_i^* (que nous appelons *Alternative optimale* pour \mathbf{a}_i), un agent \mathbf{a}_i négocie sur ses alternatives dans un ordre déterminé et met à jour à chaque cycle de négociations dans l'objectif de prendre en compte les intentions des autres agents sur ses coalitions (propositions de coalitions). Quand aucun engagement n'est obtenu pour une alternative $\alpha_{i,k}$, l'agent passe à une autre alternative $\alpha_{i,k'}$ en fonction des informations dérivées de l'historique des négociations. Ce processus est itéré jusqu'à ce qu'un accord concernant une structure de coalitions est atteint ou le temps limite des négociations est arrivé à expiration. Un agent \mathbf{a}_i peut accepter une structure de coalitions $cs_{i,s}$ d'une alternative $\alpha_{i,k}$ si $\mathbf{u}_i^e(\alpha_{i,k}) > \mathbf{u}_i^\delta$ où \mathbf{u}_i^δ dénote l'utilité de référence de \mathbf{a}_i en dessous de laquelle l'agent n'accepte pas les propositions concernant une alternative (ou une structure de coalitions). De l'exemple MATP, \mathbf{u}_i^δ peut être définie en fonction de différents facteurs comme le nombre total des agents dans le système, la nature des plans des agents ou la valeur de l'utilité qu'un agent \mathbf{a}_i peut avoir quand il réalise seul ses tâches préférées.

Par leur nature d'agents compétitifs, chaque agent \mathbf{a}_i préfère son alternative optimale qui lui permet de maximiser son utilité $\mathbf{u}_i^e(\alpha_i^*)$. Par ailleurs, chacun d'eux peut avoir plusieurs alternatives optimales. Nous supposons dans un tel cas que

5. L'utilité estimée est l'utilité qu'un agent espère avoir si tous les agents impliqués dans ses propositions les acceptent et l'utilité finale d'un agent est l'utilité que l'agent obtient réellement à la fin des négociations.

pour entamer les négociations, l'agent va sélectionner aléatoirement une alternative.

Acceptation et confirmation des propositions

Au cours d'une négociation, un agent \mathbf{a}_i envoie et reçoit des propositions, des acceptations et des confirmations. Si ses propositions sont acceptées, \mathbf{a}_i les confirme aux autres. Toutefois, il peut aussi changer d'avis et ne pas les confirmer. De plus, il peut également ne pas confirmer les propositions qu'il a reçues, même s'il les a déjà acceptées. En effet, dans notre protocole, un accord sur une coalition ne lie que les parties qui l'ont confirmée mutuellement (au moins deux agents). Ainsi, la confirmation d'une proposition par un seul agent (confirmation unilatérale) n'est pas un accord contraignant. Cependant, au cours d'un cycle de négociation, un agent \mathbf{a}_i qui a confirmé une proposition demeure engagé sur sa confirmation jusqu'à la fin du cycle. Si pendant le même cycle, l'agent reçoit des confirmations d'autres agents, alors il ne peut plus se désengager.

Définition 4.6. - Ensemble engagé. Un *ensemble engagé* $\theta_{\mathbf{c}_{i,q}} = \{\langle \mathbf{A}_{\theta_{\mathbf{c}_{i,q}}}, \mathbf{T}_{\theta_{\mathbf{c}_{i,q}}} \rangle\}$, d'une coalition $\mathbf{c}_{i,q}$ est un ensemble de tuples $\langle \mathbf{A}_{\theta_{\mathbf{c}_{i,q}}}, \mathbf{T}_{\theta_{\mathbf{c}_{i,q}}} \rangle$ où $\mathbf{A}_{\theta_{\mathbf{c}_{i,q}}}$ est l'ensemble des agents ayant mutuellement confirmé leur engagement pour réaliser les tâches $\mathbf{t} \in \mathbf{T}_{\theta_{\mathbf{c}_{i,q}}} \subseteq \mathcal{T}_{\mathbf{c}_{i,q}}$. Alors, $\forall \theta_{\mathbf{c}_{i,q}} : \mathbf{A}_{\theta_{\mathbf{c}_{i,q}}} \subseteq \mathcal{A}_{\mathbf{c}_{i,q}}$.

Définition 4.7. - Coalitions équivalentes. Deux coalitions $\mathbf{c}_{i,q}$, et $\mathbf{c}_{i',q'}$ de deux agents \mathbf{a}_i et $\mathbf{a}_{i'}$ respectivement, sont *équivalentes* : $\mathbf{c}_{i,q} \equiv \mathbf{c}_{i',q'}$ iff $\mathcal{T}_{\mathbf{c}_{i,q}} = \mathcal{T}_{\mathbf{c}_{i',q'}} \wedge (\mathbf{a}_i \in \mathcal{A}_{\mathbf{c}_{i',q'}} \wedge \mathbf{a}_{i'} \in \mathcal{A}_{\mathbf{c}_{i,q}})$.

Une fois que deux agents \mathbf{a}_i et $\mathbf{a}_{i'}$ ont mutuellement confirmé leurs coalitions respectives $\mathbf{c}_{i,q} \in \mathbf{cs}_{i,s}$ et $\mathbf{c}_{i',q'} \in \mathbf{cs}_{i',s'}$, où $\mathbf{c}_{i,q} \equiv \mathbf{c}_{i',q'}$, ils procèdent à la mise à jour de leur *ensemble engagé*, tel que $\mathbf{A}_{\theta_{\mathbf{c}_{i,q}}} \leftarrow \mathbf{A}_{\theta_{\mathbf{c}_{i,q}}} \cup \mathbf{a}_{i'}$ et $\mathbf{A}_{\theta_{\mathbf{c}_{i',q'}}} \leftarrow \mathbf{A}_{\theta_{\mathbf{c}_{i',q'}}} \cup \mathbf{a}_i$. Par ailleurs, au cours des cycles de négociations, un agent $\mathbf{a}_{i''} \in \mathbf{A}_{\theta_{\mathbf{c}_{i,q}}}$, peut faire des propositions à un agent \mathbf{a}_i uniquement parmi les structures de coalitions $\mathbf{cs}_{i'',s''}$ tel que $\exists \mathbf{c}_{i'',q''} \in \mathbf{cs}_{i'',s''}, \exists \mathbf{c}_{i,q} \in \mathbf{cs}_{i,s} : \mathbf{c}_{i'',q''} \equiv \mathbf{c}_{i,q}$. Cela réduit considérablement l'ensemble des alternatives qu'un agent peut prendre en compte dans ses cycles futurs de négociations.

Règles de communication entre agents

Les agents communiquent par échange de messages de la forme $m = (\text{Type}, a_i, a_{i'}, c_{i,q})$ où **Type** est le type du message (**Proposal** : pour une proposition, **Accept** : pour une acceptation et **Confirm** : pour une confirmation), $a_i \neq a_{i'}$, $a_i \in \mathcal{A}_{c_{i,q}}$ est l'émetteur, $a_{i'} \in \{\mathcal{A}_{c_{i,q}} \setminus \mathcal{A}_{\emptyset_{c_{i,q}}}\}$ est le récepteur (destinataire) et $c_{i,q} \in \alpha_{i,k}^{cs_{i,s}}$ est une proposition de coalition d'une alternative $\alpha_{i,k}$. L'ensemble de tous les messages envoyés et reçus d'un agent a_i est noté \mathcal{M}_i . Pour faciliter les négociations, nous introduisons quelques règles d'interaction, comme suit :

Règle 1. : $\forall a_i \in \mathcal{A}$ $m = (\text{Proposal}, a_i, a_{i'}, c_{i,q})$ peut être soumis plusieurs fois, mais pas au cours d'un même cycle r .

Règle 2. : $\forall a_i \in \mathcal{A}$, $m = (\text{Accept}, a_i, a_{i'}, c_{i',q'})$ peut être soumis au cours d'un cycle r seulement si $\exists m' \in \mathcal{M}_i : m' = (\text{Proposal}, a_{i'}, a_i, c_{i',q'})$ qui avait été reçu durant r , et $\exists m'' \in \mathcal{M}_i : m'' = (\text{Proposal}, a_i, a_{i'}, c_{i,q})$ qui avait été envoyé durant r où $c_{i,q} \equiv c_{i',q'}$.

Notons que a_i peut envoyer un message m plusieurs fois, mais durant le même cycle r . De plus, si a_i a envoyé une proposition, c.-à-d. $\exists m'' \in \mathcal{M}_i : m'' = (\text{Proposal}, a_i, a_{i'}, c_{i,q})$ et $c_{i,q} \equiv c_{i',q'}$ et avait reçu une proposition tel que $m' \in \mathcal{M}_i : m' = (\text{Proposal}, a_{i'}, a_i, c_{i',q'})$ alors a_i n'est pas tenu de répondre par un message d'acceptation $m = (\text{Accept}, a_i, a_{i'}, c_{i,q})$.

Règle 3. : $\forall a_i \in \mathcal{A}$, $m = (\text{Accept}, a_i, a_{i'}, c_{i',q'})$ peut être soumis dans un cycle r même si $\exists m' \in \mathcal{M}_i : m' = (\text{Proposal}, a_{i'}, a_i, c_{i',q'})$ reçu au cours d'un cycle $r' < r$, peu importe si a_i a soumis un message d'acceptation à cette proposition ou pas.

Règle 4. : $\forall a_i \in \mathcal{A}$, $m = (\text{Confirm}, a_i, a_{i'}, c_{i',q'})$ peut être soumis dans un cycle r seulement si $\exists m' \in \mathcal{M}_i : m' = (\text{Accept}, a_{i'}, a_i, c_{i,q})$ reçu durant un cycle r , et $\exists m'' \in \mathcal{M}_i : m'' = (\text{Accept}, a_i, a_{i'}, c_{i',q'})$ envoyé durant un cycle r ou $c_{i,q} \equiv c_{i',q'}$.

Notons que m peut être soumis plusieurs fois, mais pas durant le même cycle r et $\forall a_i \in \mathcal{A}$, a_i n'est pas tenu de répondre par un message de type $m = (\text{Confirm}, a_i, a_{i'}, c_{i',q'})$ sur un message reçu $m' \in \mathcal{M}_i : m' = (\text{Accept}, a_{i'}, a_i, c_{i,q})$, où $c_{i,q} \equiv c_{i',q'}$, même si $\exists m'' \in \mathcal{M}_i : m'' = (\text{Accept}, a_i, a_{i'}, c_{i',q'})$, soumis durant un cycle $r' \leq r$.

Règle 5. : $\forall \mathbf{a}_i \in \mathcal{A}$, $\mathbf{m} = (\text{Confirm}, \mathbf{a}_i, \mathbf{a}_{i'}, \mathbf{c}_{i',q'})$ peut être soumis durant un cycle r même si $\exists \mathbf{m}' \in \mathcal{M}_i : \mathbf{m}' = (\text{Accept}, \mathbf{a}_{i'}, \mathbf{a}_i, \mathbf{c}_{i,q})$ reçu durant un cycle $r' < r$ et $\exists \mathbf{m}'' \in \mathcal{M}_i : \mathbf{m}'' = (\text{Accept}, \mathbf{a}_i, \mathbf{a}_{i'}, \mathbf{c}_{i',q'})$ reçu durant un cycle $r' < r$ où $\mathbf{c}_{i,q} \equiv \mathbf{c}_{i',q'}$ peu importe si \mathbf{a}_i a soumis un message de confirmation.

Règle 6. : $\forall \mathbf{a}_i \in \mathcal{A}$, durant un cycle r , si $\mathbf{m} = (\text{Confirm}, \mathbf{a}_i, \mathbf{a}_{i'}, \mathbf{c}_{i',q'})$ a été soumis et $\exists \mathbf{m}' \in \mathcal{M}_i : \mathbf{m}' = (\text{Confirm}, \mathbf{a}_{i'}, \mathbf{a}_i, \mathbf{c}_{i,q})$ où $\mathbf{c}_{i,q} \equiv \mathbf{c}_{i',q'}$ a été reçu durant r , alors les deux agents \mathbf{a}_i et $\mathbf{a}_{i'}$ le considéreront comme un accord et s'engageront à former conjointement la coalition. Alors : $\theta(\mathbf{c}_{i,q}) \leftarrow \theta(\mathbf{c}_{i,q}) \cup \{\mathbf{a}_{i'}\}$.

Ces règles se traduisent par 5 transitions possibles entre les étapes : *Propositions* \longrightarrow *Propositions*, *Propositions* \longrightarrow *Acceptations*, *Acceptations* \longrightarrow *Propositions*, *Acceptations* \longrightarrow *Confirmations*, *Confirmations* \longrightarrow *Propositions*.

Utilisation des règles de communication durant les négociations

Pour participer dans le processus de négociations, un agent \mathbf{a}_i commence par sélectionner son alternative $\alpha_{i,k}^*$ et envoie ses propositions de coalitions $\mathbf{c}_{i,q} \in \alpha_{i,k}^{\text{CS}_{i,s}}$ à leurs agents respectifs $\mathbf{a}_{i'} \in \mathcal{A}_{\mathbf{c}_{i,q}}$ (*cf.* Règle 1), puis attend de recevoir les propositions des autres agents. S'il ne reçoit aucune proposition, il attend le cycle suivant et choisit une autre alternative $\alpha_{i,k'}$ à partir de laquelle il enverra ses nouvelles propositions⁶. Sinon, il procédera à l'évaluation des propositions reçues et prend une décision s'il les accepte ou pas (*cf.* Règles 2,3). Après que l'agent ait envoyé des acceptations sur des propositions qu'il a déjà reçues, il attend de recevoir des acceptations des autres agents. S'il en reçoit, il procède à leur évaluation pour décider s'il est bénéfique pour lui de les confirmer ou pas (*cf.* Règles 4,5). S'il ne reçoit aucune acceptation, il attend le cycle suivant et choisit une autre alternative $\alpha_{i,k'}$ pour formuler ses propositions à nouveau. S'il envoie des confirmations, alors il attend de recevoir celles des autres agents. Par conséquent, s'il reçoit au moins une confirmation sur une des propositions qu'il a confirmé, il devient alors engagé vis-à-vis de l'agent qui lui a envoyé la confirmation et cela même s'il n'atteint pas son utilité estimée (*cf.* Règle 6). En revanche, s'il ne reçoit aucune confirmation au

6. Notons que $\alpha_{i,k'}$ pourrait être la même que $\alpha_{i,k}^*$, en fonction de \mathcal{P}_i .

cours du même cycle, il devient alors libre de tout engagement et revient à l'étape d'échange de propositions. Alors, l'agent va considérer les confirmations qu'il a déjà envoyé comme obsolètes à la fin de l'étape d'échange de confirmations.

Dans CFDEP, chaque agent peut s'engager dans plusieurs négociations avec différents groupes disjoints d'agents concernant différents ensembles de tâches. Les interdépendances entre les tâches peuvent induire des interdépendances entre les engagements. Par conséquent, chacun des agents voudrait stratégiquement recevoir les confirmations pour un ou plusieurs sous-ensembles de tâches avant d'envoyer ses confirmations à d'autres groupes d'agents. Cela peut mener à des interblocages entre les agents. Pour éviter ces situations, nous utilisons une boîte de réception centrale⁷ qui se charge de collecter et de délivrer les messages à tous les agents en même temps, suivant deux phases pour chaque étape. À savoir, une phase de collecte des messages de tous les agents, et une phase de transmission des messages pour ceux-ci. Ainsi, à chaque cycle des négociations, chacun des agents envoie ses propositions, acceptations et confirmations à la boîte de réception centrale. Les messages ne seront délivrés aux agents qu'après expiration d'une période définie. La réception se fait alors après l'envoi des messages de tous les agents, ce qui permet d'éviter les interblocages.

4.3.4 Ordonnancement et prise de décisions

Tout au long du processus de négociations, les agents sont amenés à prendre des décisions s'ils proposent, acceptent, confirment les propositions qu'ils ont reçues ou non (*cf.* Algorithme 3). En effet, à chaque étape, un agent doit décider s'il accepte (ou confirme) les propositions qu'il a reçues durant cette étape et mettre fin à sa participation dans les négociations, ou poursuivre en quête de meilleures propositions. Par ailleurs, les agents sont confrontés au risque de ne pas parvenir à former des coalitions avant le deadline. En effet, les propositions arrivent de manière séquentielle et les agents ne connaissent pas leurs valeurs respectives avant

7. Notons que cette boîte de réception n'a aucun effet sur la prise de décisions pour les agents. Elle est uniquement utilisée comme moyen de synchronisation des messages échangés entre les agents.

leur arrivée. Ainsi, ils prennent leurs décisions en fonction de l'utilité des propositions dont ils disposent à chaque étape. Avant les négociations, un agent fixe sa valeur d'utilité de référence u_i^δ et procède à un premier ordonnancement de ses alternatives ($r = 1$) en fonction des valeurs de leurs d'utilités estimées. Ensuite, il sélectionne l'alternative $\alpha_{i,k}$ qui a l'utilité estimée la plus élevée $u_i^e(\alpha_{i,k}) > u_i^\delta$ et envoie ses propositions de coalitions $c_{i,q} \in \alpha_{i,k}^{cs_{i,s}}$ aux agents concernés ($a_{i'} \in \mathcal{A}_{c_{i,q}}$).

Après réception des propositions des autres agents, chaque agent a_i met à jour les valeurs de son utilité estimée de toutes ses alternatives $u_i^e(\alpha_{i,k=1,\dots,K_i})$ en tenant compte de ces propositions. Si $u_i^e(\alpha_{i,k}) > u_i^\delta$, il accepte les propositions reçues, sinon (si $u_i^e(\alpha_{i,k}) \leq u_i^\delta$) il attend le prochain cycle et met-à-jour l'utilité estimée de ses alternatives $u_i^e(\alpha_{i,k=1,\dots,K_i})$ afin de sélectionner une autre $\alpha_{i,k'}$ et proposer ses coalitions ($c_{i,q'} \in \alpha_{i,k'}^{cs_{i,s'}}$).

Après l'envoi des acceptations sur les propositions de coalitions dans $\alpha_{i,k}^{cs_{i,s}}$ et réception des acceptations des autres, a_i met à jour son utilité estimée sur la base des acceptations reçues. Si $u_i^e(\alpha_{i,k}) > u_i^\delta$, il envoie des confirmations aux agents qui lui ont envoyé des acceptations et attend de recevoir les leurs.

À chaque nouveau cycle r du protocole MNP, les agents doivent à chaque étape d'échange des propositions, sélectionner une alternative et sa structure de coalitions associée $\alpha_{i,k}^{cs_{i,s}}$ pour formuler leurs propositions. Ainsi, ils procèdent à la mise à jour de leur vue sur l'ensemble de leurs alternatives et les ordonnent en tenant compte des propositions reçues (Algorithme 5). Pour permettre à un agent a_i de dériver les préférences des autres agents à la base de leurs propositions, nous introduisons le concept de *structure de profil* (\mathcal{P}_s).

Définition 4.8. - Structure de profil. Une *structure de profil* $\mathcal{P}_s = [x_1, x_2, \dots, x_j]$ d'un plan π_i est un vecteur de taille $|\mathcal{T}_i|$ où $\forall x_j \in \mathcal{P}_s : \exists t_h \in \mathcal{T}_i : x_j = t_h$.

Le plan de chaque agent est caractérisé par une structure de profil spécifique, qui est le vecteur formé de ses tâches. Par exemple, pour l'agent a_1 (cf. Fig. 3.6) on a : $\mathcal{P}_{s_1} = [t_2, t_{18}, t_7, t_1, t_3, t_5, t_4, t_6]$. Par ailleurs, nous définissons ci-dessous le *profil d'alternative* (propre à chaque alternative), qui indique les tâches d'une alternative qui font partie de la structure de profil d'un plan donné.

Algorithme 3 DecisionMaking(cs_i , Type)

```

1: switch (Type)
2: case Proposal:
3:   if ( $u_i^\epsilon(cs_i) > u_i^\delta$ ) then
4:     return true;
5:   end if
6:   return false;
7: case Accept:
8:    $\mathcal{M}_i \leftarrow \mathcal{M}_i \cup$  les propositions reçues
9:    $u_i^\epsilon(cs_i) \leftarrow$  calculer la nouvelle valeur dans  $r$  en fonction des propositions
    reçues
10:  if ( $u_i^\epsilon(cs_i) > u_i^\delta$ ) then
11:    return true;
12:  end if
13:  return false;
14: case Confirm:
15:   $\mathcal{M}_i \leftarrow \mathcal{M}_i \cup$  les acceptations reçues
16:   $u_i^\epsilon(cs_i) \leftarrow$  calculer nouvelle valeur dans  $r$  en fonctions des acceptations re-
    çues
17:  if ( $u_i^\epsilon(cs_i) > u_i^\delta$ ) then
18:    return true;
19:  end if
20:  return false;
21: default:
22:  return false;
23: end switch

```

Définition 4.9. - Profil d'alternative. Un *profil d'alternative* $P_{\alpha_{i,k}} = [y_1, y_2, \dots, y_j]$ est un vecteur binaire de taille $j = |\mathcal{T}_i|$ défini sur un ensemble de tâches (ex. : les tâches d'une alternative d'un plan $\alpha_{i,k}$) en fonction de la structure de profil Ps_i ,

ou :

$$y_j = \begin{cases} 1 & \text{si : } x_j \in \mathcal{T}_i^{\alpha_i, k} \\ 0 & \text{si : } x_j \notin \mathcal{T}_i^{\alpha_i, k} \end{cases}$$

Par exemple, le profil de l'alternative $\alpha_{1,3} = \langle \{t_1, t_3, t_4, t_6\}, \{t_1 \prec t_3, t_3 \prec t_4, t_4 \prec t_6\} \rangle$ du plan π_1 (cf. Fig. 3.6) est $P_{\alpha_{1,3}} = [0, 0, 0, 1, 1, 0, 1, 1]$.

Chaque alternative α peut être associée à une structure de coalitions cs . Pour simplifier la présentation, nous considérons dans la suite de ce manuscrit qu'un profil de structure de coalitions P_{cs} est le même profil de l'alternative P_α à laquelle il est associé ($P_{cs} = P_\alpha$). En effet, nous considérons seulement cs pour spécifier implicitement à la fois la cs et son alternative associée α . En outre, nous dénotons par P_i l'ensemble de tous les profils des alternatives d'un plan π_i .

Si un agent ne considère que ses propres alternatives pour formuler ses propositions où, à chaque cycle il réitère les mêmes qui ne sont pas acceptées par les autres, les négociations peuvent boucler de manière répétitive sur des alternatives qui seront systématiquement rejetées par les autres agents. Afin d'éviter ce cas de figure, chaque agent doit prendre en considération les propositions des autres pour dériver leurs préférences et en tenir compte dans la formulation de ses propositions futures. De cette manière, il serait possible d'établir une convergence entre les propositions d'un agent et celles désirées par les autres.

Désidérabilité d'une alternative

Nous entendons par désidérabilité d'une alternative le fait qu'un ou plusieurs agents formulent des propositions sur ses tâches. Il serait donc important de la mesurer afin d'en tenir compte dans la formulation des propositions des agents. Nous définissons plusieurs concepts permettant de dériver des mesures intermédiaires nécessaires pour évaluer la désidérabilité des propositions formulées par un agent au regard des autres agents du système. Nous commençons par le concept *vue d'alternative* que nous définissons comme l'ensemble des tâches d'une alternative d'un agent a sur lesquelles il a reçu des propositions d'un autre agent a' .

Définition 4.10. - Vue d'alternative. Une *vue d'alternative* d'un agent a_i qui a envoyé des propositions dans $c_{i,q} \in cs_{i,s}$ par rapport à l'alternative $\alpha_{i,k}$ et a

reçu des propositions $c_{i',q'} \in cs_{i',s'}$ de l'agent $a_{i'}$, est un ensemble de tâches défini comme suit : $v_{\alpha_{i,k}}^{a_{i'}} = \{t | t \in \Gamma(cs_{i,s}) \cap \Gamma(cs_{i',s'}) \wedge \forall c_{i',q'} \in cs_{i',s'} \Rightarrow a_i \in \mathcal{A}_{c_{i',q'}}\}$.

$v_{\alpha_{i,k}}^{a_{i'}}$ est appelée la vue de l'agent $a_{i'}$ sur l'alternative $\alpha_{i,k}$ de l'agent a_i . En d'autres termes, il s'agit de la projection des tâches de $\alpha_{i,k}$ sur l'ensemble des propositions de coalitions $c_{i',q'} \in cs_{i',s'}$ reçues de l'agent $a_{i'}$.

Durant les négociations, un agent a besoin d'estimer le niveau de rapprochement entre une alternative $\alpha_{i,k}$ (et sa structure de coalitions associée $\alpha_{i,k}^{cs_{i,s}}$) avec les propositions reçues des autres agents. Cette valeur de rapprochement est calculée en utilisant la fonction d'empreinte — *fonction d'empreinte* — $\Theta_i^r : P \times P \rightarrow \mathbb{R}^+$ qui est calculée comme suit :

$$\Theta_i^r(P_{\alpha_{i,k}}, P_{v_{\alpha_{i,k}}^{a_{i'}}}) = \sum_{\substack{j=0..|Ps_i| \\ x_j \in P_{\alpha_{i,k}}, x'_j \in P_{v_{\alpha_{i,k}}^{a_{i'}}}}} (x_j \bowtie x'_j) \quad (4.1)$$

où,

$$x_j \bowtie x'_j = \begin{cases} 1, & \text{if : } x_j = x'_j = 1 \\ 1/|Ps_i|, & \text{if : } (x_j = 1 \wedge x'_j = 0) \vee (x_j = 0 \wedge x'_j = 1) \\ 0, & \text{if : } x_j = x'_j = 0 \end{cases}$$

Par exemple, considérons l'alternative $\alpha_{1,3}$ de l'agent a_1 (voir MATP, section 3.3.1). Supposons que a_1 ait envoyé aux autres agents des propositions qui concernent de la structure de coalitions $\alpha_{1,3}^{cs_{1,3}} = \{c_{1,2}, c_{1,4}\}$ durant le cycle $r = 2$. Le profil de l'alternative $\alpha_{1,3}$, est $P_{\alpha_{1,3}} = [0, 0, 0, 1, 1, 0, 1, 1]$. Supposons aussi que a_2 ait envoyé des propositions aux autres concernant son alternative $\alpha_{2,1}$ (où la structure de coalitions associée est : $\alpha_{2,1}^{cs_{2,1}}$). Alors, a_1 a reçu de l'agent a_2 un ensemble de propositions de coalitions concernant les tâches $\{t_1, t_3, t_4\}$ durant le cycle $r = 2$.

Pour l'agent a_1 , la vue d'alternative de l'agent a_2 sur $\alpha_{1,3}$ est $v_{\alpha_{1,3}}^{a_2} = \Gamma(\alpha_{1,3}^{cs_{1,3}}) \cap \Gamma(\alpha_{2,1}^{cs_{2,1}}) = \{t_1, t_3, t_4\}$. Son profil est $P_{v_{\alpha_{1,3}}^{a_2}} = [0, 0, 0, 1, 1, 0, 1, 0]$. Alors, l'empreinte de l'alternative $\alpha_{1,3}$ concernant les propositions de a_2 est :

$$\begin{aligned} \Theta_1^2(P_{\alpha_{1,3}}, P_{v_{\alpha_{1,3}}^{a_2}}) &= [0, 0, 0, 1, 1, 0, 1, 1] \bowtie [0, 0, 0, 1, 1, 0, 1, 0] \\ &= (0 \bowtie 0) + (0 \bowtie 0) + (0 \bowtie 0) + (1 \bowtie 1) + (1 \bowtie 1) + \\ &\quad (0 \bowtie 0) + (1 \bowtie 1) + (1 \bowtie 0) \\ &= 0 + 0 + 0 + 1 + 1 + 0 + 1 + 1/7 = 3.14. \end{aligned}$$

Notons que l'*empreinte* est calculée entre une alternative $\alpha_{i,k}$ des propositions de coalitions envoyées et la vue d'alternative des autres agents sur $\alpha_{i,k}$. Ceci est fait pour chacun des agents α_i à chaque cycle r . Cependant, l'empreinte à elle seule n'exprime pas pleinement l'évolution au fil des cycles. Pour surpasser ce problème, nous introduisons le concept de la *distance transversale* calculée au moyen de la fonction : $\Delta_{\alpha_{i,k}}^r : \mathbb{R}^+ \times \mathbb{R}^+ \longrightarrow \mathbb{R}$.

Cette fonction mesure le changement entre une alternative $\alpha_{i,k}$ et la vue d'alternative des autres agents $\alpha_{i'} \neq \alpha_i$ sur $\alpha_{i,k}$ à travers des cycles consécutifs. Sa valeur est calculée à chaque cycle r en fonction du cycle $(r - 1)$, comme suit :

$$\Delta_{\alpha_{i,k}}^r(\Theta_i^r, \Theta_i^{r-1}) = \sum_{\alpha_{i'} \in \Lambda(\alpha_{i,k}^{cs_{i,k}}), i \neq i'} \left(\Theta_i^r(P_{\alpha_{i,k}}, P_{v_{\alpha_{i,k}}^{\alpha_{i'}}}) - \Theta_i^{r-1}(P_{\alpha_{i,k}}, P_{v_{\alpha_{i,k}}^{\alpha_{i'}}}) \right) \quad (4.2)$$

Cela permet aux agents d'estimer l'évolution de la *distance transversale* entre chaque alternative et les propositions des autres agents. Par exemple, supposons que l'agent α_1 ait reçu un ensemble de propositions de coalitions $c = \{\{\alpha_2, \alpha_1\}, \{t_1\}\}$ de l'agent α_2 durant le cycle $r = 3$ (α_2 veut explorer l'alternative $\alpha_{2,2}$). Pour l'agent α_1 , $P_{v_{\alpha_{1,3}}^{\alpha_2}} = [0, 0, 0, 1, 0, 0, 0, 0]$. Alors, l'empreinte est $\Theta_1^3(P_{\alpha_{1,3}}, P_{v_{\alpha_{1,3}}^{\alpha_2}}) = 1.42$.

Donc, la distance entre $\alpha_{1,3}$ et les propositions de coalitions reçues de l'agent α_2 (lors de passage du cycle $r = 2$ au cycle $r = 3$) est :

$$\begin{aligned} \Delta_{\alpha_{1,3}}^3(\Theta_1^3, \Theta_1^2) &= \Theta_1^3(P_{\alpha_{1,3}}, P_{v_{\alpha_{1,3}}^{\alpha_2}}) - \Theta_1^2(P_{\alpha_{1,3}}, P_{v_{\alpha_{1,3}}^{\alpha_2}}) \\ &= 1.42 - 3.14 = -1.72 \end{aligned}$$

La distance transversale est interprétée comme suit (voir les détails en section 4.3.4) :

$$\text{si } \left\{ \begin{array}{l} \Delta \geq 1 : \text{forte convergence entre les cycles} \\ 0 < \Delta < 1 : \text{faible divergence entre les cycles} \\ 0 > \Delta > -1 : \text{faible convergence entre les cycles} \\ \Delta \leq -1 : \text{forte divergence entre les cycles} \\ \Delta = 0 : \text{pas d'évolution entre les cycles} \end{array} \right.$$

Ainsi, les variations de la distance transversale indiquent différents niveaux de convergence/divergence des négociations, qui reflètent les changements dans le nombre de tâches communes aux alternatives et aux propositions des agents.

Suivant cette interprétation, entre le cycle 2 et le cycle 3, l'agent α_1 a enregistré une forte divergence entre les propositions de α_2 et son alternative $\alpha_{1,3}$. En effet, alors que α_1 a reçu des propositions de α_2 à propos des tâches t_1, t_3, t_4 au cycle $r = 2$, il a reçu (de α_2) des propositions seulement à propos de la tâche t_1 au cycle $r = 3$. Ceci est considéré comme une forte divergence.

Calcul de la désirabilité

La fonction de distance transversale Δ est locale dans le sens où elle se base sur les tâches, les propositions et les cycles propres à l'agent. Il est toutefois utile que les agents aient une vision globale sur tous les agents qui participent aux négociations. En effet, ils doivent mettre à jour l'ordonnancement de leurs alternatives en tenant compte de l'historique des négociations pour refléter les tendances globales. Nous introduisons donc la fonction *désirabilité* $\mathcal{V}_{\alpha_i,k} : (\mathcal{P}_i \times \bar{\mathcal{P}}_i) \rightarrow \mathbf{R}$, avec $\mathcal{P}_i = \{\varepsilon, \omega, \gamma, \rho, \mu\}$ où $\forall e \in \mathcal{P}_i : -1 \leq e \leq 1$ est un ensemble de *paramètres caractéristiques* pour caractériser chaque alternative à chaque cycle de négociations et $\bar{\mathcal{P}}_i = \{\bar{\varepsilon}, \bar{\omega}, \bar{\gamma}, \bar{\rho}, \bar{\mu}\}$ est un ensemble de *paramètres de valorisation* permettant d'attribuer des poids à chacun des paramètres dans \mathcal{P}_i . La valeur de la *désirabilité* est définie comme la somme des puissances comme suit :

$$\mathcal{V}_{\alpha_i,k}(\mathcal{P}_i, \bar{\mathcal{P}}_i) = (\bar{\varepsilon})^\varepsilon + (\bar{\omega})^\omega + (\bar{\gamma})^\gamma + (\bar{\mu})^\mu - (\bar{\rho})^\rho \quad (4.3)$$

Les éléments de \mathcal{P}_i sont des mesures sur des indicateurs caractérisant une alternative durant les négociations selon plusieurs aspects (*Poids de l'utilité estimé* (ε), *Poids d'alternative* (ω), *Poids des tâches* (γ), *Poids des propositions envoyées* (ρ), *Poids de la distance* (μ)) et sont mis à jour avant chaque étape d'échange de propositions. Quant aux éléments de $\bar{\mathcal{P}}_i$ ils sont calculés avant chaque étape d'échange des propositions. Nous décrivons plus en détail ces paramètres en montrons leur utilité dans la suite de ce chapitre.

L'idée est, donc, de permettre à chaque agent d'établir un ordre de préférence sur l'ensemble de ses alternatives avant chaque étape d'échange de propositions d'un cycle r . En effet, \mathcal{P}_i est un ensemble de paramètres ayant des valeurs qui évoluent durant les étapes *Acceptations* et *Confirmations* dans le protocole MNP,

caractérisant ainsi, la dynamique d'évaluations des alternatives. Toutefois, les premières valeurs sont introduites par l'utilisateur avant que les négociations ne commencent. Les valeurs de \mathcal{P}_i sont réinitialisées aux valeurs utilisateur avant chaque étape *Propositions*. En effet, les valeurs de $\bar{\mathcal{P}}_i$ sont mises à jour par la fonction heuristique *IS&C* (cf. Algorithm 4) durant les négociations, et celles de \mathcal{P}_i sont calculées comme suit :

- **Poids de l'utilité estimée** (ε) : est une valeur normalisée ($\varepsilon \in [0, 1[$) de l'utilité estimée des alternatives ($u_i^\varepsilon(\alpha_{i,k})$). Sa valeur est calculée comme suit :

$$\varepsilon_{i,k} = 1 - \frac{\Phi_i^- - u_i^\varepsilon(\alpha_{i,k})}{\Phi_i^{\text{ind}}(\alpha_{i,k})} \quad (4.4)$$

- **Poids des tâches** (γ) : mesure combien les tâches $t \in \mathcal{T}_i^{\alpha_{i,k}}$ sont demandées par les autres agents à travers leurs propositions de coalitions. Sa valeur est calculée comme suit :

$$\gamma_{i,k} = \frac{\sum_{c_{i,q} \in \alpha_{i,k}^{cs_{i,k}}} \sum_{t \in \mathcal{T}_{c_{i,q}}} \frac{\text{frequence}(t)}{|\mathcal{A}_{c_{i,q}}|}}{\sum_{t' \in \mathcal{T}_i} \text{frequence}(t')} \quad (4.5)$$

Où $\text{frequence}(t)$ retourne le nombre de coalitions $c_{i,q}$ parmi toutes les propositions de coalitions reçues qui vérifient la condition : $t \in \mathcal{T}_{c_{i,q}}$.

- **Poids d'alternative** (ω) : mesure l'importance que représente une alternative $\alpha_{i,k}$, par rapport à d'autres alternatives, pour les autres agents. Sa valeur est calculée comme suit :

$$\omega_{i,k} = \frac{\text{reçu}(\alpha_{i,k})}{|\mathcal{L}(\alpha_{i,k}^{cs_{i,s}})|} \quad (4.6)$$

Où $\text{reçu}(\alpha_{i,k})$ retourne le nombre d'agents ayant envoyé des propositions de coalitions $c_{i',q'}$ où $\mathcal{T}_{c_{i',q'}} \cap \mathcal{T}_i^{\alpha_{i,k}} \neq \emptyset$.

Notons que, tandis que γ est calculé sur la base des propositions reçues concernant une tâche t , ω est calculé sur la base des agents qui ont envoyé des propositions à propos de la tâche $t \in \mathcal{T}_i^{\alpha_{i,k}}$. Le premier paramètre renvoie une valeur en fonction du nombre de tâches dans un plan et le second retourne une valeur en fonction du nombre d'agents impliqués dans une structure de coalitions $\mathcal{L}(\alpha_{i,k}^{cs_{i,s}})$.

- **Poids des propositions envoyées** (ρ) : mesure l'importance d'une alternative $\alpha_{i,k}$ pour un agent, compte tenu du nombre de fois où cette alternative a été

sélectionnée pour proposer ses coalitions aux autres. Sa valeur est calculée comme suit :

$$\rho_{i,k} = \frac{\text{envoyé}(\alpha_{i,k})}{r} \quad (4.7)$$

Où $\text{envoyé}(\alpha_{i,k})$ renvoie le nombre de fois où $\alpha_{i,k}$ a été choisi pour envoyer ses propositions de coalitions $c \in \alpha_{i,k}^{cs_i,s}$. Plus une alternative $\alpha_{i,k}$ est sélectionnée, plus la structure de coalitions qui lui est associée $\alpha_{i,k}^{cs_i,s}$ est préférée par l'agent. ρ est utilisé pour ajuster (diminuer) la valeur de la désirabilité $\mathcal{V}_{\alpha_{i,k}}$ afin de permettre d'explorer plus d'alternatives. Il s'agit alors, d'éviter de tomber dans des négociations cycliques où chaque agent propose ses propositions les plus préférées dans $\alpha_i^* \in \mathcal{Q}_i$ de façon répétée, même si α_i^* ne permet pas d'avoir les engagements des autres agents.

- **Poids de la distance** (μ) : est une valeur normalisée de la distance $\Delta_{\alpha_{i,k}}^r$. Sa valeur est calculée comme suit :

$$\mu_{i,k} = \begin{cases} -\frac{\Delta_{\alpha_{i,k}}^r \times |\mathcal{T}^{\alpha_{i,k}}|}{|\mathcal{T}_i|^2}, & \text{si } -1 < \Delta_{\alpha_{i,k}}^r < 0 \text{ ou } 0 < \Delta_{\alpha_{i,k}}^r < 1 \\ \frac{\Delta_{\alpha_{i,k}}^r \times |\mathcal{T}^{\alpha_{i,k}}|}{|\mathcal{T}_i|^2}, & \text{sinon} \end{cases} \quad (4.8)$$

Dans MNP, au début des négociations (cycle $r = 1$), chacun des agents ordonne ces alternatives en utilisant l'utilité estimée u_i^e .

À partir de $r > 1$, avant chaque étape d'échange de propositions, les agents ordonnent leurs alternatives sur la base des valeurs de leurs désirabilités $\mathcal{V}_{\alpha_{i,k}}$, de la plus grande à la plus petite.

La valeur de $\mathcal{V}_{\alpha_{i,k}}$ sur une alternative $\alpha_{i,k}$ permet de combiner d'une part, les préférences de l'agent α_i sur $\alpha_{i,k}$ et d'autre part, la désirabilité de $\alpha_{i,k}$ par les autres agents.

Ordonnement des alternatives

Rappelons que $\mathcal{V}_{\alpha_{i,k}}$ est calculée sur la base des paramètres \mathcal{P}_i et $\bar{\mathcal{P}}_i$. Les paramètres $\bar{\mathcal{P}}_i$ permettent aux agents de réagir dynamiquement aux tendances qui apparaissent dans les négociations. Ils sont de deux catégories :

- **Déterminants** : Leurs valeurs n'évoluent pas au cours des négociations :
 - $\bar{\varepsilon}$: valorisation du *Poids de l'utilité estimée* (ε).
 - $\bar{\omega}$: valorisation du *Poids d'alternative* (ω).

- **Commutateurs** : Leurs valeurs évoluent au cours des négociations durant chaque cycle. Ils peuvent avoir un impact important sur les négociations, car ils impactent la valeur de \mathcal{V} . Ils sont initialisés à leurs valeurs initiales telles qu'introduites par l'utilisateur au début de chaque nouveau cycle :
 - $\bar{\gamma}$: valorisation du *Poids des tâches* (γ).
 - $\bar{\rho}$: valorisation du *Poids des propositions envoyées* (ρ).
 - $\bar{\mu}$: valorisation du *Poids de la distance* (μ).

Le rôle des *commutateurs* est de permettre aux agents de passer d'une alternative à l'autre en modifiant la valeur de \mathcal{V} . Ils sont calculés à chaque cycle en utilisant l'heuristique *IS&C* (cf. Algorithme 4) selon Δ . *IS&C* considère la distance transversale entre l'alternative souhaitée de chaque agent et les coalitions proposées par les autres. En effet, ils sont utilisés pour refléter la tendance des négociations et permettre aux agents de faire des concessions sur leurs exigences en établissant un ordre de préférence (ordonnancement) sur \mathcal{Q}_i qui tient compte des préférences dérivées des autres agents⁸.

8. Notons que si $\Delta \neq 0$, la valeur de la désirabilité $\mathcal{V}\alpha_{i,k}$ serait principalement déterminée par les *commutateurs*.

Algorithme 4 `IterativeSearchConverge`($\mathcal{P}_i, \bar{\mathcal{P}}_i, \Delta$)

```

1: switch ( $\Delta$ )
2: case ( $\Delta \geq 1$ ):
3:    $\bar{\mu} = \bar{\mu} + \bar{\mu} * \Delta$ ;
4: case ( $\Delta > 0 \ \& \ \Delta < 1$ ):
5:    $\bar{\gamma} = \bar{\gamma} + (\bar{\mu})^\mu$ ;
6: case ( $\Delta == 0$ ):
7:   if  $\bar{\mu} > 1$  then
8:      $\bar{\mu} = 1$ ;
9:   end if
10:   $\bar{\rho} = \bar{\rho} + (\bar{\rho} * \log(\text{passedTime}))$ ;
11: case ( $\Delta < 0 \ \& \ \Delta > -1$ ):
12:   $\bar{\mu} = \bar{\mu} + (\bar{\mu} * (-\Delta))$ ;
13: case ( $\Delta \leq -1$ ) :
14:   $\bar{\rho} = \bar{\rho} + (\bar{\mu})^\mu$ ;
15:   $\bar{\gamma} = \bar{\gamma} + (\bar{\mu})^\mu$ ;
16:   $\bar{\mu} = 1$ ;
17: end switch

```

Les paramètres dans $\bar{\mathcal{P}}_i$ permettent alors aux agents d'établir un nouvel ordre de préférence centré sur certains aspects des négociations. Si un paramètre de valorisation particulier est significativement plus grand que les autres, la probabilité que l'agent sélectionne des structures de coalitions dans lesquelles ce paramètre se manifeste augmente en conséquence. Par exemple, si Tom veut trouver une solution à faible coût, il peut introduire une valeur plus élevée pour le paramètre de valorisation $\bar{\varepsilon}$ ⁹.

9. Notons qu'il y a un certain ratio à respecter, selon la nature des scénarios d'application. Par exemple, avoir une valeur plus élevée de $\bar{\varepsilon}$ pourrait amener les agents à ne négocier que sur les alternatives les plus bénéfiques individuellement, car les agents ne feront pas de concessions significatives. Dans ce cas, une valeur suffisamment élevée du paramètre de valorisation $\bar{\rho}$ peut obliger les agents à faire des concessions et alterner les propositions qu'ils envoient pour explorer davantage de possibilités.

Comme illustré dans l'Algorithme 4, l'heuristique $\mathcal{IS\&C}$ met à jour les valeurs des paramètres *commutateurs* en fonction de la distance transversale Δ . L'interprétation des valeurs de Δ pour un agent α_i , concernant son alternative α vis-à-vis d'un agent $\alpha_{i'}$ se fait suivant cinq cas :

1. $\Delta \geq 1$ (forte convergence) : $\exists t \in \alpha$ qui n'était pas proposée par $\alpha_{i'}$ dans $r-1$, mais qu'il a proposé dans r ; alors la valeur du paramètre de valorisation du *poids de la distance* $\bar{\mu}$ est augmentée comme une tentative de maintenir α parmi les alternatives les mieux classées dans l'ordre de préférence établi par α dans le cycle courant r . Par ce comportement, α veut conserver la convergence des propositions des agents sur l'alternative α pour aboutir à une structure de coalitions acceptée. En effet, à l'issue de l'heuristique $\mathcal{IS\&C}$, $\bar{\mu}$ sera le paramètre avec la valeur la plus élevée dans \mathcal{V} . Ceci devrait permettre à l'agent α d'explorer la convergence enregistrée. Des valeurs successives où $\Delta > 1$ signifient qu'à chaque nouveau cycle r , il y a au moins une autre nouvelle tâche dans α qui est désirée (proposée) par $\alpha_{i'}$ comparativement à $r-1$.
2. $0 < \Delta < 1$ (faible divergence) : $\exists t \notin \alpha$ qui était proposée par $\alpha_{i'}$ dans $r-1$, mais qu'il n'a pas proposé dans r ; alors l'importance du paramètre de valorisation *Poids des tâches* $\bar{\gamma}$ est augmentée. En effet, comme il y a une faible divergence et que α n'a pas permis d'aboutir à une solution, l'augmentation de la valeur de $\bar{\gamma}$ va faire augmenter la désirabilité des autres alternatives différentes de α . Ce comportement est une tentative d'évoluer vers d'autres propositions aux prochains cycles.
3. $\Delta = 0$ (pas d'évolution dans les négociations) : l'effet du *Poids de la distance* ($\mu = 1$) est annulé et la valeur du *poids des propositions envoyées* est augmentée¹⁰ pour devenir la valeur la plus importante dans le calcul de \mathcal{V} , proportionnellement au temps écoulé dans les négociations.

10. La valeur du paramètre *poids des propositions envoyées* sera soustraite de la valeur de $\mathcal{V}_{\alpha_i, k}$ pour faire diminuer l'indicateur d'acceptation des alternatives qui sont envoyées plus fréquemment. En faisant ainsi, les agents auront un comportement explorateur sur l'ensemble de leur alternative pour tenter de trouver une structure de coalitions comme solution.

4. $-1 < \Delta < 0$ (faible convergence) : $\exists t \notin \alpha$ qui n'était pas proposée par α_i dans $r-1$, mais qu'il a proposé dans r (avoir une nouvelle tâche t proposée par α_i même si $t \notin \alpha$ peut rendre une autre alternative α' plus proche des propositions de l'agent α_i durant les cycles à venir); alors, seule la valeur de $\bar{\mu}$ est mise à jour. Ce comportement vise à maintenir la convergence enregistrée dans les prochains cycles.
5. $\Delta \leq -1$ (forte divergence) : $\exists t \in \alpha$ qui était proposée par α_i dans $r-1$ mais qu'il n'a pas proposé dans r ; alors les valeurs de $\bar{\rho}$ et $\bar{\gamma}$ sont augmentées de la même manière pour augmenter les valeurs des mesures du *Poids des tâches* et du *Poids des propositions envoyées*, tandis que le *poids de la distance* est ramené à sa valeur initiale en réinitialisant $\bar{\mu} = 1$. Ce comportement vise à amener les agents à explorer d'autres alternatives. En effet, en augmentant γ , les alternatives désirées par les autres agents seront favorisées et en augmentant ρ les propositions envoyées par α_i perdront du poids dans son nouvel ordre de préférence. Des valeurs successives $\Delta < -1$ signifient qu'à chaque nouveau cycle r il y a au moins une autre nouvelle tâche de l'alternative α qui n'est pas désirée par un autre agent par rapport à $r-1$.

4.3.5 Raffinement individuel de la solution

À la fin des négociations, le résultat de CFDEA est une structure de coalitions (et implicitement, une alternative) dénotée \hat{cs}_i . Un agent décide d'arrêter les négociations si tous les engagements nécessaires concernant une structure de coalitions $cs_{i,s}$ ont été obtenus, c'est-à-dire : $\bigcup_{c_{i,q} \in cs_{i,s}} \mathcal{A}_{c_{i,q}} = \bigcup_{c_{i,q} \in cs_{i,s}} \mathcal{A}_{\theta_{c_{i,q}}}$ ou le temps limite τ est atteint. Par ailleurs, les structures de coalitions dans *l'espace de recherche* \mathcal{S}_i peuvent évoluer tout au long des négociations.

Évolution des structures de coalitions

Définition 4.11. - État initial. Une structure de coalitions $cs_{i,s}$ est dans un *état initial* si : $\forall c_{i,q} \in cs_{i,s}, \exists \mathcal{T}_i^c \langle \mathcal{A}^c, \mathcal{T}_{\mathcal{A}^c}^c \rangle : (\mathcal{A}_{c_{i,k}} = \mathcal{A}^c) \wedge (\mathcal{T}_{c_{i,q}} \subseteq \mathcal{T}_{\mathcal{A}^c}^c)$.

Algorithme 5 Scheduler(\mathcal{Q}_i, r)

```

1:
2: if  $r == 1$  then
3:   for all  $\alpha_{i,k} \in \mathcal{Q}_i$  do
4:      $u_i^\varepsilon(\alpha_{i,k}) \leftarrow$  calculer la nouvelle valeur dans  $r$ 
5:   end for
6:    $\mathcal{Q}_i \leftarrow$  ordonner  $\mathcal{Q}_i$  par ordre décroissant de  $u_i^\varepsilon$ 
7:    $\alpha_i^* \leftarrow$  meilleure alternative dans  $\mathcal{Q}_i$ 
8:   return  $cs_i$  associée à  $\alpha_i^*$ 
9: else
10:  for all  $\alpha_{i,k} \in \mathcal{Q}_i$  do
11:     $P_{v_{\alpha_{i,k}}}^{a_i'}$   $\leftarrow$  calculer la nouvelle valeur dans  $r$ 
12:     $\Theta_i^r(P_{\alpha_{i,k}}, P_{v_{\alpha_{i,k}}}^{a_i'}) \leftarrow$  calculer la nouvelle valeur dans  $r$ 
13:     $\Delta_{\alpha_{i,k}}^r(\Theta_i^r, \Theta_i^{r-1}) \leftarrow$  calculer la nouvelle valeur dans  $r$ 
14:     $\mathcal{IS\&C\_heuristic}(P_i, \bar{P}_i, \Delta_{\alpha_{i,k}}^r)$  : calculer les nouvelles valeurs des para-
        mètres dans  $P_i$  et  $\bar{P}_i$ 
15:     $\mathcal{V}_{\alpha_{i,k}}(P_i, \bar{P}_i) \leftarrow$  calculer nouvelle valeur dans  $r$ 
16:  end for
17:   $\mathcal{Q}_i \leftarrow$  ordonner  $\mathcal{Q}_i$  par ordre décroissant de  $\mathcal{V}_{\alpha_{i,k}}$ 
18:   $\alpha_i \leftarrow$  meilleure alternative dans  $\mathcal{Q}_i$ 
19:  return  $cs_i$  associée à  $\alpha_i$ 
20: end if
    
```

Par exemple, considérons la structure de coalitions $cs_{1,3} = \{\langle\{a_1, a_2\}, \{t_1, t_3\}\rangle, \langle\{a_1, a_2, a_4\}, \{t_4\}\rangle, \langle\{a_1, a_3\}, \{t_7\}\rangle\}$. Elle se trouve dans un état initial puisqu'il existe un ensemble commun $\mathcal{T}_1^c = \{\langle\{a_1, a_2\}, \{t_1, t_3\}\rangle, \langle\{a_1, a_2, a_4\}, \{t_4\}\rangle, \langle\{a_1, a_3, a_4\}, \{t_5\}\rangle, \langle\{a_1, a_3\}, \{t_7\}\rangle\}$ où : $\forall c \in cs_{1,3}, \exists \langle\mathcal{A}^c, \mathcal{T}_{\mathcal{A}^c}^c\rangle : (\mathcal{A}_c = \mathcal{A}^c) \wedge (\mathcal{T}_c \subseteq \mathcal{T}_{\mathcal{A}^c}^c)$. Initialement, les structures de coalitions dans \mathcal{S}_i d'un agent a_i sont toutes dans leur *état initial*. Nous dénotons l'ensemble de toutes les structures de coalitions dans leur *état initial* par \mathcal{S}_i^* . Pendant les négociations, les transitions de phase de la forme *Propositions* \rightarrow *Acceptations* et *Acceptations* \rightarrow *Confirmations* conduisent à l'évolution sur l'en-

semble des structures de coalitions sélectionnées. Par conséquent, une structure sélectionnée $cs_{i,s}$ peut avoir moins d'agents impliqués que dans son *état initial*. En effet, il peut y avoir des agents qui n'ont pas accepté ou confirmé les propositions dans $cs_{i,s}$ durant les deux dernières étapes du protocole MNP. Alors, $cs_{i,s}$ devient dans un *état intermédiaire* que nous dénotons $cs_{i,s}^{inter}$.

Définition 4.12. - État intermédiaire. Une structure de coalitions $cs_{i,s}$ est dans un *état intermédiaire* $cs_{i,s}^{inter}$ si : $\forall c_{i,q} \in cs_{i,s} : \exists \mathcal{T}_i^c \Rightarrow (\mathcal{A}_{c_{i,q}} \subset \mathcal{A}^c) \wedge (\mathcal{T}_{c_{i,q}} \subseteq \mathcal{T}_{\mathcal{A}^c}^c)$.

Par exemple, un état intermédiaire de la structure de coalitions $cs_{1,3}$ peut-être : $cs_{1,3}^{inter} = \{\{c_{1,1} = \langle \{a_1, a_2\}, \{t_1, t_3\}, \{t_1 \prec t_3\} \rangle, c_{1,2} = \langle \{a_1, a_2, a_4\}, \{t_4\}, \{\} \rangle, c_{1,6} = \langle \{a_1\}, \{t_7\}, \{\} \rangle\}$. Notons que l'agent a_3 qui est dans la coalition $c_{1,4}$ n'est pas présent dans $c_{1,6}$. Ainsi, une structure de coalitions peut avoir plusieurs états intermédiaires¹¹. Nous dénotons l'ensemble de toutes les structures dans leur *état intermédiaire* par \mathcal{S}_i^+ . Si aucun agent n'est intéressé par les coalitions dans $cs_{i,s}$ pendant les étapes *Acceptations* et *Confirmations*, $cs_{i,s}$ est dite dans un *état atomique* que nous dénotons $cs_{i,s}^{atom}$.

Définition 4.13. - État atomique. Une structure de coalitions $cs_{i,s}$ d'un agent a_i est dans un *état atomique* $cs_{i,s}^{atom}$ si : $\forall c_{i,q} \in cs_{i,s} : \mathcal{A}_{c_{i,q}} = \{a_i\}$.

Par exemple, l'état atomique d'une structure de coalitions $cs_{1,3}$ est : $cs_{1,3}^{atom} = \{\{c_{1,7} = \langle \{a_1\}, \{t_1, t_3\}, \{t_1 \prec t_3\} \rangle, c_{1,8} = \langle \{a_1\}, \{t_4\}, \{\} \rangle, c_{1,9} = \langle \{a_1\}, \{t_7\}, \{\} \rangle\}$. Lorsqu'une structure de coalitions passe d'un état initial à un état atomique, cela veut dire que les agents figurant dans son état initial ne sont pas intéressés par ses coalitions.

Sélection de l'alternative finale

À la fin des négociations, l'alternative sélectionnée (et la structure de coalitions correspondante \hat{cs}_i) pourrait être dans un des cas suivants :

11. Durant les phases *Acceptations* et *Confirmations*, il peut y avoir des agents qui ne sont pas intéressés par cs . L'agent émetteur en tient compte en mettant à jour sa structure de coalitions en conséquence et n'envoie pas d'acceptations ou de confirmations aux agents non intéressés.

Pas d'alternative sélectionnée : Le temps limite des négociations est atteint et α_i n'a pas d'engagements mutuels avec les autres agents sur aucune des coalitions. Alors, $\forall c_{i,s} \in \mathcal{S}_i, \forall c_{i,q} \in \mathcal{C}_{i,s} : |\mathcal{A}_{\theta_{c_{i,q}}}| = 0$. Dans ce cas, l'agent va sélectionner l'alternative qui a un coût ϕ_i^- le plus faible.

Alternative partiellement sélectionnée : Le temps limite des négociations est atteint et $\hat{\mathcal{C}}_i$ vérifie : $\exists c_{i,q} \in \hat{\mathcal{C}}_i : |\mathcal{A}_{\theta_{c_{i,q}}}| \geq 1$. Dans ce cas, l'alternative à retenir sera celle avec le plus faible coût $\phi_i(\alpha_{i,k})$ qui vérifie $\forall c_{i,q} \in \hat{\mathcal{C}}_i : |\mathcal{A}_{\theta_{c_{i,q}}}| \geq 1 \Rightarrow \mathcal{T}_{c_{i,q}} \subseteq \mathcal{T}_i^{\alpha_{i,k}}$, *c.-à-d.* la solution finale représente une alternative où sa structure de coalitions associée est acceptée dans un état intermédiaire.

Alternative entièrement sélectionnée : À la fin des négociations, la structure de coalitions sélectionnée $\hat{\mathcal{C}}_i \in \mathcal{S}_i^*$ est dans son *état initial*. Alors, $\forall c_{i,q} \in \hat{\mathcal{C}}_i \Rightarrow |\mathcal{A}_{\theta_{c_{i,q}}}| = |\mathcal{A}_{c_{i,q}}|$. Donc, l'alternative finale $\alpha_{i,k}$ vérifie $\forall c_{i,q} \in \hat{\mathcal{C}}_i : \mathcal{T}_{c_{i,q}} \subseteq \mathcal{T}_i^{\alpha_{i,k}}$. Notons que dans ce cas le temps limite des négociations peut ne pas être atteint.

4.4 Scénario illustratif

Pour illustrer le fonctionnement de CFDEA, nous décrivons dans cette section un scénario illustratif basé sur les plans de la figure Fig. 3.6. Nous commençons par la présentation des paramètres relatifs au scénario puis nous enchaînons par la présentation de différentes phases de négociations entre les agents.

4.4.1 Paramètres

En plus de ce que nous avons introduit dans la description de MATP, nous dénotons par $\phi_i^{\text{ind}}(\alpha_{i,k}) = \sum_{t \in \mathcal{T}_i^{\alpha_{i,k}}} \phi_i(t)$ le coût individuel d'une alternative pour un agent α_i s'il se déplace seul de v_i^s à v_i^d suivant $\alpha_{i,k}$. Le coût minimum individuel ϕ_i^- pour α_i est le coût de référence qu'il utilisera pour calculer son utilité estimée (u_i^ϵ) et son utilité de référence (u_i^δ). Chaque agent peut calculer une estimation sur l'utilité de chacune de ses alternatives, comme suit : $u_i^\epsilon(\alpha_{i,k}) = \phi_i^- - \phi_i^\epsilon(\alpha_{i,k})$, où $\phi_i^\epsilon(\alpha_{i,k}) = \sum_{t \in \mathcal{T}_i^{\alpha_{i,k}}} \left(\frac{\phi^{\text{pub}}(t)}{n} + \phi_i^{\text{priv}}(t) \right)$ est le coût estimé de $\alpha_{i,k}$ et n est

le nombre d'agents qui peuvent réaliser conjointement \mathbf{t} . $\phi_i^\epsilon(\alpha_{i,k})$ est le coût de l'alternative $\alpha_{i,k}$ si un α_i se déplace conjointement avec d'autres agents suivant $\alpha_{i,k}$. Nous supposons que chaque agent a un coût de réservation (ϕ_i^{res}) qu'il ne peut pas dépasser durant les négociations. ϕ_i^{res} est obtenu en fonction d'un facteur de coût limite $\phi_i^{\text{limit}} \in [0, 1]$ défini par l'utilisateur lui-même. Alors, $\phi_i^{\text{res}} = \phi_i^- \times \phi_i^{\text{limit}}$. L'utilité de référence u_i^δ d'un agent α_i est $u_i^\delta = \phi_i^- - \phi_i^{\text{res}}$. ϕ_i^{limit} sert comme un moyen d'ajustage sur la valeur de l'utilité estimée, pour au final améliorer la valeur finale de l'utilité obtenue. Cependant une valeur trop petite de ϕ_i^{limit} peut empêcher α_i de conclure des accords. Une alternative $\alpha_{i,k}$ sera considérée dans les négociations seulement si elle vérifie la condition : $\phi_i^\epsilon(\alpha_{i,k}) < \phi_i^{\text{res}}$ (où $u_i^\epsilon(\alpha_{i,k}) > u_i^\delta$). Une faible valeur de ϕ_i^{limit} mène α_i à rechercher une valeur plus élevée de son utilité, et vice-versa.

4.4.2 Phase d'initialisation

Après l'étape d'*échange de plans*, chaque agent α_i calcule toutes ses alternatives $\alpha_{i,k} \in \mathcal{Q}_i$, ses tâches communes $\mathbf{t}^c \in \mathcal{T}_i^c$ et ses ensembles communs $\mathcal{T}_i^{\bar{c}}$ puis calcule son ensemble de structures de coalitions $\mathcal{CS}_i \in \mathcal{S}_i$. Ensuite, il calcule tous ses coûts individuels $\phi_i^{\text{ind}}(\alpha_{i,k=1,\dots,K_i})$, les coûts estimés $\phi_i^\epsilon(\alpha_{i,k=1,\dots,K_i})$, les valeurs de l'utilité estimée de ses alternatives $u_i^\epsilon(\alpha_{i,k=1,\dots,K_i})$, les valeurs de son coût de réservation et, ensuite, initialise son utilité de référence u_i^δ . Le tableau Tab. 4.3 illustre l'ensemble des coalitions possibles qui forment les structures de coalitions (possibles) dans leur état initial pour chacune des alternatives de chaque agent. Le tableau Tab. 4.4 illustre les valeurs calculées des paramètres présentés ci-dessus, par alternative de chacun des agents. Ici, nous fixons les valeurs initiales des paramètres de valorisation comme suit : $\bar{\mathcal{P}}_i = \{\bar{\epsilon} = 20, \bar{\omega} = 2, \bar{\gamma} = 5, \bar{\rho} = 7, \bar{\mu} = 10\}$.

Agents	α	Coalitions possibles	Structures de coalitions
a_1	$\alpha_{1,1}$	$c_{1,1} = \langle \{a_1, a_3\}, \{t_7\} \rangle$	$cs_{1,1} = \{c_{1,1}\}$
	$\alpha_{1,2}$	$c_{1,2} = \langle \{a_1, a_2\}, \{t_1, t_3\} \rangle$, $c_{1,3} = \langle \{a_1, a_3, a_4\}, \{t_5\} \rangle$, $c_{1,4} = \langle \{a_1, a_3\}, \{t_7\} \rangle$	$cs_{1,2} = \{c_{1,2}, c_{1,3}, c_{1,4}\}$
	$\alpha_{1,3}$	$c_{1,2} = \langle \{a_1, a_2\}, \{t_1, t_3\} \rangle$, $c_{1,4} = \langle \{a_1, a_2, a_4\}, \{t_4\} \rangle$	$cs_{1,3} = \{c_{1,2}, c_{1,4}\}$
a_2	$\alpha_{2,1}$	$c_{2,1} = \langle \{a_2, a_1\}, \{t_1, t_3\} \rangle$, $c_{2,2} = \langle \{a_2, a_1, a_4\}, \{t_4\} \rangle$, $c_{2,3} = \langle \{a_2, a_4\}, \{t_8\} \rangle$	$cs_{2,1} = \{c_{2,1}, c_{2,2}, c_{2,3}\}$
	$\alpha_{2,2}$	$c_{2,4} = \langle \{a_2, a_1\}, \{t_1\} \rangle$, $c_{2,5} = \langle \{a_2, a_3\}, \{t_9\} \rangle$, $c_{2,6} = \langle \{a_2, a_4\}, \{t_{10}\} \rangle$	$cs_{2,2} = \{c_{2,4}\}$
a_3	$\alpha_{3,1}$	$c_{3,1} = \langle \{a_3, a_2\}, \{t_9\} \rangle$	$cs_{3,1} = \{c_{3,1}\}$
	$\alpha_{3,2}$	$c_{3,2} = \langle \{a_3, a_4\}, \{t_{12}\} \rangle$, $c_{3,3} = \langle \{a_3, a_1, a_4\}, \{t_5\} \rangle$, $c_{3,4} = \langle \{a_3, a_1\}, \{t_7\} \rangle$	$cs_{3,2} = \{c_{3,2}, c_{3,3}, c_{3,4}\}$
a_4	$\alpha_{4,1}$	$c_{4,1} = \langle \{a_4, a_3\}, \{t_{12}\} \rangle$, $c_{4,2} = \langle \{a_4, a_1, a_3\}, \{t_5\} \rangle$	$cs_{4,1} = \{c_{4,1}, c_{4,2}\}$
	$\alpha_{4,2}$	$c_{4,1} = \langle \{a_4, a_3\}, \{t_{12}\} \rangle$, $c_{4,3} = \langle \{a_4, a_1, a_2\}, \{t_4\} \rangle$, $c_{4,4} = \langle \{a_4, a_2\}, \{t_8\} \rangle$	$cs_{4,2} = \{c_{4,1}, c_{4,3}, c_{4,4}\}$
	$\alpha_{4,3}$	$c_{4,5} = \langle \{a_4, a_2\}, \{t_{10}\} \rangle$	$cs_{4,3} = \{c_{4,5}\}$

Tab. 4.3 – Les structures de coalitions possibles (dans leur état initial).

4.4.3 Phase des négociations

Rappelons que pour s'engager dans les négociations, chaque agent choisit l'alternative ayant la valeur u_i^e la plus élevée pour transmettre ses propositions aux autres. Dans le premier cycle des négociations, les agents procèdent à l'ordonnement de leurs alternatives sur la base de leurs utilités estimées. À partir du deuxième cycle, ils les ordonnent sur la base de la valeur de leur *désirabilité* (\mathcal{V}). La figure Fig. 4.4 constitue une illustration graphique du premier cycle r_1 des négociations (avec les valeurs du tableau Tab. 4.4) et le cycle r_2 (avec les valeurs des deux tableaux : Tab. 4.5 et Tab. 4.6).

Agents	Alternatives	ϕ^{ind}	ϕ^ϵ	u^ϵ	ϕ^{res}	ϕ^-	u^δ
Tom (\mathbf{a}_1)	$\alpha_{1,1}$	130	115.0	15	117.00	130	13.00
	$\alpha_{1,2}$	165	103.33	26.67			
	$\alpha_{1,3}$	150	116.66	13.34			
Lisa (\mathbf{a}_2)	$\alpha_{2,1}$	145	94.16	45.84	126.00	140	14.00
	$\alpha_{2,2}$	140	92.5	47.5			
Mark (\mathbf{a}_3)	$\alpha_{3,1}$	140	117.5	22.5	126.00	140	14.00
	$\alpha_{3,2}$	140	88.33	51.67			
Alice (\mathbf{a}_4)	$\alpha_{4,1}$	140	103.33	21.67	112.50	125	12.5
	$\alpha_{4,2}$	140	99.16	25.84			
	$\alpha_{4,3}$	125	110.0	15			

Tab. 4.4 – Les coûts des alternatives.

Avant d'entamer les négociations, chaque agent trie ses alternatives par ordre décroissant de leur utilité estimée. Ceci est illustré dans la Fig. 4.4 où les valeurs les plus élevées sont représentées par des cercles verts. De même, les deuxièmes valeurs les plus élevées sont en bleu et les troisièmes en rouge. En effet, durant la première étape de r_1 (cf. Fig. 4.4-a), \mathbf{a}_1 choisit son alternative $\alpha_{1,2}$, \mathbf{a}_2 choisit $\alpha_{2,2}$, \mathbf{a}_3 choisit $\alpha_{3,2}$ et \mathbf{a}_4 choisit $\alpha_{4,2}$. Puis, chacun envoie les propositions de coalitions \mathbf{c} de la structure de coalitions \mathbf{cs} associée à son alternative choisie à leurs agents respectifs (chaque proposition de coalition \mathbf{c} aux agents $\mathcal{A}_{\mathbf{c}}$). Les propositions envoyées sont en cercle vert dans la Fig. 4.4. Par exemple, \mathbf{a}_1 envoie la proposition $\mathbf{c}_{1,2}$ à \mathbf{a}_2 et la proposition $\mathbf{c}_{1,3}$ à \mathbf{a}_3 et \mathbf{a}_4 .

À la deuxième étape de r_1 (cf. Fig. 4.4-b), chaque agent recalcule l'utilité estimée de l'alternative sélectionnée en fonction des propositions reçues. Par exemple, l'agent \mathbf{a}_3 a reçu des propositions concernant les tâches $\mathbf{t}_5, \mathbf{t}_7$ de \mathbf{a}_1 et \mathbf{t}_{12} de \mathbf{a}_4 . Par conséquent, il met à jour son coût estimé et l'utilité estimée de son alternative sélectionnée $\alpha_{3,2}$, c'est-à-dire $\phi_3^\epsilon(\alpha_{3,2}) = (10 + 5) + (\frac{20}{2} + 15) + (\frac{40}{2} + 15) + (\frac{30}{2} + 5) = 95$. Par conséquent, la nouvelle valeur de son utilité estimée est $u_3^\epsilon(\alpha_{3,2}) = \phi_3^- - \phi_3^\epsilon(\alpha_{3,2}) = 140 - 95 = 45$ qui reste une valeur satisfaisante pour la

condition $u_3^\epsilon > u_3^\delta$ ($45 > 14$). Ainsi, \mathbf{a}_3 envoie des acceptations à \mathbf{a}_4 concernant $\mathbf{c}_{4,1}$ et à \mathbf{a}_1 concernant $\mathbf{c}_{1,3}$ et $\mathbf{c}_{1,1}$. Notons que la structure de coalitions sélectionnée $\alpha_{3,2}^{cs_{3,2}}$ de \mathbf{a}_3 (cf. Fig. 4.4-b) est à ce niveau dans un état intermédiaire (cf. Déf. 4.12), c'est-à-dire, $\alpha_{3,2}^{cs_{3,2}}$ devient $\alpha_{3,2}^{cs_{3,2}^{inter}} = \{ \mathbf{c}_{3,2} = \langle \{\mathbf{a}_3, \mathbf{a}_4\}, \{\mathbf{t}_{12}\} \rangle, \mathbf{c}_{3,3} = \langle \{\mathbf{a}_3, \mathbf{a}_1\}, \{\mathbf{t}_5\} \rangle, \mathbf{c}_{3,4} = \langle \{\mathbf{a}_3, \mathbf{a}_1\}, \{\mathbf{t}_7\} \rangle \}$. En effet, \mathbf{a}_4 n'est plus dans la proposition de coalition, $\mathbf{c}_{3,3}$ car \mathbf{a}_3 n'a pas reçu de proposition concernant $\mathbf{t}_5 \in \mathcal{T}_{\mathbf{c}_{3,3}}$ (cf. Fig. 4.4-a). Les agents $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_4$ n'enverront pas d'acceptations, car la condition $u^\epsilon > u^\delta$ n'est plus vérifiée à l'étape d'acceptations (cf. Fig. 4.4-b).

À la troisième étape de r_1 (cf. Fig. 4.4-c), aucune confirmation n'est échangée entre les agents. Même si \mathbf{a}_3 a envoyé des messages d'acceptations à $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_4\}$, \mathbf{a}_3 n'a reçu aucun message d'acceptation de leur part. Ainsi, son nouveau coût estimé est réévalué comme suit : $\phi_3^\epsilon(\alpha_{3,2}) = (10+5) + (\frac{20}{1}+15) + (\frac{40}{1}+15) + (\frac{30}{1}+5) = 140$. Par conséquent, son utilité estimée devient : $u_3^\epsilon(\alpha_{3,2}) = \phi_3^- - \phi_3^\epsilon(\alpha_{3,2}) = 140 - 140 = 0$; ce qui fait que $u_3^\epsilon(\alpha_{3,2}) > u_3^\delta$ devient à cette étape, une condition non vérifiée. De ce fait, tous les agents passeront au cycle r_2 sans avoir eu d'accords entre eux.

Avant d'entamer l'étape d'échange des propositions durant le cycle r_2 , chacun et sur la base des informations collectées durant le cycle r_1 réordonne ses alternatives selon les valeurs de sa fonction *désirabilité* \mathcal{V} . Les tableaux Tab. 4.5 et Tab. 4.6 illustrent les valeurs des *paramètres de caractérisation*, des *paramètres de valorisation* et de la *désirabilité* ainsi que l'ordre des alternatives établi par chacun des agents. Ces valeurs concernent 5 cycles de négociations nécessaires dans ce scénario pour trouver une solution au problème de la formation de coalitions.

La figure Fig. 4.4-d, montre le nouvel ordre d'alternatives (et implicitement, l'ordre des structures de coalitions associées) pour r_2 . Par exemple, durant le cycle r_1 l'agent \mathbf{a}_1 a sélectionné $\alpha_{1,2}$. Puis, durant le cycle r_2 il a sélectionné l'alternative $\alpha_{1,3}$ suivant la valeur la plus élevée de \mathcal{V} (cf. Tab. 4.5). Après ré-ordonnement, \mathbf{a}_2 a sélectionné son alternative $\alpha_{2,1}$, \mathbf{a}_3 son alternative $\alpha_{3,1}$ et \mathbf{a}_4 son alternative $\alpha_{4,1}$.

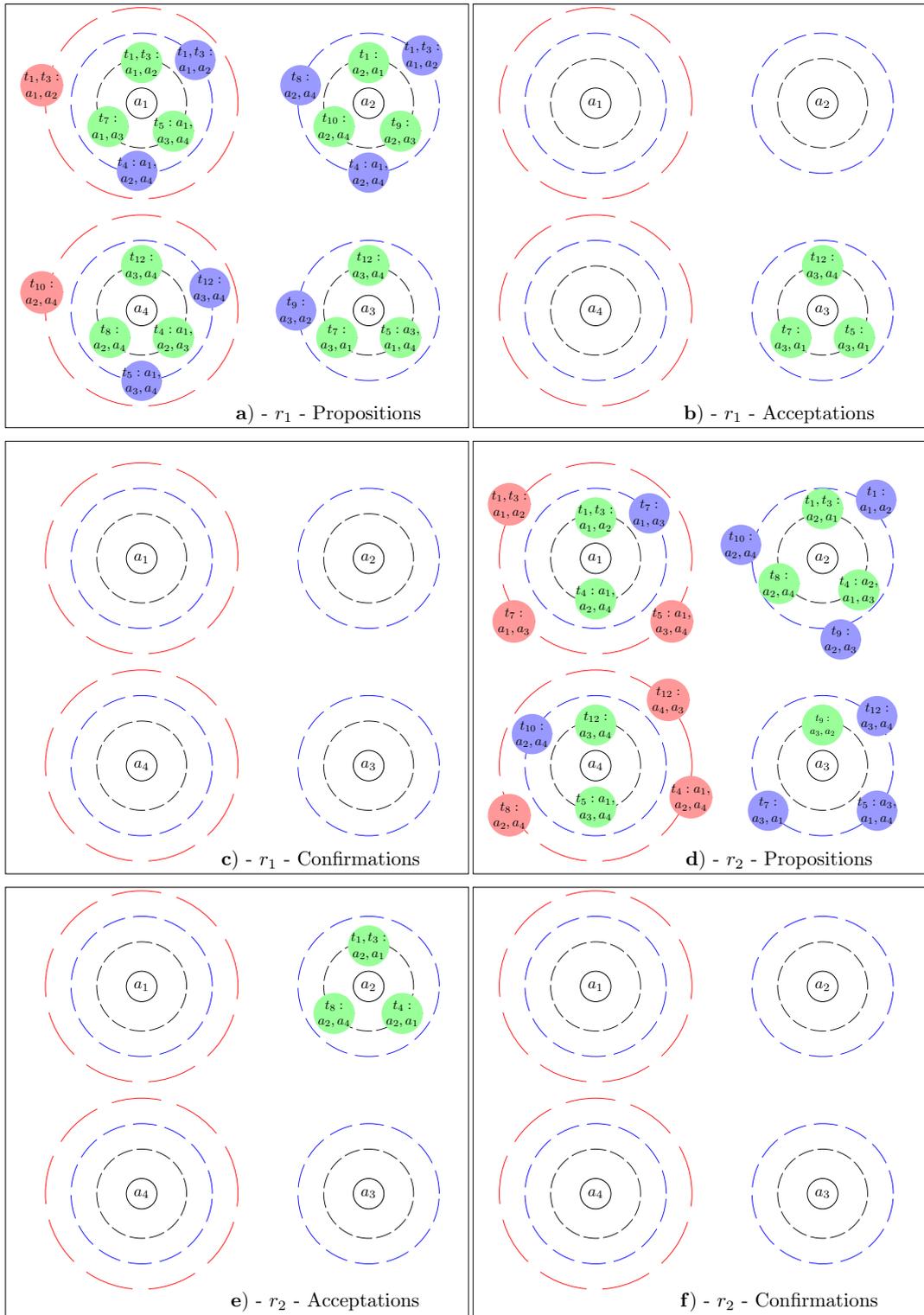


Fig. 4.4 – Illustration graphique du processus des négociations.

		α_1 (Tom)			α_2 (Lisa)	
		$\alpha_{1,1}$	$\alpha_{1,2}$	$\alpha_{1,3}$	$\alpha_{2,1}$	$\alpha_{2,2}$
r_1	u^ϵ	15.0	26.67	13.34	45.84	47.5
	Ordre	2	1	3	2	1
r_2	$(\epsilon, \bar{\epsilon})$	(0.08,20.0)	(0.25,20.0)	(0.22,20.0)	(0.35,20.0)	(0.07,20.0)
	$(\omega, \bar{\omega})$	(1.0,2.0)	(0.66,2.0)	(0.6,2.0)	(0.83,2.0)	(0.5,2.0)
	$(\gamma, \bar{\gamma})$	(0.0,5.0)	(1.0,5.0)	(0.0,5.0)	(0.6,5.0)	(0.4,5.0)
	$(\rho, \bar{\rho})$	(0.0,10.33)	(1.0,10.33)	(0.0,10.33)	(0.0,10.33)	(1.0,10.33)
	$(\mu, \bar{\mu})$	(0.0,1.0)	(0.0,1.0)	(0.0,1.0)	(0.0,1.0)	(0.0,1.0)
	\mathcal{V}	4.28	-0.58	4.46	7.26	-4.78
	Ordre	2	3	1	1	2
r_3	$(\epsilon, \bar{\epsilon})$	(0.08,20.0)	(0.25,20.0)	(0.19,20.0)	(0.20,20.0)	(0.07,20.0)
	$(\omega, \bar{\omega})$	(1.0,2.0)	(1.0,2.0)	(1.0,2.0)	(1.0,2.0)	(0.75,2.0)
	$(\gamma, \bar{\gamma})$	(0.0,5.0)	(1.0,5.0)	(0.0,5.05)	(0.62,5.0)	(0.37,5.0)
	$(\rho, \bar{\rho})$	(0.0,7.0)	(0.5,11.21)	(0.5,7.0)	(0.5,7.0)	(0.5,7.0)
	$(\mu, \bar{\mu})$	(1.03,17.77)	(0.0,1.0)	(-1.29,10.0)	(1.61,18.57)	(0.21,11.42)
	\mathcal{V}	23.06	6.81	2.22	117.53	3.79
	Ordre	1	2	3	1	2
r_4	$(\epsilon, \bar{\epsilon})$	(0.08,20.0)	(0.25,20.0)	(0.19,20.0)	(0.19,20.0)	
	$(\omega, \bar{\omega})$	(1.0,2.0)	(1.0,2.0)	(1.0,2.0)	(1.5,2.0)	
	$(\gamma, \bar{\gamma})$	(0.0,5.0)	(1.0,5.0)	(0.0,5.05)	(1.0,5.00)	
	$(\rho, \bar{\rho})$	(0.33,11.89)	(0.33,7.0)	(0.33,7.0)	(0.66,7.00)	
	$(\mu, \bar{\mu})$	(0.0,1.0)	(1.29,17.77)	(-1.29,10.0)	(-5.12,1.0)	
	\mathcal{V}	3.00	48.96	2.95	6.98	
	Ordre	2	1	3	1	-
r_5	$(\epsilon, \bar{\epsilon})$		(0.06,20.0)			
	$(\omega, \bar{\omega})$		(6.0,2.0)			
	$(\gamma, \bar{\gamma})$		(1.0,5.0)			
	$(\rho, \bar{\rho})$		(0.5,12.44)			
	$(\mu, \bar{\mu})$		(0.0,1.0)			
	\mathcal{V}		67.68			
	Ordre	-	1	-	1	-

 Tab. 4.5 – Les valeurs de \mathcal{P} , $\bar{\mathcal{P}}$ et \mathcal{V} suivant r pour α_1 et α_2 .

		α_3 (Mark)		α_4 (Alice)		
		$\alpha_{3,1}$	$\alpha_{3,2}$	$\alpha_{4,1}$	$\alpha_{4,2}$	$\alpha_{4,3}$
r_1	u^e	22.5	51.67	21.67	25.84	15.0
	Ordre	2	1	2	1	3
r_2	$(\varepsilon, \bar{\varepsilon})$	(0.16,20.0)	(0.23,20.0)	(0.26,20.0)	(0.07,20.0)	(0.12,20.0)
	$(\omega, \bar{\omega})$	(1.0,2.0)	(0.75,2.0)	(1.0,2.0)	(0.25,2.0)	(1.0,2.0)
	$(\gamma, \bar{\gamma})$	(0.375,5.0)	(0.625,5.0)	(0.53,5.0)	(0.23,5.0)	(0.23,5.0)
	$(\rho, \bar{\rho})$	(0.0,10.33)	(1.0,10.33)	(0.0,10.33)	(1.0,10.33)	(0.0,10.33)
	$(\mu, \bar{\mu})$	(0.0,1.0)	(0.0,1.0)	(0.0,1.0)	(0.0,1.0)	(0.0,1.0)
	\mathcal{V}	5.44	-2.92	6.57	-5.46	4.88
	Ordre	1	2	1	3	2
r_3	$(\varepsilon, \bar{\varepsilon})$	(0.0,20.0)	(0.23,20.0)	(0.0,20.0)	(0.07,20.0)	(0.12,20.0)
	$(\omega, \bar{\omega})$	(1.0,2.0)	(1.0,2.0)	(1.0,2.0)	(1.0,2.0)	(1.0,2.0)
	$(\gamma, \bar{\gamma})$	(0.37,5.0)	(0.62,5.0)	(0.35,5.00)	(0.5,5.0)	(0.14,5.0)
	$(\rho, \bar{\rho})$	(0.5,7.0)	(0.5,7.0)	(0.5,7.00)	(0.5,7.0)	(0.0,7.0)
	$(\mu, \bar{\mu})$	(-1.41,10.0)	(-1.76,10.0)	(-4.07,1.0)	(2.40,24.44)	(0.88,18.88)
	\mathcal{V}	2.22	4.10	3.11	2200.26	17.33
	Ordre	2	1	3	1	2
r_4	$(\varepsilon, \bar{\varepsilon})$		(0.30,20.0)		(0.03,20.0)	
	$(\omega, \bar{\omega})$		(1.33,2.0)		(4.0,2.0)	
	$(\gamma, \bar{\gamma})$		(1.0,5.0)		(1.0,5.0)	
	$(\rho, \bar{\rho})$		(0.66,7.0)		(0.66,7.0)	
	$(\mu, \bar{\mu})$		(1.54,18.75)		(1.48,18.88)	
	\mathcal{V}		99.48		96.18	
	Ordre	-	1	-	1	-
r_5	$(\varepsilon, \bar{\varepsilon})$					
	$(\omega, \bar{\omega})$					
	$(\gamma, \bar{\gamma})$					
	$(\rho, \bar{\rho})$					
	$(\mu, \bar{\mu})$					
	\mathcal{V}					
	Ordre	-	1	-	1	-

 Tab. 4.6 – Les valeurs de \mathcal{P} , $\bar{\mathcal{P}}$ et \mathcal{V} suivant r pour α_3 et α_4 .

À la deuxième étape de r_2 (*cf.* Fig. 4.4-e), la condition $u_2^\epsilon(\alpha_{2,1}) > u_2^\delta$ est vérifiée seulement pour l'agent a_2 pour son alternative sélectionnée $\alpha_{2,1}$ dans son état intermédiaire. De ce fait, a_2 envoie des acceptations à a_1 à propos de $c_{1,2}$ et $c_{1,4}$, et à a_4 à propos de $c_{4,4}$. Les agents a_1, a_3, a_4 quant à eux, attendent le prochain cycle parce que la condition $u^\epsilon > u^\delta$ n'est plus satisfaite pour eux à cette étape. À la troisième étape de r_2 (*cf.* Fig. 4.4-f), comme la condition $u_2^\epsilon(\alpha_{2,1}) > u_2^\delta$ n'est plus satisfaite pour a_2 (parce qu'il n'a reçu aucune acceptation de a_1 et de a_4), il n'a pas envoyé de confirmation.

Le reste des cycles de négociations est illustré par les tableaux Tab. 4.5 et Tab. 4.6.

4.4.4 La solution

Comme on peut le constater, les agents ont alterné les propositions de coalitions et ont convergé vers une solution globale à la fin des négociations au cycle r_4 . La figure Fig. 4.5 montre les structures de coalitions retenues où tous les agents ont réussi à former au moins deux coalitions ; ce qui leur permet d'atteindre leurs buts avec des coûts réduits.

À la fin des négociations, une structure de coalitions sélectionnée \hat{c}_i peut être associée à différentes alternatives possibles¹² $\alpha_{i,k} \in \mathcal{Q}_i^{\hat{c}_i}$ où $\mathcal{Q}_i^{\hat{c}_i} = \{\alpha_{i,k} | \Gamma(\hat{c}_i) \subseteq \mathcal{T}_i^{\alpha_{i,k}}\}$ qui vérifie $\Phi_i(\alpha_{i,k}) = \min_{\alpha_{i,k} \in \mathcal{Q}_i^{\hat{c}_i}} (\Phi_i(\alpha_{i,k}))$. Ainsi, la structure de coalitions sélectionnée pour Tom est $\hat{c}_1 = \{\langle\{a_1, a_2\}, \{t_1, t_3\}\rangle, \langle\{a_1, a_3\}, \{t_5, t_7\}\rangle\}$ qui est associée à l'alternative $\alpha_{1,2} = \langle\{t_1, t_3, t_5, t_7\}, \{t_1 \prec t_3, t_3 \prec t_5, t_5 \prec t_7\}\rangle$.

Dans cette solution finale, dite globale, chaque individu a sa propre structure de coalitions. En effet, Tom a des coalitions pour toutes les tâches de l'alternative $\alpha_{1,2} : \Gamma(\hat{c}_1) = \Gamma(\alpha_{1,2}^{cs})$. Dans d'autres cas, l'ensemble des tâches dans la structure de coalitions sélectionnée peut ne pas couvrir toutes les tâches nécessaires pour une alternative comme dans le cas d'Alice, $\hat{c}_4 = \{\langle\{a_4, a_3\}, \{t_{12}\}\rangle, \langle\{a_4, a_2\}, \{t_4, t_8\}\rangle\}$.

12. Cette situation se produit lorsqu'un agent n'a obtenu l'engagement que pour quelques coalitions parmi ses propositions formulées. Cela peut conduire à des situations où la structure de coalitions retenue comme solution (qui est dans un *état intermédiaire*) peut être associée à d'autres alternatives. Ceci est possible parce que les alternatives ont des tâches en commun.

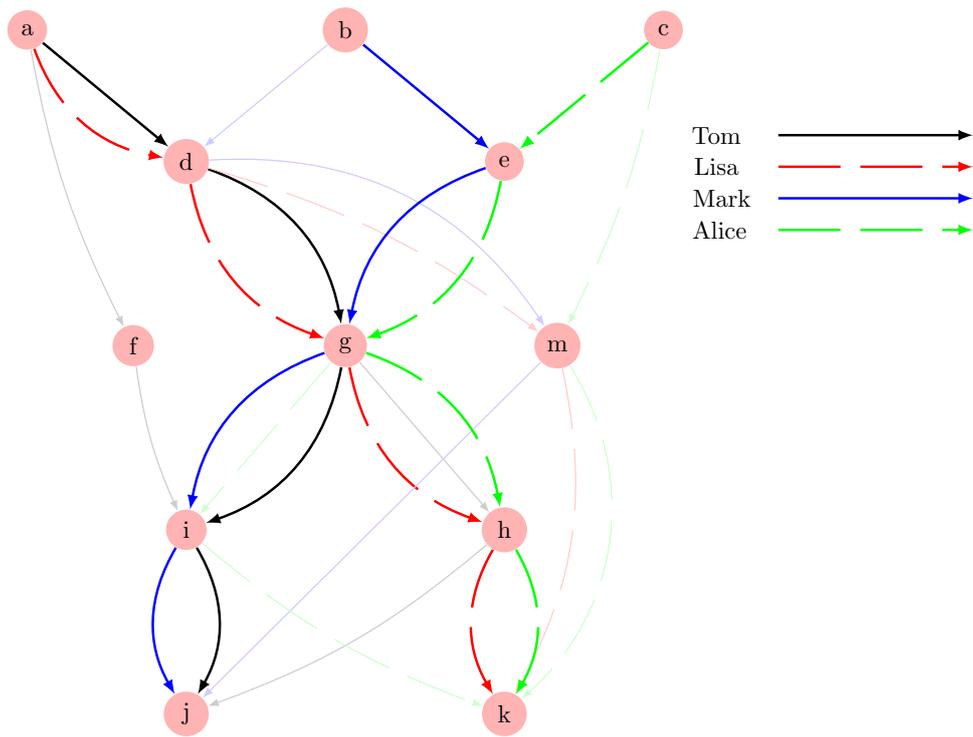


Fig. 4.5 – Les structures de coalitions retenues.

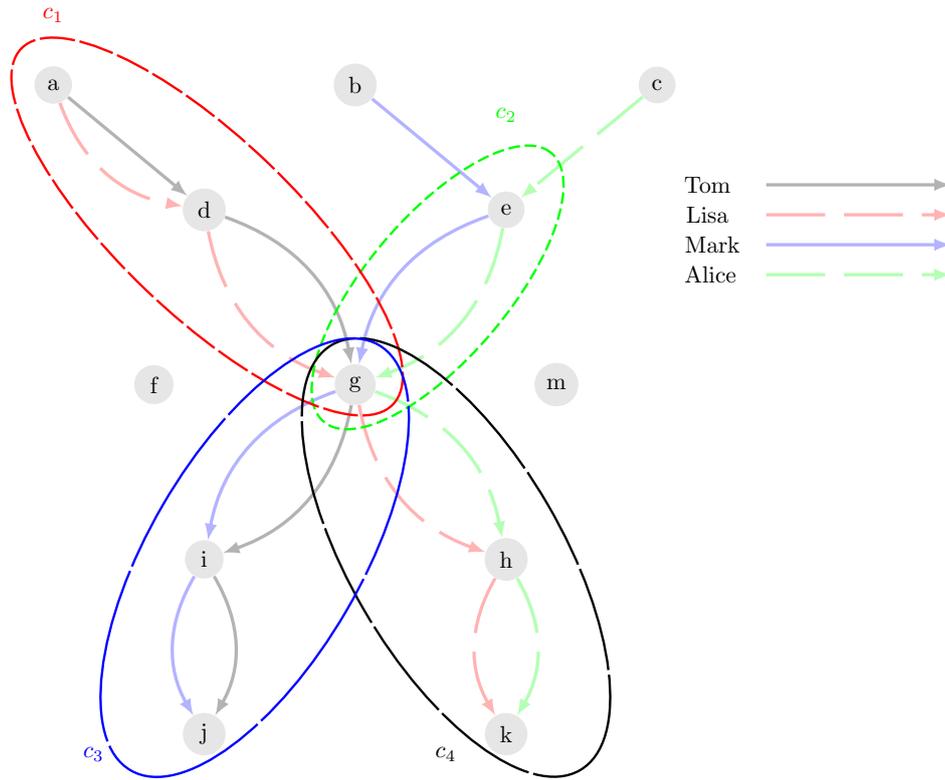


Fig. 4.6 – Les dépendances entre les coalitions.

Dans la pratique, plusieurs alternatives peuvent être possibles pour une structure de coalitions sélectionnée $\hat{\mathcal{C}}_i$.

Agents	Alternative sélectionnée	ϕ^-	ϕ	u
Tom (\mathbf{a}_1)	$\alpha_{1,2}$	130	110	20
Lisa (\mathbf{a}_2)	$\alpha_{2,1}$	140	97.5	42.5
Mark (\mathbf{a}_3)	$\alpha_{3,2}$	140	95	45
Alice (\mathbf{a}_4)	$\alpha_{4,2}$	125	102.5	22.5

Tab. 4.7 – L'utilité finale des agents.

Le tableau Tab. 4.7, illustre l'utilité finale obtenue par chacun des agents.

En résumé, comme on peut le voir sur la figure Fig. 4.6, la coalition c_3 dépend des deux coalitions c_1 et c_2 et la coalition c_4 dépend aussi des coalitions c_1 et c_2 .

4.5 Conclusion du chapitre

Nous avons proposé dans ce chapitre un algorithme de formation de coalitions (CFDEA) avec des externalités dynamiques dans sa version distribuée. Nous avons exposé un protocole séquentiel de négociations multi-agents qui s'exécute en cycles avec trois différentes étapes par cycles. Nous avons détaillé également les règles de négociations qui permettent aux agents de déterminer l'action à entreprendre à chaque étape ainsi qu'une stratégie de prise de décision basée sur une fonction d'ordonnancement qui tient compte de plusieurs paramètres caractéristiques, différenciant chaque étape de négociations par rapport aux différentes propositions formulées par un agent et celles reçues des autres agents. Ainsi, notre stratégie est basée sur l'utilisation de l'historique des négociations dans le but de guider la formation des coalitions et de faire converger les propositions des agents sans partager leurs informations privées. La prise de décision est sur trois niveaux : (i) un niveau formulation des propositions où l'agent peut décider de formuler des propositions ou de laisser passer le tour sans intervenir (ii) un niveau acceptation des propositions où l'agent peut accepter des propositions d'autres agents qui lui semble bonnes à prendre, mais cela ne constitue pas en soit un engagement. (iii) et enfin un niveau confirmation où l'agent confirme sa volonté de s'engager dans la coalition. Ce n'est qu'au bout d'une confirmation mutuelle entre plusieurs agents que ces derniers sont considérés comme engagés dans la coalition en question.

Dans le chapitre suivant, nous présentons une évaluation expérimentale et montrons l'efficacité de notre algorithme dans différentes situations selon différents critères comme l'efficacité, la déviation et le taux de succès.

CHAPITRE 5

ÉVALUATION EXPÉRIMENTALE DE L'ALGORITHME CFDEA

L'objectif de l'évaluation de notre algorithme de formation de coalitions est triple. Premièrement, il s'agit de montrer le comportement adopté par les agents durant les négociations et la façon dont ils tiennent compte des propositions des uns et des autres afin d'aboutir à une solution. Deuxièmement, il s'agit d'étudier à l'issue des négociations les performances du système en termes de gains obtenus par les agents, de nombre d'agents ayant formé au moins une coalition et de déviation enregistrée sur les valeurs de l'utilité individuelle de chacun. Troisièmement, il s'agit d'étudier l'impact que peut avoir le nombre d'agents dans le système ainsi que le nombre d'alternatives dont dispose chaque agent sur les performances du système.

5.1 Introduction

Lors de l'évaluation expérimentale des performances de CFDEA, nous avons implémenté notre algorithme et développé un système multi-agents en utilisant le langage JAVA et la plateforme multi-agents JADE. Cette dernière est une plateforme permettant de développer des agents capables de communiquer par en-

voi de messages. Les agents évoluent de manière indépendante. Chacun possède un identifiant unique (AID) au sein de la plateforme et peut obtenir la liste de tous les AID des autres. JADE offre alors la possibilité aux agents de communiquer en s’envoyant des messages à partir des AID. De plus, elle propose aux concepteurs de développer leurs agents suivant une architecture spécifique.

En effet, un agent JADE peut être vu comme une liste de comportements (classe Behaviour) et correspond à un thread JAVA. Chaque comportement est une routine JAVA exécutable. L’ensemble des comportements forme une file d’attente et ceux-ci sont exécutés successivement par un ordonnanceur. Un comportement est défini par deux méthodes : la méthode `action()` qui est exécutée en premier et qui correspond au code du comportement défini pour l’agent et la méthode `done()` qui est exécutée après la méthode `action()` et vérifie si le comportement doit être remis dans la file d’attente de l’ordonnanceur ou non. Un comportement peut donc être exécuté plusieurs fois, mais un agent n’exécute qu’un seul comportement à la fois.

Afin de rendre l’évaluation aussi réaliste que possible, nous avons développé un *Générateur aléatoire de plans* (GAP) qui produit des situations où les individus veulent se déplacer d’une ville de départ à une ville de destination selon plusieurs alternatives possibles. Dans la suite de ce chapitre, nous décrivons d’abord l’approche d’évaluation expérimentale que nous adoptons, puis nous présentons et discutons les résultats expérimentaux.

5.2 Approche expérimentale

Les objectifs que nous nous sommes fixés pour l’évaluation expérimentale sont triples : Premièrement, nous voulons démontrer la pertinence de CFDEP pour les systèmes multi-agents et montrer la faisabilité et l’efficacité de CFDEA. Notre but ici est d’étudier le comportement des agents pour former leurs coalitions (trouver une structure de coalitions pour chacun d’eux). L’évaluation comprend deux études de cas. Dans le premier cas, nous visons à observer le comportement des agents et le nombre de cycles nécessaires pour arriver à une solution. Dans le second cas, nous visons à examiner comment l’heuristique *IS&C* impacte ce comportement

pendant les négociations.

Deuxièmement, nous voulons démontrer l'efficacité de l'heuristique *IS&C* pour parvenir à une solution $cs \in (\mathcal{S}^+ \cup \mathcal{S}^*)$. L'idée est de montrer comment cette heuristique affecte le comportement des agents en fonction des données collectées lors des cycles précédents des négociations et comment les agents réagissent lorsqu'une convergence dans leurs propositions est identifiée au cours d'un cycle donné.

Enfin, nous voulons évaluer la performance du système multi-agents dans son ensemble. Nous voulons démontrer l'efficacité du CFDEA par rapport à l'Algorithme de la Meilleure Alternative (*AMA*) que nous allons présenter dans la section suivante, et étudier la qualité de la solution obtenue en termes d'efficacité, de taux de réussite et de taux de déviation.

Pour atteindre ces trois objectifs, nous avons mené plusieurs expérimentations suivant deux cas :

1. Dans le premier cas, nous avons varié le nombre d'agents impliqués dans les négociations et le nombre de leurs alternatives respectives.
2. Dans le deuxième cas, nous avons mené des expérimentations sur deux ensembles d'agents pour examiner la mise à l'échelle du CFDEA et son effet sur la performance. Pour cela, nous introduisons trois mesures de performance : l'efficacité, le taux de réussite et la déviation (l'écart type). Leurs définitions et leurs utilisations sont décrites ci-après.

5.2.1 Paramètres

Dans cette section, nous présentons la structure de nos agents et introduisons de nouveaux paramètres nécessaires à nos expérimentations. Toutefois, nous précisons que ces paramètres ne font pas partie du problème lui-même ni des algorithmes de résolution. Nous les introduisons pour illustrer un scénario réaliste basé sur des notions comme le coût et le gain.

Nous avons doté les agents d'un comportement permettant l'envoi de messages à un ou plusieurs autres agents de la plateforme. Les messages contiennent les informations relatives aux coalitions proposées ainsi que les primitives du protocole MNP, à savoir : *Proposal*, *Accept* et *Confirm*. JADE offre par ailleurs la possibilité

de réceptionner de manière automatique les messages dans une liste d'attente. Nous avons développé un comportement de lecture de la file d'attente qui se répète de manière cyclique pendant toute la simulation. Les agents récupèrent les messages par ordre de réception dans chaque étape des négociations. Ils sont définis de telle manière à ce qu'ils lisent les messages séquentiellement et les traitent par groupe. C'est-à-dire, ils lisent durant l'étape courante tous les messages reçus de l'étape précédente, les traitent puis répondent durant la même étape. Les réponses seront lues à la prochaine étape des négociations. Ainsi, à chaque étape, il y a deux phases : une première phase qui consiste à lire les messages reçus de la précédente étape et une deuxième phase qui consiste à envoyer les messages aux autres agents qui seront lus à la prochaine étape.

Rappelons que les agents prennent leurs décisions sur la base de leur utilité estimée qui est elle-même basée sur les coûts des alternatives. Le coût d'une alternative est calculé comme suit :

$$\phi_i(\alpha_{i,k}) = \sum_{c_{i,q} \in \alpha_{i,k}^{cs_{i,s}}} \sum_{t \in \mathcal{T}_{c_{i,q}}} \phi_i(t) + \sum_{t \in (\mathcal{T}_i^{\alpha_i, k} - \Gamma(\alpha_{i,k}^{cs_{i,s}}))} \phi_i(t) \quad (5.1)$$

et nous maintenons l'hypothèse qu'il existe un coût de réservation (ϕ_i^{res}),

$$\phi_i^{\text{res}} = \left(\frac{\phi_i^+ + \phi_i^-}{2} \right) \times \phi_i^{\text{limit}} \quad (5.2)$$

qui est obtenu sur la base d'un $\phi_{i=1, \dots, N}^{\text{limit}} \in [0.3, 1]$. La valeur de ϕ^{limit} fera l'objet de plusieurs expérimentations sur l'intervalle $[0.3, 1]$ pour étudier l'effet de limiter les coûts possibles sur la qualité de la solution. Par ailleurs, le coût minimum individuel (ϕ_i^-) est considéré comme un coût de référence utilisé pour calculer l'utilité estimée (u_i^ϵ) et l'utilité de référence (u_i^δ). Une utilité estimée d'une alternative est définie comme suit :

$$u_i^\epsilon(\alpha_{i,k}) = \phi_i^- - \left(\sum_{c_{i,q} \in \alpha_{i,k}^{cs_{i,s}}} \sum_{t \in \mathcal{T}_{c_{i,q}}} \left(\frac{\phi^{\text{pub}}(t)}{|\mathcal{A}_{c_{i,q}}|} + \phi_i^{\text{priv}}(t) \right) + \sum_{t \in (\mathcal{T}_i^{\alpha_i, k} - \Gamma(\alpha_{i,k}^{cs_{i,s}}))} \phi_i(t) \right) \quad (5.3)$$

Dans nos expérimentations, une structure de coalitions est acceptable pour un agent a_i si elle vérifie la condition $u_i^\epsilon(\alpha_{i,k}) > u_i^\delta$. En faisant varier la valeur $\phi_{i=1, \dots, N}^{\text{limit}}$

dans l'intervalle $]0.3, 1]$, nous pouvons contrôler la complexité de la recherche d'une solution. Des valeurs inférieures à ϕ_i^{limit} amèneront les agents à rechercher une utilité supérieure, car ils auront des coûts limités autorisés à dépenser pour les coalitions. Notons que, pour toutes les expérimentations, les résultats sont une moyenne de 30 exécutions différentes pour 10 valeurs différentes de ϕ_i^{res} selon une variation graduelle de $0.3 < \phi_i^{\text{limit}} < 1$.

5.2.2 Générateur aléatoire de plans

Nous avons développé un *Générateur aléatoire de plans* (GAP) qui génère des plans de manière aléatoire (par exemple, les alternatives, les points de départ, les objectifs et les coûts privés et publics des tâches). Les détails du GAP sont illustrés par le tableau Tab. 5.1.

Nombre de tâches possibles	180
Nombre de plans possibles	$\approx 2 * 10^9$
Nombre d'alternatives par plans	$3 \leq n \leq 25$
Nombre de tâches par alternative	$3 \leq n \leq 7$
Nombre de positions de départs possibles	8
Nombre de positions d'arrivées possibles (buts)	8
Le coût public des tâches	$60 \leq \phi^{\text{pub}} \leq 80$
Le coût privé des tâches	$30 \leq \phi^{\text{priv}} \leq 100$

Tab. 5.1 – Les paramètres du générateur aléatoire de plans.

5.2.3 Algorithme de la meilleure alternative

Comme il n'y a pas d'approche dans la littérature comparable à notre algorithme, nous faisons une comparaison de nos résultats avec ceux de l'*Algorithme de la meilleure alternative* (BAA – *Best Alternative Approach* –), qui est une approche intuitive pour la résolution de CFDEP. Dans BAA, chaque agent ordonne ses alternatives puis sélectionne sa préférée α_i^* et propose la structure de coalitions

cs (α^{cs}) qui lui est associée aux agents impliqués $\mathbf{a} \in \Lambda(sc)$. Ensuite, chaque agent $\mathbf{a} \in \Lambda(cs)$ ayant reçu une proposition concernant une coalition $c \in cs$ (c est aussi dans son α^{cs} préférée) acceptera et confirmera cette coalition à son émetteur. Ici, il n'y a pas de négociations entre les agents. Chacun d'eux ne considère que son alternative préférée issue du premier ordonnancement sur la base de son utilité estimée, puis accepte de réaliser conjointement les tâches communes entre son alternative préférée et les structures de coalitions préférées des autres agents (chacun reçoit des propositions des autres agents uniquement depuis leurs structures de coalitions préférées). En d'autres termes, dans BAA, chaque agent choisit sa structure de coalitions préférée et l'adopte comme solution. Ainsi, chaque agent n'aura donc que les coalitions qui figurent également dans les structures de coalitions préférées des autres agents.

5.2.4 Mesure de la performance

Dans nos expérimentations, nous utilisons trois mesures pour évaluer la performance du système multi-agents, à savoir : *Efficacité*, *Taux de réussite* et *Déviaton*. Leurs valeurs sont des moyennes sur plusieurs expérimentations.

Efficacité : L'efficacité φ est la moyenne des gains réalisés par tous les agents.

$$\varphi = \frac{\frac{1}{n} \sum_{l=1}^n \left(\frac{\sum_{\mathbf{a}_i \in \mathcal{A}} u_i(\alpha_{i,k})}{|\mathcal{A}|} \right)}{N_{\text{exp}}} \quad (5.4)$$

Où n est le nombre d'exécutions pour une expérimentation pour une valeur de ϕ_i^{limit} et N_{exp} est le total des expérimentations. L'utilité de l'alternative sélectionnée $\alpha_{i,k}$ pour un agent est :

$$u_i(\alpha_{i,k}) = \frac{\phi_i^- - \phi_i^f(\alpha_{i,k})}{\phi_i^-} \quad (5.5)$$

Où $\phi_i^f(\alpha_{i,k})$ est le coût réel (final) de $\alpha_{i,k}$ suivant les coalitions formées.

Taux de réussite : Le taux de réussite R_s est défini comme le nombre d'agents a_i qui ont formé des coalitions dans $\hat{c}s_i \in (\mathcal{S}_i^+ \cup \mathcal{S}_i^*)$. Formellement :

$$R_s = \frac{\sum_1^n \frac{N_e}{|\mathcal{A}|}}{n} \quad (5.6)$$

Où n est le nombre d'exécutions dans une expérimentation pour une valeur de ϕ_i^{limit} et N_e est le nombre d'agents qui ont formé au moins une coalition à l'issue de chaque exécution.

Déviatiion : La déviatiion (ou écart type) mesure les différences entre les gains individuels en efficacité des membres d'un groupe. Elle est basée sur la définition statistique de l'écart-type. Dans notre contexte, la déviatiion mesure l'équité dans la répartition des gains. Plus l'écart augmente, plus l'équité diminue.

$$\text{Déviatiion} = \sqrt{\frac{1}{|\mathcal{A}| - 1} \sum_{i=1}^{|\mathcal{A}|} (u_i - \bar{u})^2} \quad (5.7)$$

Où \bar{u} est la valeur moyenne de $u_{i=1, \dots, |\mathcal{A}|}$.

5.3 Étude de la dynamique de formation de coalitions avec CFDEA

Dans cette section, nous analysons deux aspects essentiels de la dynamique de formation de coalitions avec l'algorithme CFDEA. Premièrement, nous examinons le nombre de cycles des négociations nécessaires pour aboutir à une solution. Deuxièmement, nous étudions l'effet de l'heuristique $\mathcal{IS\&C}$ sur le comportement des agents pour converger et aboutir à des solutions (structures de coalitions).

5.3.1 Cycles des négociations

Avant toute chose, nous étudions la relation entre ϕ^{limit} et le nombre de cycles nécessaires pour aboutir à une solution. Nous avons réalisé nos expérimentations

avec différentes valeurs de ϕ^{limit} qui varient de 0,52 à 1. Les résultats sont décrits dans la figure Fig. 5.1. L'expérience est réalisée avec trois agents, $\mathcal{A} = \{a_1, a_2, a_3\}$ possédant le même plan (mêmes tâches), mais des coûts de tâches différents. Chaque plan comporte 12 tâches formant six alternatives ($\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$). $\bar{\mathcal{P}}_{v_i=1,\dots,3} = \{\bar{\varepsilon} = 30, \bar{\omega} = 2, \bar{\gamma} = 5, \bar{\rho} = 7, \bar{\mu} = 15\}$. De toute évidence, la figure Fig. 5.1 montre que des valeurs ϕ^{limit} plus petites se traduisent par un plus grand nombre de cycles pour aboutir à une solution, et vice versa.

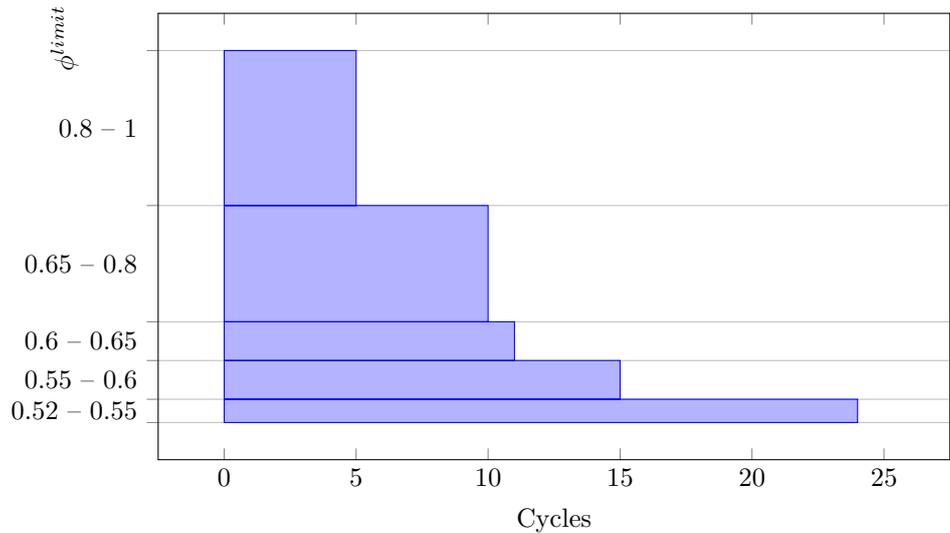


Fig. 5.1 – Le nombre de cycles suivant les valeurs de ϕ^{limit} .

Par exemple, pour $\phi^{\text{limit}} = 0.6$, une solution a été trouvée à $r = 11$ et pour $\phi^{\text{limit}} = 0.52$, une solution a été trouvée à $r = 24$.

5.3.2 Effet de l'heuristique $\mathcal{IS\&C}$ sur le comportement des agents

Dans cette section, nous nous concentrons sur le comportement individuel des agents utilisant l'heuristique $\mathcal{IS\&C}$ pour deux objectifs : (i) pour comprendre comment ils se comportent pour explorer leurs espaces de recherche (\mathcal{S}) ; et (ii) pour comprendre comment se fait la convergence vers une solution possible (si elle existe). Dans le premier cas, nous étudions la façon dont les agents explorent

\mathcal{S} . Comme décrit dans la section 4.3.4, l'heuristique $\mathcal{IS\&C}$ permet aux agents de rechercher et de converger de façon itérative vers une solution possible. Alors, nous avons réalisé des expérimentations avec trois agents qui ont les mêmes plans, mais avec des coûts privés différents ayant chacun 6 alternatives. L'utilisation de mêmes plans simplifie la compréhension du déroulement des négociations. En effet, avoir les mêmes alternatives rend le suivi des propositions échangées plus compréhensible en facilitant leur traçabilité. Par ailleurs, pour observer comment les négociations s'effectuent, d'une part, d'une étape à une autre, et d'autre part d'un cycle à un autre, nous avons configuré les agents de sorte à ne pas confirmer les solutions possibles pour faire durer les négociations. Cela permet une meilleure observation du processus de négociations.

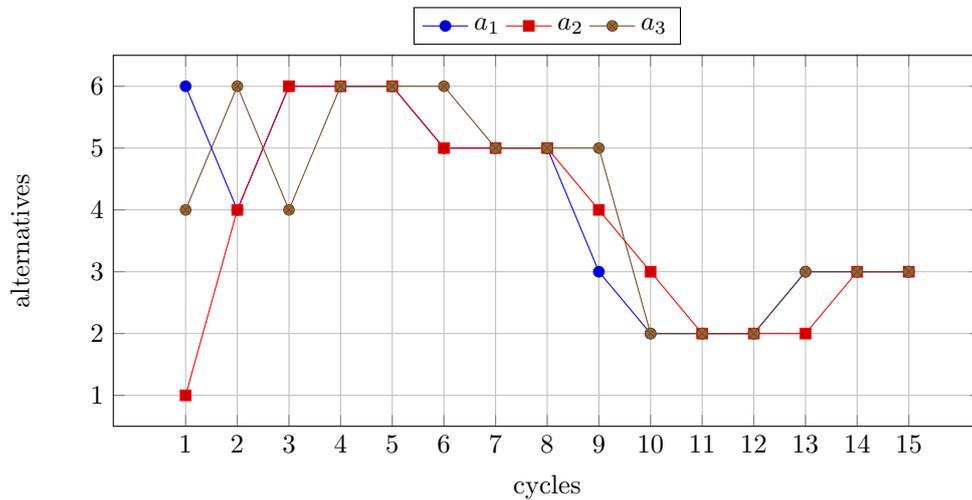


Fig. 5.2 – Le comportement des agents avec l'heuristique $\mathcal{IS\&C}$.

La figure Fig. 5.2 illustre le comportement des agents pendant les négociations et montre comment ils alternent entre les phases de recherche et de convergence ($\bar{\mathcal{P}}_{vi=1,\dots,3} = \{\bar{\varepsilon} = 7, \bar{\omega} = 2, \bar{\gamma} = 5, \bar{\rho} = 7, \bar{\mu} = 15\}$ et $\phi_{vi=1,\dots,3}^{\text{limit}} = 0.52$).

Comme nous pouvons l'observer, les négociations se déroulent selon un processus itératif composé de deux phases : les phases de recherche et les phases de convergence. Dans les cycles 1 et 2, les négociations étaient dans une phase de recherche au cours de laquelle les agents ont proposé différentes alternatives. Dans

le cycle $r = 3$, les agents ont identifié une convergence entre leurs propositions et ont mené une phase de convergence. Dans les cycles 4 et 5, les agents ont examiné si l'alternative α_6 peut être adoptée comme solution commune. Cependant, α_6 n'a pas été acceptée comme solution et les agents sont revenus à une phase de recherche dans $r = 6$ où α_2 a proposé une alternative α_5 . Ceci a conduit les agents à une phase de convergence (cycles 7 et 8) dans laquelle les autres agents ont suivi en proposant la même alternative α_5 . Encore une fois, α_5 n'a pas été retenue comme solution et les agents sont revenus à une phase de recherche (cycles 9 et 10), dans laquelle ils ont proposé d'autres alternatives. Dans $r = 10$, ils ont été dans une phase de convergence pour tester si l'alternative α_2 peut être acceptée. Cela n'a pas été le cas et les agents α_1 et α_3 ont changé leurs propositions à α_3 durant le cycle $r = 13$. À $r = 14$, α_2 a suivi α_1 et α_3 en proposant la même alternative, à savoir α_3 . Ceci les a menés à l'issue du cycle $r = 13$ à une autre phase de convergence¹.

À chaque cycle, la prise de décision par les agents est influencée par les paramètres $(\bar{\varepsilon}^\varepsilon, \bar{\omega}^\omega, \bar{\gamma}^\gamma, \bar{\mu}^\mu, \bar{\rho}^\rho)$. Nous avons donc mené des expérimentations pour observer l'effet de ces paramètres sur la fonction *désirabilité* $\mathcal{V}_{\alpha_i,k}$. Nous avons examiné les alternatives α_3 et α_6 suivant l'évolution de Δ à travers les cycles des négociations². Nous avons mesuré leur contribution relative par rapport à l'alternative α_6 (*cf.* Fig. 5.3-a) et l'alternative qui a été sélectionnée comme solution finale qui est α_3 (*cf.* Fig. 5.3-b). La figure Fig. 5.3 illustre les effets de ces paramètres sur la valeur de la fonction $\mathcal{V}_{\alpha_i,k}$ pour l'agent α_1 . Cela reflète les

1. Notons que dans cette expérience, les phases de convergence sont proches les unes des autres en raison de la valeur relativement élevée du paramètre de valorisation de la convergence $\bar{\mu}$. Les deux phases de négociations sont affectés par les valeurs des paramètres de valorisation. Des valeurs réduites de $\bar{\rho}$ font que les agents proposent plus fréquemment leurs alternatives préférées. Des valeurs élevées du paramètre $\bar{\gamma}$ font qu'ils proposent leurs alternatives pour lesquelles ils ont reçu plus de propositions de la part des autres agents. L'augmentation des valeurs $\bar{\varepsilon}$ incite les agents à se concentrer sur les alternatives qui maximisent leur utilité. Comme mentionné, $\bar{\mu}$ est associé à la convergence. $\bar{\omega}$ incite les agents à préférer les alternatives pour lesquelles ils ont reçu des propositions portant sur l'ensemble de leurs tâches et pas seulement sur quelques-unes d'entre elles.

2. Les valeurs de Δ sont normalisées, *i.e.* $\Delta \times 10^{-1}$, pour rendre les graphes plus compréhensibles.

changements observés dans les valeurs de Δ . Deux alternatives sont considérées pour l'agent α_1 . Au final, une solution a été trouvée à l'issue du cycle $r = 10$. $\bar{\mathcal{P}}_{\forall i=1,\dots,3} = \{\bar{\varepsilon} = 8, \bar{\omega} = 2, \bar{\gamma} = 20, \bar{\rho} = 15, \bar{\mu} = 5\}$, et $\phi_{\forall i=1,\dots,3}^{\text{limit}} = 0.52$. Notons que α_1 a initialement 6 alternatives, ordonnées comme suit : $\phi_1^\varepsilon(\alpha_{1,6}) < \phi_1^\varepsilon(\alpha_{1,3}) < \phi_1^\varepsilon(\alpha_{1,1}) < \phi_1^\varepsilon(\alpha_{1,2}) < \phi_1^\varepsilon(\alpha_{1,5}) < \phi_1^\varepsilon(\alpha_{1,4})$.

La figure Fig. 5.3-a montre que pour la plus préférée des alternatives $\alpha_{1,6}$ et pour $r < 6$, la valeur (absolue) de $\bar{\rho}^\rho$ était plus élevée dans des cycles où $\Delta = 0$ ($r = 1, r = 3, r = 5$) qui signifie que les négociations n'ont pas enregistrées des évolutions. En effet, le fait d'avoir une valeur $\bar{\rho}^\rho$ substantielle devrait permettre à l'agent de changer l'alternative à proposer dans les cycles à venir. Pour $r \in [6, 10]$, $\bar{\rho}^\rho$ était moins substantielle et α_1 a envoyé d'autres propositions que $\alpha_{1,6}$. Ce qui a fait que les autres paramètres ($\bar{\varepsilon}^\varepsilon, \bar{\omega}^\omega, \bar{\gamma}^\gamma$) ont produit un effet sur la valeur de $\mathcal{V}_{\alpha_{i,k}}$.

Dans l'ensemble, il y a eu deux phases des négociations concernant l'alternative α_6 : la première durant $r \in [1, 5]$ au cours de laquelle α_1 a enregistré une forte convergence dans les cycles $r = 2$ et $r = 4$ sans toutefois concrétiser une solution. La deuxième phase durant $r \in]5, 10]$ au cours de laquelle α_1 a enregistré majoritairement des divergences, qui s'expliquent par le fait qu'il explorait d'autres alternatives.

Dans la figure Fig. 5.3-b, la première observation est que pour $r < 9$, les coalitions possibles suivant α_3 n'ont pas été proposées aux autres agents (la valeur de $\bar{\rho}^\rho$ est restée inchangée) et ses tâches n'étaient pas désirées par les autres agents (la valeur de $\bar{\gamma}^\gamma$ est restée inchangée ainsi que celle de $\bar{\omega}^\omega$). La valeur la plus substantielle est le coût estimé $\bar{\varepsilon}^\varepsilon$. Pendant les négociations, pour $r \leq 3$, $\Delta = 0$ ce qui signifie qu'aucune évolution n'a été observée concernant α_3 . Pour $r \in [4, 7]$, α_1 a observé une divergence ($0 < \Delta < 1$); ce qui est le résultat de l'exploration d'autres alternatives par les autres agents. Pour $r \in [9, 10]$, α_1 a observé une forte convergence ($\Delta > 1$) avec les propositions des autres et, par conséquent, la valeur de $\bar{\mu}^\mu$ a été ajustée, ce qui a mené à une solution à l'issue du cycle $r = 10$.

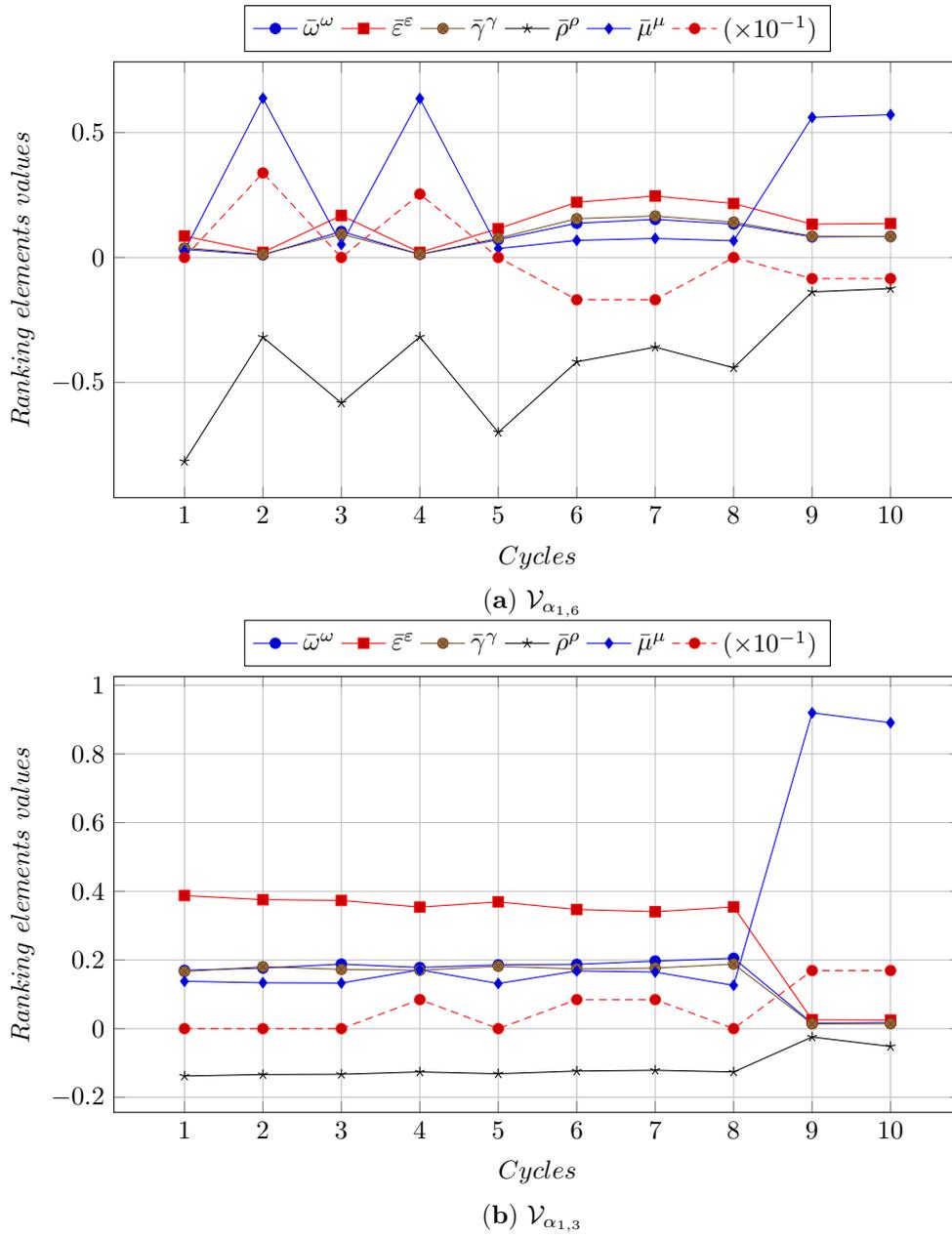


Fig. 5.3 – Le taux de contribution des paramètres $\bar{\varepsilon}^\varepsilon$, $\bar{\omega}^\omega$, $\bar{\gamma}^\gamma$, $\bar{\mu}^\mu$, $\bar{\rho}^\rho$ dans \mathcal{V}_α .

5.4 Performances de CFDEA

Rappelons que nous nous intéressons à la performance collective plutôt qu'à la performance individuelle. Elle est donc la moyenne de la performance individuelle de tous les agents, calculée sur la base des mesures pour toutes les expérimentations. Dans cette section, nous analysons la performance du système multi-agents en observant les caractéristiques des coalitions obtenues dans la solution finale. Pour atteindre cet objectif, nous avons réalisé deux séries d'expérimentations. Dans la première, nous étudions le comportement des agents avec l'algorithme CFDEA comparé à celui avec l'algorithme BAA. Dans cet ensemble d'expérimentations, nous faisons varier graduellement le nombre d'agents. Dans la deuxième série, nous étudions la performance du système avec différents agents et différentes alternatives pour chacun de ceux-ci.

Dans la première série d'expérimentations, nous mesurons l'efficacité dans deux cas, à savoir, pour $\varepsilon = 8$ et $\varepsilon = 50$. Ainsi, nous avons effectué des expérimentations avec 6 ensembles de 10 agents, 50 exécutions pour chaque ensemble ($\phi_{\forall i=1, \dots, 10}^{\text{limit}} = \{1, 0.95, 0.90, 0.85, 0.80, 0.75, 0.70, 0.65, 0.60, 0.55\}$). Le nombre maximum de cycles de négociations est fixé à $r = 30$. La figure Fig. 5.4 montre que des valeurs plus élevées du paramètre de valorisation des coûts impliquent une plus grande efficacité dans tous les ensembles.

Dans la deuxième partie, nous visons à évaluer le CFDEA en variant le nombre d'agents négociateurs et le nombre d'alternatives. Dans le premier cas, nous examinons 8 ensembles différents d'agents à partir de $|\mathcal{A}| = 3$ à $|\mathcal{A}| = 10$ où chacun a plusieurs alternatives. Pour chaque ensemble, nous avons effectué 3 sous-expériences différentes en variant le nombre d'alternatives pour chacun, allant de 6, 12 à 18 alternatives par agent. Enfin, pour chaque expérience, nous avons réalisé 50 séries en variant les tâches et leurs coûts. Nous avons fixé la limite de coût $\phi^{\text{limit}} = 0.70$ et la durée de la négociation à 25 tours. Les figures Fig. 5.5, Fig. 5.6 et Fig. 5.7 illustrent les résultats moyens des sous-expériences.

Nous constatons dans les figures Fig. 5.5, Fig. 5.6 et Fig. 5.7, l'efficacité de CFDEA est nettement supérieure à celle de BAA. Nous remarquons qu'elle est

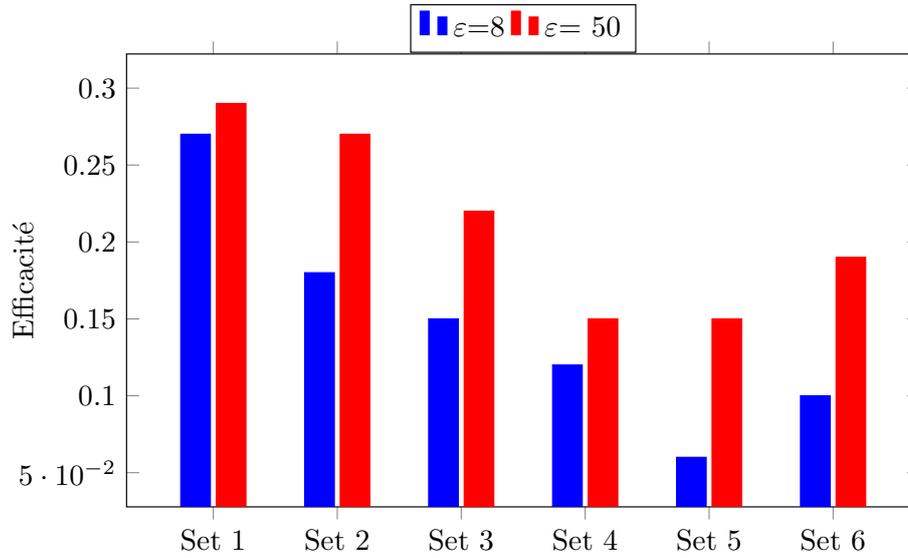


Fig. 5.4 – Efficacité du système suivant le paramètre de valorisation des coûts.

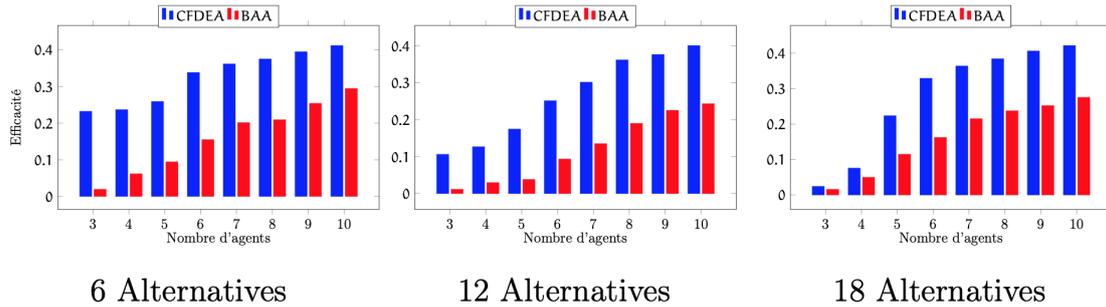


Fig. 5.5 – Efficacité du système suivant le nombre d’agents négociateurs et le nombre des alternatives par agent.

en corrélation avec le nombre d’agents. En ce qui concerne les taux de réussite, nous observons qu’à travers les expériences, les taux de réussite augmentent en augmentant le nombre d’agents dans les deux cas. Mais dans l’ensemble, les taux de réussite de BAA sont inférieurs à ceux de CFDEA. Les valeurs observées de la déviation montrent que dans le cas de BAA les valeurs de la déviation augmentent avec l’augmentation du nombre d’agents alors que dans le cas de CFDEA elles diminuent avec l’augmentation du nombre d’agents à partir de 5 agents.

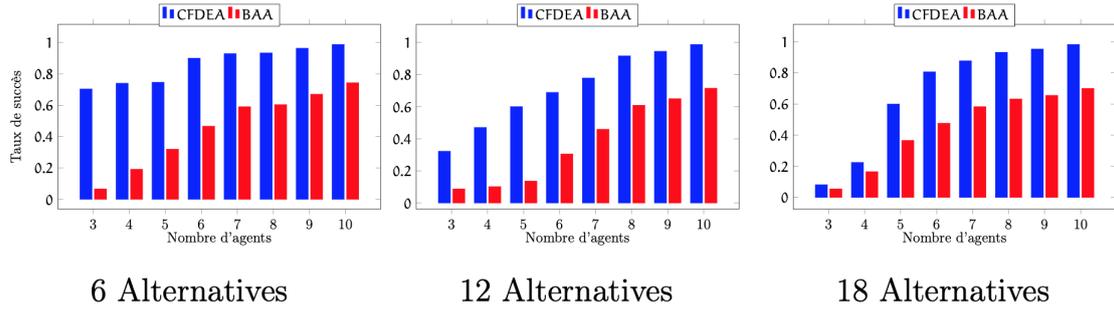


Fig. 5.6 – Le taux de succès suivant le nombre d’agents et le nombre de leurs alternatives.

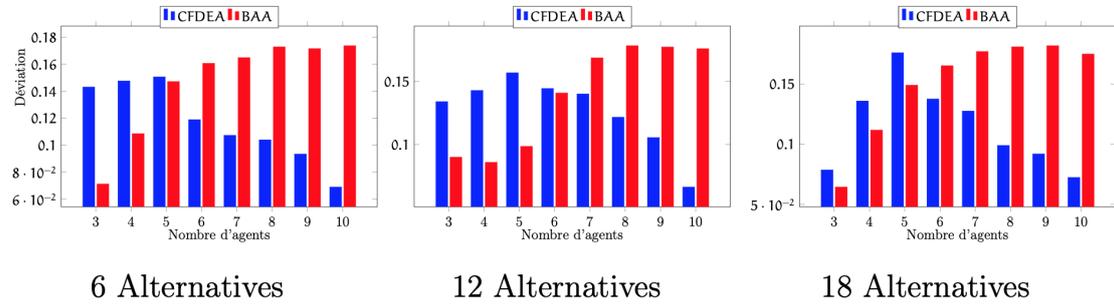
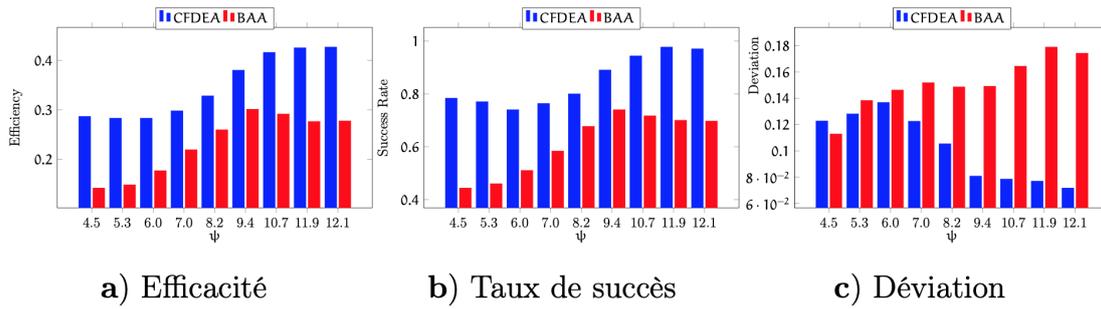
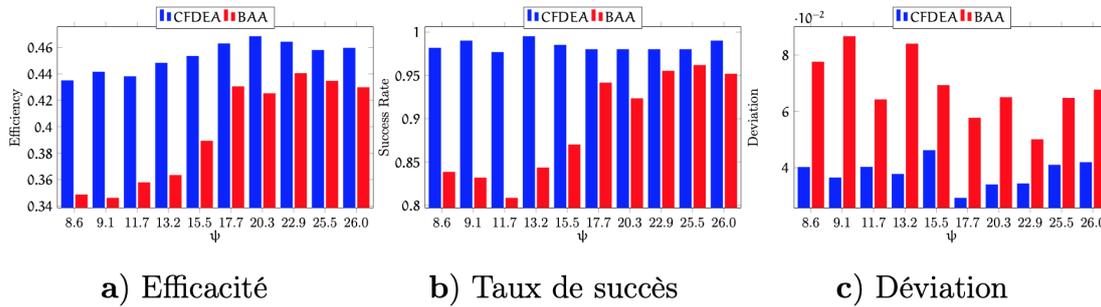


Fig. 5.7 – La déviation sur les valeurs d’efficacité des agents, suivant leur nombre et le nombre des alternatives par agent.

Dans le second cas, nous visons à étudier la performance du système multi-agents avec 10 puis 20 agents. Pour cela, nous avons effectué deux ensembles d’expériences avec 50 exécutions pour chacun en fixant la valeur limite de coût $\phi^{\text{limit}} = 0.70$. Pour distinguer les ensembles des agents, nous introduisons une nouvelle métrique, appelée *Cumulative Plans Overlap* et dénotée ψ , pour mesurer le degré de similarité entre les plans des agents.

$$\psi = \frac{\sum_{\alpha_i, k \in \mathcal{Q}_i} \sum_{\alpha_{i'}, k' \in \{\mathcal{Q}_A | i \neq i'\}} \left(\frac{|\mathcal{T}_i^{\alpha_i, k} \cap \mathcal{T}_{i'}^{\alpha_{i'}, k'}|}{|\mathcal{T}_i^{\alpha_i, k} \cup \mathcal{T}_{i'}^{\alpha_{i'}, k'}|} \right)}{|\tilde{\mathcal{T}}|} \quad (5.8)$$

Les figures Fig. 5.8 et Fig. 5.9 illustrent les résultats obtenus. La première observation est que dans les deux cas, CFDEA affiche une efficacité et un taux de réussite supérieurs à BAA. La deuxième observation est que, dans l’ensemble,


 Fig. 5.8 – Les performances du système avec 10 agents suivant les valeurs de ψ .

 Fig. 5.9 – Les performances du système avec 20 agents suivant les valeurs de ψ .

CFDEA a une meilleure performance dans le cas de 20 agents et le niveau de déviation pour CFDEA est moindre que pour BAA.

5.5 Conclusion du chapitre

Nous avons proposé dans ce chapitre une évaluation expérimentale des travaux présentés au chapitre 4. Nous avons développé deux configurations d'un système multi-agents sur la plateforme JADE. Une première configuration avec des agents participant à des négociations sur toutes leurs alternatives en utilisant l'algorithme CFDEA pour trouver une solution et une seconde configuration avec des agents qui ne considèrent que leur meilleure alternative (BAA). Les résultats montrent que les agents qui utilisent l'algorithme CFDEA obtiennent des performances meilleures que ceux qui utilisent l'algorithme BAA. L'heuristique $\mathcal{IS\&C}$ permet aux agents de parcourir leur espace de recherche d'une manière itérative tout en explorant les

possibilités de convergences lorsqu'un rapprochement entre leurs propositions est observé. Les stratégies de prise de décision et d'ordonnancement permettent aux agents de négocier sur les alternatives qu'ils considèrent comme plus avantageuses tout en concédant des concessions en fonction des intentions dérivées des autres agents.

Dans le chapitre suivant, nous présentons le problème CFDCE dans sa version centralisée ainsi que la solution que nous proposons pour le résoudre.

CHAPITRE 6

UNE APPROCHE CENTRALISÉE POUR LA RÉOLUTION DE CFDEP

La résolution du problème CFDEP dans sa version centralisée nécessite un algorithme qui serait capable de fonctionner dans un temps raisonnable en dépit du nombre élevé de structures de coalitions possibles. Il faudrait donc que l'algorithme, d'une part, tienne compte des dépendances existantes entre les tâches pour réduire l'espace de recherche et éviter les calculs redondants, par exemple, la génération multiple de toute coalition qui n'est pas réalisable à cause des contraintes. Et d'autre part, de bien combiner les coalitions réalisables pour générer efficacement des structures de coalitions valides. En effet, il peut être coûteux de passer par toutes les combinaisons possibles de coalitions et de vérifier s'il existe des conflits entre elles. Sur la base de ces exigences, nous présentons dans ce chapitre une formalisation du problème CFDEP dans le contexte centralisé, suivi de notre algorithme de résolution et de son évaluation expérimentale.

6.1 Introduction

Dans ce chapitre, nous présentons une version centralisée du problème CFDEP suivi de notre algorithme de résolution, qui permet de trouver la structure

de coalitions qui garantit la meilleure performance pour le système multi-agents en réduisant considérablement le nombre d'itérations nécessaires pour parcourir l'espace de recherche des agents.

Nous rappelons que nous considérons des agents compétitifs possédant chacun un ensemble d'alternatives (sous-ensembles) de tâches qui lui permet d'atteindre un but et une fonction d'utilité pour évaluer l'intérêt de former une coalition. Un agent doit donc former des coalitions pour ses sous-ensembles de tâches. Cependant, il existe des relations de dépendance au sein des tâches d'un même agent et chaque tâche a un coût de réalisation. Cela rend donc les agents particulièrement intéressés par la réalisation conjointe de leurs tâches, quand celles-ci le permettent, pour réduire le coût global nécessaire pour atteindre leurs buts. Dans le contexte centralisé, nous supposons que les agents acceptent de divulguer leur utilité ou fonction d'utilité à un agent central de confiance qui se charge de calculer le gain qu'ils peuvent obtenir de chaque structure de coalitions.

En tenant compte des particularités de CFDEP, nous proposons un algorithme de formation de coalitions suivant une démarche de **Parcours Sélectif** guidé par les **Dépendances (PSD)**, de l'espace de recherche. L'algorithme PSD est dit *anytime* où il peut retourner une solution à n'importe quel moment de son exécution et la qualité de sa solution s'améliore continuellement jusqu'à sa terminaison.

La suite de ce chapitre est organisée comme suit : premièrement, nous présentons les différents concepts relatifs à la version centralisée du problème CFDEP. Nous redéfinissons quelques concepts déjà introduits dans le chapitre 4 pour tenir compte des caractéristiques spécifiques à CFDEP.

Deuxièmement, nous présentons notre algorithme (PSD) pour la recherche de la structure de coalitions optimale, qui est fondé sur une approche de **Parcours Sélectif** guidé par les **Dépendances**. Le parcours de l'espace de recherche se fait par itération où, à chaque itération, l'ensemble des dépendances entre les tâches est mis à jour à chaque fois qu'une tâche est sélectionnée pour une coalition. Ainsi, l'ensemble des tâches restantes est également mis à jour à chaque itération ; ce qui permet, d'une part, d'identifier les tâches pour lesquelles il est possible de former des coalitions, et d'autre part, d'identifier celles pour lesquelles des dépen-

dances conflictuelles avec les tâches déjà sélectionnées empêchent la formation de coalitions. Ces dernières sont alors exclues de l'espace de recherche durant cette itération. En procédant ainsi, l'ensemble des structures de coalitions à explorer évolue et change dynamiquement en fonction de la structure explorée à chaque itération.

À l'issue de ce chapitre, nous présentons l'évaluation expérimentale de l'algorithme PSD qui démontre son efficacité pour réduire le nombre d'itérations nécessaires pour trouver la structure de coalitions optimale. Nous utilisons tout au long de ce chapitre, le cas d'étude CMATP présenté dans la section 3.3.2.

6.2 CFDEP dans le contexte centralisé

En considérant que les individus sont des agents et leurs déplacements sont des tâches à accomplir, nous ramenons le problème CFDEP à un problème de formation de coalitions avec externalités. Sa résolution peut donc se faire par des techniques de formation de coalitions, notamment par la génération de structures de coalitions (GSC). Cependant, la particularité du CFDEP par rapport aux travaux existants sur GSC [62, 93, 94, 59, 58, 63, 61] est que les agents ont plusieurs alternatives pour atteindre un but, qui elles-mêmes sont composées de plusieurs tâches interdépendantes. De plus, une tâche peut faire partie de plusieurs alternatives à la fois. À la fin du processus de formation de coalitions, chaque agent a_i retient une seule alternative pour accomplir ses tâches. En d'autres termes, un agent a_i n'a pas seulement à chercher des agents avec qui il vaut mieux former des coalitions, mais doit aussi chercher les tâches possibles ainsi que les groupes d'agents possibles pour chaque tâche. De plus, il doit tenir compte des dépendances entre ses tâches ; ce qui constitue une différence majeure avec le problème GSC tel que défini dans la littérature [62, 93, 94, 59, 58, 63, 61].

Rappelons que selon Rahwan et al. [62], le problème GSC dans les jeux à fonction de partition (PFG) traite la question comme suit : comment un groupe d'agents peut-il se diviser en sous-groupes afin d'améliorer leur performance ? Autrement dit, l'accent est mis sur la partition de l'ensemble des agents en coalitions mutuel-

lement disjointes de sorte que la récompense totale des coalitions résultantes soit maximisée. Généralement, le problème de GSC consiste à partitionner les agents en groupes disjoints afin d'améliorer leur performance [62].

Exemple 6.1. Par exemple, pour un ensemble de 04 agents $\{a_1, a_2, a_3, a_4\}$, il y a 15 structures de coalitions possibles (*cf.* Fig. 6.1).

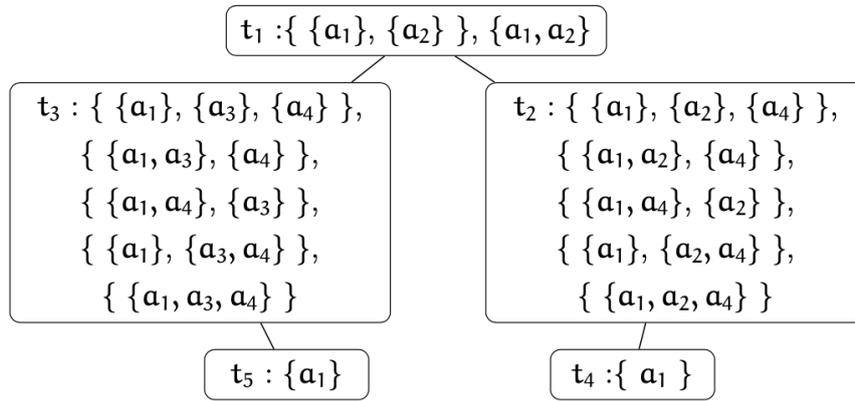
$$\begin{array}{lll}
 \{\{a_1\}\{a_2\}\{a_3\}\{a_4\}\} & \{\{a_1\}\{a_2, a_4\}\{a_3\}\} & \{\{a_1\}\{a_2, a_3, a_4\}\} \\
 \{\{a_1, a_2\}\{a_3\}\{a_4\}\} & \{\{a_1\}\{a_2\}\{a_3, a_4\}\} & \{\{a_1, a_2\}\{a_3, a_4\}\} \\
 \{\{a_1, a_3\}\{a_2\}\{a_4\}\} & \{\{a_1, a_2, a_3\}\{a_4\}\} & \{\{a_1, a_3\}\{a_2, a_4\}\} \\
 \{\{a_1, a_4\}\{a_2\}\{a_3\}\} & \{\{a_1, a_2, a_4\}\{a_3\}\} & \{\{a_1, a_4\}\{a_2, a_3\}\} \\
 \{\{a_1\}\{a_2, a_3\}\{a_4\}\} & \{\{a_1, a_3, a_4\}\{a_2\}\} & \{\{a_1, a_2, a_3, a_4\}\}
 \end{array}$$

Fig. 6.1 – Les structures de coalitions pour 4 agents.

Rappelons aussi que le terme externalité dans le domaine des jeux à fonction de partition signifie que la valeur d'une coalition c change lors de la fusion de deux autres coalitions existantes [62]. Cela veut dire que la valeur d'une coalition peut être impactée par la formation d'autres coalitions.

Exemple 6.2. Par exemple, ayant deux structures de coalitions $cs = \{c_1, c_2, c_3\}$ et $cs' = \{c_1, c_2 \cup c_3\}$ la valeur de c_1 peut être différente dans cs de celle dans cs' .

Dans le cadre de nos travaux sur CFDEP, le problème de GSC se concentre sur l'identification des groupes de tâches bénéfiques ainsi que des groupes d'agents — pas nécessairement disjoints — pour les accomplir. Alors que dans les PFGs, les coalitions mutuellement disjointes sont formées en divisant l'ensemble des agents où chaque tâche étant attribuée à un groupe d'agents. Dans CFDEP, nous n'attribuons pas nécessairement toutes les tâches puisque nous ne cherchons pas nécessairement à les réaliser entièrement. De plus, les agents peuvent être impliqués dans plusieurs coalitions. Les dépendances entre les tâches et la pluralité des sous-ensembles possibles de tâches permettant d'atteindre un but ont une influence prépondérante ici.


 Fig. 6.2 – L'espace de recherche de l'agent a_1 .

Fondamentalement, le problème de GSC dans CFDEP est différent du problème de GSC dans les PFGs tel qu'il est décrit dans [62] comme suit :

1. Dans PFGs, les coalitions formées sont mutuellement disjointes (*cf.* Fig. 6.1). Dans CFDEP, les coalitions ne sont pas nécessairement disjointes.
2. Dans PFGs, l'espace de recherche demeure le même pour tous les agents. Dans CFDEP, chaque agent dispose de son propre espace de recherche (*cf.* Fig. 6.2).
3. Dans PFGs, l'ensemble des tâches est commun à tous les agents. Chaque agent peut être affecté à n'importe quelle tâche selon les valeurs des coalitions possibles et, à la fin, toutes les tâches seront exécutées par au moins un agent. Dans CFDEP, les tâches ne sont pas toutes communes à tous les agents et l'ensemble final des tâches à exécuter pour chaque agent est un sous-ensemble de son ensemble initial de tâches.
4. Dans PFGs, les externalités représentent le changement de la valeur d'une coalition à la suite de la fusion de deux autres coalitions coexistantes. Dans CFDEP, les externalités représentent l'influence d'une coalition sur la possibilité (voire l'impossibilité) d'une autre coalition à se former. C'est-à-dire, la possibilité de former (ou pas) une coalition dépend de l'existence (ou non) d'une autre coalition.

6.3 PSD : un algorithme centralisé de formation de coalitions

Dans cette section, nous présentons notre algorithme de *Parcours Sélectif guidé par les Dépendances* (PSD) (Algorithme 6) pour former une série de coalitions séquentielles interdépendantes pour le CFDEP. Mais avant d'entrer dans les détails de PSD, nous introduisons quelques concepts fondamentaux à sa compréhension.

Soit un ensemble de n agents $\mathcal{A} = \{a_i | i = 1, \dots, n\}$, ayant chacun un but $g_i \in \mathcal{G}$ à atteindre. Chacun des agents dispose d'un ensemble de tâches \mathcal{T}_i lui permettant d'atteindre son but à travers la réalisation d'un sous-ensemble de tâches parmi les tâches dans \mathcal{T}_i . C'est-à-dire, chaque agent a_i dispose de plusieurs sous-ensembles de tâches $\mathcal{T}_i^s \subseteq \mathcal{T}_i$ (plusieurs alternatives¹ ($\alpha_{i,j}$)) le menant à la réalisation de son but g_i . L'ensemble des agents \mathcal{A} forme un vecteur, que nous appelons *Vecteur d'agents* et notons $V_{\mathcal{A}} = \langle a_1, a_2, \dots, a_n \rangle$.

Nous dénotons l'union des ensembles de tâches de tous les agents par $\mathcal{T} = \{t_j | j = 1, \dots, m\}$, de taille m . Il existe une relation de dépendance, notée \succ , entre les tâches dans \mathcal{T} . $t \succ t'$ veut dire que t doit être réalisée pour pouvoir réaliser t' . Ainsi, l'ensemble des dépendances entre les tâches de l'exemple 3.3.2 est illustré dans le tableau Tab. 6.1. Notons que la relation \succ est transitive.

Agents	Dépendances
a_1	$t_1 \succ t_3 \succ t_5$
	$t_1 \succ t_2 \succ t_5$
a_2	$t_1 \succ t_2$
a_3	$t_3 \succ t_7 \succ t_8$
a_4	$t_3 \succ t_{10}$
	$t_2 \succ t_{11}$

Tab. 6.1 – Les dépendances entre les tâches.

L'ensemble \mathcal{T} forme un *Vecteur de tâches* $V_{\mathcal{T}} = \langle t_1, t_2, \dots, t_m \rangle$. La réalisation

1. Nous gardons la définition introduite dans le chapitre 04 pour une alternative de tâche.

de chaque tâche $t \in \mathcal{T}_i$ engendre un coût $\phi(t)$, qui représente la somme d'un coût privé $\phi^{\text{priv}}(t)$ et d'un coût public $\phi^{\text{pub}}(t)$. Ainsi, pour atteindre un but, chaque alternative engendre un coût global $\phi(\alpha)$ qui est la somme des coûts de ses tâches. Donc, l'objectif de chaque agent α_i est d'atteindre son but g_i avec un minimum de coût possible pour $\phi(\alpha)$. Il est question alors, pour chacun, d'identifier en parallèle et à la fois, les sous-ensembles de tâches \mathcal{T}_i^s où $\mathcal{T}_i^s \subseteq \mathcal{T}_i \subseteq \mathcal{T}$, lui permettant d'atteindre son but avec moins de coûts possibles et les groupes d'agents avec qui il est possible de réaliser chacune (ou plusieurs à la fois) des tâches le menant à son but.

Exemple 6.3. Soit l'ensemble d'agents $\mathcal{A} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ et l'ensemble des tâches $\mathcal{T} = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}\}$. Nous avons le vecteur d'agents $V_{\mathcal{A}} = \langle \alpha_1, \alpha_2, \dots, \alpha_n \rangle$ et celui des tâches $V_{\mathcal{T}} = \langle t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11} \rangle$. Les ensembles de tâches des agents $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ sont respectivement : $\mathcal{T}_1 = \{t_1, t_2, t_3, t_4, t_5\}$, $\mathcal{T}_2 = \{t_1, t_2, t_6\}$, $\mathcal{T}_3 = \{t_3, t_7, t_8, t_9\}$ et $\mathcal{T}_4 = \{t_2, t_3, t_9, t_{10}, t_{11}\}$. Les sous-ensembles de tâches qui permettent à l'agent α_1 d'atteindre son but sont : $\mathcal{T}_1^1 = \{t_1, t_3, t_5\}$ et $\mathcal{T}_1^2 = \{t_1, t_2, t_4\}$.

La question pour l'agent α_1 est donc, des deux sous-ensembles \mathcal{T}_1^1 et \mathcal{T}_1^2 , lequel est le mieux pour atteindre son but et avec quels groupes d'agents pour chacune des tâches? Comme nous pouvons le constater, il existe parmi les tâches dans \mathcal{T} , plusieurs tâches communes à différents agents. Nous dénotons par \vec{v}_{t_j} l'ensemble des agents α_i , où $t_j \in \mathcal{T}_i$, que nous appelons : *vecteur d'agents de tâche*.

Définition 6.1. - Vecteur d'agents de tâche. Un *Vecteur d'agents de tâche* d'une tâche t_j , $\vec{v}_{t_j} = \langle x_1, x_2, \dots, x_{n'} \rangle$, de la taille n' , est un vecteur de n' agents, où : $\forall x_{i'}, \exists \alpha_i \in \mathcal{A} : x_{i'} = \alpha_i \wedge t_j \in \mathcal{T}_i$.

Par exemple, le *Vecteur d'agents de tâche* de t_1 est : $\vec{v}_{t_1} = \langle \alpha_1, \alpha_2 \rangle$. En d'autres termes, \vec{v}_{t_1} contient tous les agents ayant la tâche t_1 dans leur ensemble de tâches. Ainsi, \vec{v}_{t_1} représente l'ensemble des agents réalisateurs de t_1 . C'est-à-dire, l'ensemble des agents que t_1 peut mener à atteindre, conjointement, leurs buts.

Définition 6.2. - Tâche conjointe. Une tâche t_j est dite conjointe si $|\vec{v}_{t_j}| \geq 2$. C'est-à-dire, s'il existe au moins deux agents l'ayant dans leur ensemble de tâches.

L'ensemble de toutes les tâches conjointes $t_j \in \mathcal{T}$ forme un vecteur que nous dénotons par ν . Les éléments dans ν sont ordonnés suivant l'ordre défini dans $V_{\mathcal{T}}$. Dans l'exemple 6.3, nous avons $\nu = \langle t_1, t_2, t_3, t_9 \rangle$.

Rappelons que chaque agent a_i possède son propre ensemble de tâches \mathcal{T}_i parmi lesquelles il cherche un sous-ensemble (une alternative) de tâches \mathcal{T}_i^s qu'il peut réaliser conjointement avec d'autres agents, pour atteindre son but g_i . Inversement, chaque tâche peut être réalisée par plusieurs sous-ensembles d'agents possibles. Nous appelons ces sous-ensembles : des partitions.

Exemple 6.4. Considérons la tâche t_2 . Nous avons donc $\vec{\nu}_{t_2} = \langle a_1, a_2, a_4 \rangle$. Alors, l'ensemble des partitions d'agents possibles pour réaliser la tâche t_2 est : $\{\emptyset, \{a_1\}, \{a_2\}, \{a_4\}, \{a_1, a_2\}, \{a_1, a_4\}, \{a_2, a_4\}, \{a_1, a_2, a_4\}\}$.

Pour résoudre le problème de la formation de coalitions tel qu'il est décrit dans la section 3.3.2, il suffit alors de former des coalitions pour les tâches $t_j \in \nu$. Cela va réduire considérablement l'espace de recherche. De plus, il suffit de considérer pour chaque tâche t_j uniquement les agents $a_i \in \vec{\nu}_{t_j}$ pour rechercher les coalitions possibles pour t_j . Ces deux points constituent l'idée de base de notre algorithme PSD.

Définition 6.3. - Coalition. Une *coalition* est un tuple $c = (t_j, \bar{\nu}_{t_j}^q)$, où $t_j \in \nu$, et $\bar{\nu}_{t_j}^q = \langle x'_1, x'_2, \dots, x'_{|\vec{\nu}_{t_j}|} \rangle$, de la taille $|\vec{\nu}_{t_j}|$, est la q ème instance de $\vec{\nu}_{t_j} = \langle x_1, x_2, \dots, x_{|\vec{\nu}_{t_j}|} \rangle$, où :

$$x'_i = \begin{cases} 1 & \text{si : l'agent } x_i \in \vec{\nu}_{t_j} \\ 0 & \text{si : l'agent } x_i \notin \vec{\nu}_{t_j} \end{cases}$$

Nous appelons $\bar{\nu}_{t_j}^q$ *Instance de Vecteur d'agents*, dans lequel, chaque élément x_i fait référence à la présence ou pas d'un agent a_i dans $\vec{\nu}_{t_j}$. Notons que, pour un $\vec{\nu}_{t_j}$ il peut y avoir plusieurs instances ($\bar{\nu}_{t_j}^q$). Par exemple, des instances possibles de $\vec{\nu}_{t_1}$ sont : $\{\bar{\nu}_{t_1}^1 = \langle 1, 1 \rangle, \bar{\nu}_{t_1}^2 = \langle 1, 0 \rangle, \bar{\nu}_{t_1}^3 = \langle 0, 1 \rangle, \bar{\nu}_{t_1}^4 = \langle 0, 0 \rangle\}$.

Nous dénotons par C l'ensemble de toutes les coalitions possibles. Nous avons donc : $|C| = \sum_{j=1}^{j=|\mathcal{T}|} (2^{|\vec{\nu}_{t_j}|} - 1)$. Le tableau Tab. 6.2 illustre l'ensemble des coalitions possibles pour les tâches dans \mathcal{T} .

Vecteur d'agents de tâche (\vec{v}_{t_j})	Coalitions : $c = (t_j, \vec{v}_{t_j}^q)$
$\vec{v}_{t_1} = \langle a_1, a_2 \rangle$	$c_{1,1} = (t_1, \langle 1, 1 \rangle), c_{1,2} = (t_1, \langle 1, 0 \rangle), c_{1,3} = (t_1, \langle 0, 1 \rangle)$
$\vec{v}_{t_2} = \langle a_1, a_2, a_4 \rangle$	$c_{2,1} = (t_2, \langle 1, 1, 1 \rangle), c_{2,2} = (t_2, \langle 1, 1, 0 \rangle), c_{2,3} = (t_2, \langle 1, 0, 1 \rangle), c_{2,4} = (t_2, \langle 0, 1, 1 \rangle), c_{2,5} = (t_2, \langle 0, 1, 0 \rangle), c_{2,6} = (t_2, \langle 0, 0, 1 \rangle), c_{2,7} = (t_2, \langle 1, 0, 0 \rangle)$
$\vec{v}_{t_3} = \langle a_1, a_3, a_4 \rangle$	$c_{3,1} = (t_3, \langle 1, 1, 1 \rangle), c_{3,2} = (t_3, \langle 1, 1, 0 \rangle), c_{3,3} = (t_3, \langle 1, 0, 1 \rangle), c_{3,4} = (t_3, \langle 0, 1, 1 \rangle), c_{3,5} = (t_3, \langle 0, 1, 0 \rangle), c_{3,6} = (t_3, \langle 0, 0, 1 \rangle), c_{3,7} = (t_3, \langle 1, 0, 0 \rangle)$
$\vec{v}_{t_4} = \langle a_1 \rangle$	$c_{4,1} = (t_4, \langle 1 \rangle)$
$\vec{v}_{t_5} = \langle a_1 \rangle$	$c_{5,1} = (t_5, \langle 1 \rangle)$
$\vec{v}_{t_6} = \langle a_2 \rangle$	$c_{6,1} = (t_6, \langle 1 \rangle)$
$\vec{v}_{t_7} = \langle a_3 \rangle$	$c_{7,1} = (t_7, \langle 1 \rangle)$
$\vec{v}_{t_8} = \langle a_3 \rangle$	$c_{8,1} = (t_8, \langle 1 \rangle)$
$\vec{v}_{t_9} = \langle a_3, a_4 \rangle$	$c_{9,1} = (t_9, \langle 1, 1 \rangle), c_{9,2} = (t_9, \langle 1, 0 \rangle), c_{9,3} = (t_9, \langle 0, 1 \rangle)$
$\vec{v}_{t_{10}} = \langle a_4 \rangle$	$c_{10,1} = (t_{10}, \langle 1 \rangle)$

 Tab. 6.2 – L'ensemble des coalitions possibles pour les tâches dans \mathcal{T} .

Nous définissons la fonction $\Phi(\vec{v}_{t_j}^q) = \sum_{j=1}^n x_j'$ qui retourne le nombre d'agents dans une *Instance de Vecteur d'agents* $\vec{v}_{t_j}^q$. Par exemple, pour $\vec{v}_{t_1}^1 = \langle 1, 1 \rangle$, nous avons $\Phi(\vec{v}_{t_1}^1) = 2$ ($\vec{v}_{t_1}^2 = \langle 1, 0 \rangle$ et $\Phi(\vec{v}_{t_1}^2) = 1$). Nous dénotons également par $\Lambda(\vec{v}_{t_j}^q)$ l'ensemble des agents présents dans $\vec{v}_{t_j}^q$. Par exemple, $\Lambda(\vec{v}_{t_1}^1) = \{a_1, a_2\}$ et $\Lambda(\vec{v}_{t_1}^2) = \{a_1\}$.

6.3.1 Partitionnement

Le partitionnement de l'ensemble des agents pour former des coalitions constitue la base de la plupart des travaux sur CSG [62, 93, 94, 59, 58, 63, 61]. Dans CFDEP, le partitionnement doit tenir compte des dépendances entre les tâches. Il en résulte alors que l'ensemble des dépendances entre les tâches évolue au fur et à mesure des choix faits sur les tâches à réaliser conjointement.

Définition 6.4. - Partition possible. Une partition possible P_{t_j} est une *Instance du Vecteur d'agents de tâche* $\vec{v}_{t_j}^k$ où $\forall k | k = 1, \dots, |\vec{v}_{t_j}| : (\Phi(\vec{v}_{t_j}^k) \neq 1)$.

C'est-à-dire, une P_{t_j} d'une tâche t_j est une *Instance du Vecteur d'agents* \vec{v}_{t_j} , qui est possible à un moment donné lors du parcours de l'espace de recherche en fonction de l'évolution dans l'espace des dépendances entre les tâches.

Exemple 6.5. Soit le vecteur d'agents \vec{v}_{t_3} de la tâche t_3 . Nous avons $\vec{v}_{t_3} = \langle t_1, t_3, t_4 \rangle$. Alors, $P_{t_3} = \{P_{t_3}^1 = \langle 0, 0, 0 \rangle, P_{t_3}^5 = \langle 1, 1, 0 \rangle, P_{t_3}^6 = \langle 1, 0, 1 \rangle, P_{t_3}^7 = \langle 0, 1, 1 \rangle, P_{t_3}^8 = \langle 1, 1, 1 \rangle\}$. Le tableau Tab. 6.3 illustre l'interprétation de chaque représentation d'une partition.

P_{t_3}	Description
$\langle 0, 0, 0 \rangle$	Le cas vide. Veut dire que la tâche t_3 sera soit réalisée par un seul agent (peu importe lequel), soit laissée et ne sera pas réalisée.
$\langle 1, 1, 0 \rangle$	les agents a_1, a_3 vont réaliser la tâche t_3 conjointement.
$\langle 1, 0, 1 \rangle$	les agents a_1, a_4 vont réaliser la tâche t_3 conjointement.
$\langle 0, 1, 1 \rangle$	les agents a_3, a_4 vont réaliser la tâche t_3 conjointement.
$\langle 1, 1, 1 \rangle$	tous les agents a_1, a_3, a_4 vont réaliser la tâche t_3 conjointement.

Tab. 6.3 – Interprétation des partitions possibles pour la tâche t_3 .

6.3.2 Réduction de l'espace de recherche

La condition $(\Phi(\vec{v}_{t_j}^k) = 0 \vee \Phi(\vec{v}_{t_j}^k) > 1)$ signifie que toutes les possibilités ayant un seul agent $(\Phi(\vec{v}_{t_j}^k) = 1)$ ne seront pas considérées. Il convient de noter que dans l'exemple 6.5, l'ensemble des partitions possibles ne comporte pas les instances $\langle 1, 0, 0 \rangle, \langle 0, 1, 0 \rangle$, et $\langle 0, 0, 1 \rangle$. En effet, dans la recherche de la structure de coalitions optimale, l'algorithme PSD ne traite pas (ignore) les cas où une tâche t est réalisée par un seul des agents dans \vec{v}_t , mais les considère tous en une seule représentation. Tous ces cas d'instances ($\langle 1, 0, 0 \rangle, \langle 0, 1, 0 \rangle$, et $\langle 0, 0, 1 \rangle$) sont représentés par l'instance $\langle 0, 0, 0 \rangle$ qui signifie que la tâche t est libre. C'est-à-dire, l'agent qui va la réaliser importe peu. Toutefois, son ensemble d'agents réalisateur \vec{v}_t sera mis à jour au fur et à mesure du parcours de l'espace de recherche. Pour explorer l'ensemble des partitions, nous ne considérons que les

partitions qui vérifient la condition $\Phi(\bar{\mathbf{v}}_{t_1}^k) \neq 1$. En effet, lors du parcours, il n'est pas nécessaire de tenir compte des tâches qui seront réalisées individuellement par les agents. Par contre, l'information essentielle dont il faut tenir compte est de savoir si cette tâche sera réalisée ou pas. Pour cela, avoir l'information $\Phi(\bar{\mathbf{v}}_{t_1}^k) = 0$ où $\Phi(\bar{\mathbf{v}}_{t_1}^k) > 1$ suffit. De cette façon, la taille de l'espace de recherche à explorer peut être considérablement réduite. En éliminant les cas où $\Phi(\bar{\mathbf{v}}_{t_1}^k) = 1$, on réduit l'ensemble des partitions à $|\mathbf{C}| = \sum_{j=1}^{j=|\mathcal{T}|} \left(2^{|\bar{\mathbf{v}}_{t_j}| - |\bar{\mathbf{v}}_{t_j}|} \right)$. Le tableau Tab. 6.4 illustre l'ensemble des partitions à considérer pour parcourir l'espace de recherche des tâches dans \mathbf{v} .

$\bar{\mathbf{v}}_{t_j}$	Partitions possibles
$\bar{\mathbf{v}}_{t_1} = \langle \mathbf{a}_1, \mathbf{a}_2 \rangle$	$\langle 0, 0 \rangle, \langle 1, 1 \rangle$
$\bar{\mathbf{v}}_{t_2} = \langle \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_4 \rangle$	$\langle 0, 0, 0 \rangle, \langle 1, 1, 0 \rangle, \langle 1, 0, 1 \rangle, \langle 1, 0, 1 \rangle, \langle 1, 1, 1 \rangle$
$\bar{\mathbf{v}}_{t_3} = \langle \mathbf{a}_1, \mathbf{a}_3, \mathbf{a}_4 \rangle$	$\langle 0, 0, 0 \rangle, \langle 1, 1, 0 \rangle, \langle 1, 0, 1 \rangle, \langle 1, 0, 1 \rangle, \langle 1, 1, 1 \rangle$
$\bar{\mathbf{v}}_{t_9} = \langle \mathbf{a}_3, \mathbf{a}_4 \rangle$	$\langle 0, 0 \rangle, \langle 1, 1 \rangle$

Tab. 6.4 – Les partitions possibles pour les tâches dans \mathbf{v} .

Ainsi, l'ensemble des coalitions à explorer deviendra alors $|\mathbf{C}| = \sum_{j=1}^{j=|\mathcal{V}|} (2^{|\bar{\mathbf{v}}_{t_j}| - (|\bar{\mathbf{v}}_{t_j}| + 1)})$. Le tableau Tab. 6.5 illustre l'ensemble des coalitions possibles pour l'ensemble des tâches $t_j \in \mathbf{v}$. Nous remarquons que, partant d'un ensemble de coalitions d'une taille 26 (*cf.* Tab. 6.2) et avec les concepts qu'on a introduits ci-dessus, nous réduisons ce nombre pour ne considérer au final qu'un sous-ensemble de 10 coalitions (*cf.* Tab. 6.5).

Définition 6.5. - Structure de coalitions. Une *structure de coalitions* $\mathbf{cs} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{|\mathbf{cs}|}\}$ est définie comme l'ensemble des coalitions permettant à tous les agents dans \mathcal{A} d'atteindre leurs bus.

Exemple 6.6. Un exemple d'une structure de coalitions est : $\mathbf{cs}_1 = \{\mathbf{c}_1, \mathbf{c}_4, \mathbf{c}_7\}$.

Nous dénotons par $\Gamma(\mathbf{cs}) \subseteq \mathcal{T}$ l'ensemble de toutes les tâches pour lesquelles sont formées les coalitions dans \mathbf{cs} . Nous avons donc : $\Gamma(\mathbf{cs}_1) = \{t_1, t_2, t_3\}$. Par ailleurs, pour chaque \mathbf{cs} , $\Gamma(\mathbf{cs}) \subseteq \mathbf{v}$.

\vec{v}_{t_j}	Coalitions : $c = (t_j, P_{t_j})$
$\vec{v}_{t_1} = \langle a_1, a_2 \rangle$	$c_{1,1} = (t_1, \langle 1, 1 \rangle)$
$\vec{v}_{t_2} = \langle a_1, a_2, a_4 \rangle$	$c_{2,1} = (t_2, \langle 1, 1, 1 \rangle), c_{2,2} = (t_2, \langle 1, 1, 0 \rangle), c_{2,3} = (t_2, \langle 1, 0, 1 \rangle), c_{2,4} = (t_2, \langle 0, 1, 1 \rangle)$
$\vec{v}_{t_3} = \langle a_1, a_3, a_4 \rangle$	$c_{3,1} = (t_3, \langle 1, 1, 1 \rangle), c_{3,2} = (t_3, \langle 1, 1, 0 \rangle), c_{3,3} = (t_3, \langle 1, 0, 1 \rangle), c_{3,4} = (t_3, \langle 0, 1, 1 \rangle)$
$\vec{v}_{t_9} = \langle a_3, a_4 \rangle$	$c_{9,1} = (t_9, \langle 1, 1 \rangle)$

 Tab. 6.5 – L'ensemble des coalitions possibles pour les tâches dans ν .

6.3.3 Parcours de l'espace de recherche

Pour explorer l'espace de recherche \mathcal{S} il est impératif de le parcourir pour calculer la fonction caractéristique des cs . Compte tenu de la taille de \mathcal{S} qui peut être très importante, il est essentiel d'utiliser un mécanisme qui permet de le parcourir sans énumérer et maintenir en mémoire toutes les possibilités. Pour cela, nous introduisons deux concepts, qui sont : le *Vector minimum index* $\nu^{\min} = \langle x_1, x_2, \dots, x_n \rangle$ où x_j indique la coalition courante explorée parmi l'ensemble des coalitions possibles pour la tâche t_j et le *Vector maximum index* $\nu^{\max} = \langle x_1, x_2, \dots, x_n \rangle$ où $x_j = |P_{t_j}|$ qui est le vecteur des tailles des ensembles de coalitions possibles de chaque tâche dans ν .

Exemple 6.7. Soit le vecteur $\nu = \langle t_1, t_2, t_3, t_9 \rangle$. Alors, $\nu^{\max} = \langle 2, 5, 5, 2 \rangle$. Maintenant, supposons que pendant le processus de parcours nous avons le *Vector minimum index* qui est $\nu^{\min} = \langle 1, 1, 1, 1 \rangle$ ². Cela signifie que pour toutes les tâches, l'algorithme considèrera la première coalition pour chaque tâche. La valeur possible du *Vector minimum index* à la prochaine itération sera alors $\nu^{\min} = \langle 1, 1, 1, 2 \rangle$. La figure Fig. 6.3 illustre un exemple de parcours sur \mathcal{S} avec les vecteurs ν^{\min} et ν^{\max} .

Rappelons qu'une tâche t_j peut affecter une autre tâche $t_{j'}$ en rendant sa réalisation par un agent a_i possible ou impossible, suivant les dépendances entre les tâches dans \mathcal{T} . Pour faciliter la prise en compte de ces dépendances, nous définissons pour chaque tâche t_j l'ensemble *Task exclusive Tasks* (TeT).

2. Notons que la valeur 1 dans ν^{\min} fait référence à l'ensemble vide d'agents.

$v_{\mathcal{T}} :$	t_1	t_2	t_3	t_9
v^{\max}	2	5	5	2
v_1^{\min}	1	1	1	1
v_2^{\min}	1	1	1	2
v_3^{\min}	1	1	2	1
v_4^{\min}	1	1	2	2
v_5^{\min}	1	1	3	1
v_6^{\min}	1	1	3	2
v_7^{\min}	1	1	4	1
v_8^{\min}	1	1	4	2
...				
...				
...				
v_k^{\min}	2	5	5	2

Fig. 6.3 – Évolution de v^{\max} et v^{\min} pendant le parcours de \mathcal{S} .

Définition 6.6. - Task exclusive Tasks. L'ensemble *Task exclusive Tasks* (TeT_j) d'une tâche t_j selon le vecteur v est l'ensemble de toutes les tâches $t_{j'}$ qui ne peuvent pas être réalisables si t_j est sélectionnée pour être réalisée où $j > j'$ et $\forall t_{j'} : t_{j'} \in \text{TeT}(t_j) \Rightarrow \Phi(\vec{v}_{t_{j'}}) \geq 2$.

En effet, pour un même agent, si des dépendances conflictuelles existent entre deux tâches t_j et $t_{j'}$, alors si une des deux est sélectionnée pour lui former une coalition, l'autre sera inévitablement irréalisable. Par exemple, $\text{TeT}_2 = \{t_3, t_9\}$ et $\text{TeT}_9 = \{t_2, t_3\}$. Le tableau Tab. 6.6 illustre le TeT des tâches dans v .

Tâche	TeT
t_1	{ }
t_2	{ t_3, t_9 }
t_3	{ t_2, t_9 }
t_9	{ t_2, t_3 }

 Tab. 6.6 – Les ensembles TeT pour les tâches dans ν .

Définition 6.7. - Path. Le $Path_j$ d'une tâche t_j selon le vecteur ν est l'ensemble des tâches explorées $t_{j'}$ où $j' \leq j$ et $\forall t' \in Path_j \Rightarrow (\Phi(\vec{\nu}_{t_j}) \geq 2 \wedge t' \notin TeT_j)$.

Pendant le parcours de \mathcal{S} , $Path$ représente toutes les tâches explorées et maintenues comme des tâches à réaliser en coalition. En d'autres termes, le $Path_j$ de la tâche t_j est l'ensemble des tâches qui précèdent la tâche t_j dans le vecteur ν . Inversement, nous définissons $Rest_j$ de la tâche t_j comme l'ensemble des tâches qui succèdent la tâche t_j dans le vecteur ν . Ces deux concepts sont importants dans notre algorithme puisqu'ils permettent d'établir un parcours simplifié de l'espace de recherche. De cette manière, il ne sera pas nécessaire d'énumérer toutes les coalitions possibles; ce qui aurait nécessité un espace mémoire et un temps de calcul considérables. Donc, il serait suffisant de ne garder que les indices des coalitions à travers le vecteur ν^{min} . Ainsi, chaque ensemble d'indices permettra de représenter implicitement une structure de coalitions. Et parcourir l'espace de recherche reviendrait à une simple opération d'incrémentatation des indices sur ν^{min} .

Exemple 6.8. Les ensembles $Path$ des tâches dans ν sont : $Path_1 = \emptyset$, $Path_2 = \{t_1\}$, $Path_3 = \{t_1, t_2\}$, $Path_9 = \{t_1, t_2, t_3\}$ et leurs ensembles $Rest$ sont : $Rest_1 = \{t_2, t_3, t_9\}$, $Rest_2 = \{t_3, t_9\}$, $Rest_3 = \{t_9\}$, $Rest_9 = \emptyset$.

La dépendance entre deux tâches t_j et $t_{j'}$ où $t_j \in TeT_{j'}$ (ou $t_{j'} \in TeT_j$) affecte non seulement l'(im)possibilité des deux tâches l'une sur l'autre, mais elle affecte également l'(im)possibilité des agents a_i qui peuvent les réaliser. Notamment, lorsqu'une des deux tâches est déjà affectée à un groupe d'agents (une coalition). Clairement, prendre en compte cet aspect va permettre de réduire davantage l'espace de recherche. Pour cela, nous définissons le concept de *Task exclusive Agents (TeA)*.

Définition 6.8. - Task exclusive Agents (TeA). le *Task exclusive Agents* (TeA_j) d'une tâche $t_j \in \mathcal{V}$ est l'ensemble des agents qui ne peuvent plus réaliser (ou participer à la réalisation) de la tâche t_j à cause du fait qu'ils soient affectés à (au moins) une autre tâche $t_{j'} \in \mathcal{V}$ où $t_j \in TeT_{j'}$.

TeA est construit selon l'ensemble des tâches exclusives de la tâche t . Il contient alors les agents affectés à une (ou plusieurs tâches) t' dans l'ensemble des tâches exclusives de la tâche t .

Exemple 6.9. Par exemple, supposons que l'agent a_1 fait partie de la coalition $c_7 = (t_3, \{a_1, a_3\})$. Comme $t_2 \in TeT_3$, cela fait que l'agent a_1 ne va pas pouvoir réaliser (ou participer à la réalisation) de la tâche t_2 ; c.-à-d. $a_1 \in TeA_2$.

Maintenant que nous avons introduit les concepts fondamentaux nécessaires à la compréhension de l'algorithme PSD, nous passons dans la section suivante à la présentation des étapes de son fonctionnement.

6.3.4 Algorithme PSD

Sur la base des concepts définis ci-dessus, nous proposons un algorithme de *Parcours Sélectif guidé par les Dépendances* (PSD) (Algorithme 6) pour former une série de coalitions séquentielles interdépendantes pour le CFDEP. PSD, d'une part, prend en compte les dépendances entre les tâches et, d'autre part, permet de parcourir toutes les structures de coalitions possibles de manière itérative en excluant à chaque itération l'ensemble des coalitions conflictuelles. Alors, en tenant compte des dépendances entre les tâches, l'algorithme PSD permet un parcours sélectif de l'espace de recherche dans le sens où il n'est pas nécessaire de le faire pour toutes les tâches, mais de réduire la taille de cet espace en fonction des tâches sélectionnées à chaque itération. En effet, seul l'ensemble des tâches qui peuvent être réalisées conjointement sont prises en compte. Cet ensemble est mis à jour à chaque itération. De cette manière, l'algorithme PSD permet de réduire le nombre de structures de coalitions à parcourir en ne considérant que celles qui sont possibles en fonction des coalitions déjà sélectionnées.

Algorithme 6 PSD()

Require: \mathcal{T}, \mathcal{A} : List

```

1: calculer  $\nu$ 
2:  $\forall t_j \in \nu$  : calculer  $\vec{\nu}_{t_j}$ 
3:  $\forall t_j \in \nu$  : calculer  $P_{t_j}$ 
4:  $\varphi \leftarrow 0$ ;
5:  $\varphi_{\text{current}} \leftarrow 0$ ;
6: end  $\leftarrow$  false;
7: initialiser( $\text{TeT}, \nu^{\text{max}}, \nu^{\text{min}}$ );
8: while  $\neg$ end do
9:   for all  $t_i \in \nu$  do
10:     $P_{t_i} \leftarrow \text{getNextComb}(t_i, P_t)$ ;
11:     $c_i \leftarrow \text{newCoalition}(t_i, P_{t_i})$ ;
12:    if  $(\Phi(P_{t_i}) \geq 2)$  then
13:       $\text{setCollectiveRealization}(c_i)$ ;
14:    else
15:       $\text{setIndividualRealization}(c_i)$ ;
16:    end if
17:     $\text{tempCS} \leftarrow \text{tempCS} \cup c_i$ ;
18:  end for
19:   $\varphi_{\text{current}} \leftarrow \text{computeEffeciency}(\text{tempC})$ ;
20:  if  $(\varphi_{\text{current}} > \varphi)$  then
21:     $\varphi \leftarrow \varphi_{\text{current}}$ ;
22:     $\text{bestCS} \leftarrow \text{tempCS}$ ;
23:  end if
24:   $\text{updateVectors}(\text{end})$ ;
25: end while
26: return  $\text{bestCS}$ 

```

Dans la phase d'initialisation de l'algorithme PSD (Algorithme 6), nous calculons le vecteur des tâches jointes ν , le vecteur $\vec{\nu}_{t_j}$ pour toutes les tâches, l'ensemble initial des partitions possibles P_{t_j} pour chaque tâche ainsi que leur ensemble TeT et les deux vecteurs ν^{max} et ν^{min} (Lignes 1-7).

Dans la phase de la formation de coalitions (Ligne 8-26), le parcours de l'espace \mathcal{S} se fait par itération et est basé sur la partition sélectionnée P_{t_i} (courante) de la tâche courante t_i à la base des valeurs de v^{\min} . Si dans P_{t_i} au moins deux agents sont présents ($\Phi(P_{t_i}) \geq 2$), une coalition c_i est formée pour la tâche t_i et ajoutée à la structure de coalitions temporaire **tempCS**. Dans le cas où ($\Phi(P_{t_i}) < 2$), la tâche t_i dans la coalition c_i est considérée comme une tâche à réaliser seule (Lignes 11-18). Notons que pour la première itération, pour les tâches dans ν , l'algorithme (Algorithm 6) prend les ensembles vides des agents pour toutes les tâches, *c.-à-d.*, on considère que toutes les tâches dans ν sont réalisées seules sans formation de coalitions.

Après avoir parcouru toutes les tâches dans ν , nous calculons leurs coalitions (si elles existent) et formons la structure de coalitions temporaire **tempCS**. L'utilité du système est aussi calculée et comparée à celle de l'étape précédente. Si **tempCS** offre une meilleure utilité, alors elle est conservée comme la meilleure solution actuelle **bestCS** (lignes 19-23).

Par la suite, l'algorithme procède à la mise à jour des indices du vecteur v^{\min} pour déterminer l'ensemble des partitions possibles à considérer à la prochaine itération (Ligne 24).

Dans l'algorithme (Algorithm 7), si la valeur v^{\min} de la première tâche dans ν est égale à la valeur v^{\max} de la même tâche, cela signifie que tout l'espace de recherche est parcouru (lignes 26-28) et que l'algorithme retourne la dernière meilleure structure de coalitions calculée. À noter qu'à partir de la deuxième itération, la valeur de v^{\min} change à chaque nouvelle itération pour explorer les partitions possibles. La valeur de v^{\max} peut également changer dans deux cas : i) soit lorsque l'ensemble des possibilités d'une tâche t_j est exploré (Lignes 4-25), ii) soit parce que la coalition sélectionnée pour une tâche t_j rend d'autres tâches impossibles à réaliser par certains agents. Dans ce dernier cas, un recalcul des coalitions possibles est effectué pour les tâches dans **Rest $_j$** . Ce qui induit une mise à jour de l'ensemble des coalitions possibles pour toutes les tâches dans **Rest $_j$** . Alors, le fait d'avoir une valeur changée dans v^{\max} peut affecter l'ensemble des partitions possibles restantes pour les tâches dans **Rest $_j$** pour les prochaines itérations selon leurs ensembles **TeT**

Algorithme 7 UpdateVectors()

Require: $\varphi \leftarrow 0, \varphi_{\text{current}} \leftarrow 0$: float

- 1: $v^{\min}(|\mathcal{V}|) \leftarrow v^{\min}(|\mathcal{V}|) + 1$;
- 2: $\text{index} \leftarrow |\mathcal{V}|$;
- 3: $i \leftarrow |\mathcal{V}|$
- 4: **while** ($\text{index} > 0$) **do**
- 5: **if** ($v^{\min}(\text{index}) \geq v^{\max}(\text{index})$) **then**
- 6: $v^{\min}(\text{index}) \leftarrow 1$;
- 7: $v^{\min}(\text{index} - 1) \leftarrow v^{\min}(\text{index} - 1) + 1$;
- 8: **if** ($i \geq \text{index} - 1$) **then**
- 9: $i \leftarrow \text{index} - 1$;
- 10: **end if**
- 11: **end if**
- 12: $\text{index} \leftarrow \text{index} - 1$;
- 13: **end while**
- 14: **while** ($i \leq |\mathcal{V}|$) **do**
- 15: $j \leftarrow i$;
- 16: **while** ($j \leq |\mathcal{V}|$) **do**
- 17: **if** ($t_i \in \text{TeT}(t_j)$) **then**
- 18: $\text{TeA}(t_i) \leftarrow \text{TeA}(t_i) \cup \text{computeTeA}(\text{TeT}(t_j))$;
- 19: **end if**
- 20: $j \leftarrow j + 1$;
- 21: **end while**
- 22: $P_{t_i} \leftarrow \text{updatePartitionSet}(P_{t_i}, \text{TeA}(t_i))$;
- 23: $v^{\max}(i) \leftarrow |P_{t_i}|$;
- 24: $i \leftarrow i + 1$;
- 25: **end while**
- 26: **if** ($(v^{\min}(1) = v^{\max}(1))$ and $(v^{\min}(|\mathcal{V}|) = v^{\max}(|\mathcal{V}|))$) **then**
- 27: $\text{end} \leftarrow \text{true}$;
- 28: **end if**

et TeA (Ligne 22).

Après avoir calculé pour chaque agent les résultats de la partition possible

courante basée sur \mathbf{v}^{\min} , dans la phase de calcul de la prochaine itération, la valeur \mathbf{v}^{\min} de la dernière tâche dans \mathbf{v} est augmentée de un (1)(Ligne 1), puis toutes les valeurs de \mathbf{v}^{\min} de toutes les tâches sont vérifiées si elles sont égales ou supérieures à leur valeur \mathbf{v}^{\max} (Lignes 5-11). Si c'est le cas, la valeur \mathbf{v}^{\min} est remise à 1 (Ligne 6). Ensuite, l'indice i de la tâche ayant la dernière valeur modifiée de \mathbf{v}^{\min} est enregistrée (Lignes 8-10). En d'autres termes, après avoir considéré la prochaine partition possible pour la dernière tâche, tous les autres indices \mathbf{v}^{\min} des tâches restantes qui ont atteint leur dernière partition possible sont réinitialisés à 1.

Après chaque itération et après avoir envisagé une autre partition possible pour une tâche t , l'ensemble des partitions possibles pour les tâches qui sont dans Rest_t peut évoluer. Pour prendre en compte l'évolution dans l'ensemble des dépendances entre les tâches, l'ensemble des partitions possibles de la tâche est calculé (Ligne 22) en tenant compte du nouvel ensemble Task exclusive Agents (TeA) (Lignes 17-19) pour les tâches restantes Rest_t .

6.4 Scénario illustratif

Pour décrire de manière concrète le fonctionnement de l'algorithme PSD, nous présentons dans cette section un scénario illustratif qui montre à travers ses itérations comment le parcours sélectif de l'espace de recherche est effectué et comment il est guidé par les dépendances entre les tâches.

Agent	Ensembles de tâches
\mathbf{a}_1	$\mathcal{T}_1 = \{t_1, t_2, t_3, t_4, t_5\}$
\mathbf{a}_2	$\mathcal{T}_2 = \{t_1, t_2, t_6\}$
\mathbf{a}_3	$\mathcal{T}_3 = \{t_3, t_7, t_8, t_9\}$
\mathbf{a}_4	$\mathcal{T}_4 = \{t_2, t_3, t_9, t_{10}, t_{11}\}$

Tab. 6.7 – Les ensembles des tâches des agents.

Soit l'ensemble d'agents $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4\}$ et l'ensemble des tâches $\mathcal{T} = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}\}$ (cf. Fig. 3.6). Nous avons alors : $\mathcal{V}_{\mathcal{A}} = \langle \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4 \rangle$

et $V_{\mathcal{T}} = \langle t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11} \rangle$. Le tableau Tab. 6.7 illustre les ensembles des tâches de tous les agents.

Les sous-ensembles permettant à chacun des agents dans \mathcal{A} d'atteindre son but sont illustrés dans le tableau Tab. 6.8 et le tableau Tab. 6.9 illustre les vecteurs d'agents de toutes les tâches dans \mathcal{T} .

Agent	Sous-ensembles de tâches
a_1	$\mathcal{T}_1^1 = \{t_1, t_3, t_5\}$ et $\mathcal{T}_1^2 = \{t_1, t_2, t_4\}$.
a_2	$\mathcal{T}_2^1 = \{t_1, t_2\}$ et $\mathcal{T}_2^2 = \{t_6\}$
a_3	$\mathcal{T}_3^1 = \{t_3, t_7, t_8\}$ et $\mathcal{T}_3^2 = \{t_9\}$
a_4	$\mathcal{T}_4^1 = \{t_3, t_{10}\}$ et $\mathcal{T}_4^2 = \{t_9\}$ et $\mathcal{T}_4^3 = \{t_2, t_{11}\}$

Tab. 6.8 – Les sous-ensembles de tâches (alternatives) permettant aux agents d'atteindre leurs buts.

Tâche	Vecteur d'agents
t_1	$\vec{v}_{t_1} = \langle a_1, a_2 \rangle$
t_2	$\vec{v}_{t_2} = \langle a_1, a_2, a_4 \rangle$
t_3	$\vec{v}_{t_3} = \langle a_1, a_3, a_4 \rangle$
t_4	$\vec{v}_{t_4} = \langle a_1 \rangle$
t_5	$\vec{v}_{t_5} = \langle a_1 \rangle$
t_6	$\vec{v}_{t_6} = \langle a_2 \rangle$
t_7	$\vec{v}_{t_7} = \langle a_3 \rangle$
t_8	$\vec{v}_{t_8} = \langle a_3 \rangle$
t_9	$\vec{v}_{t_9} = \langle a_3, a_4 \rangle$
t_{10}	$\vec{v}_{t_{10}} = \langle a_4 \rangle$
t_{11}	$\vec{v}_{t_{11}} = \langle a_4 \rangle$

Tab. 6.9 – Les *Vecteurs d'agents* des tâches dans \mathcal{T} .

Du tableau Tab. 6.9, nous déduisons le vecteur des tâches conjointes $\mathbf{v} = \langle t_1, t_2, t_3, t_9 \rangle$. Avant de procéder au déroulement de l'algorithme PSD sur l'ensemble des itérations nécessaires pour aboutir à une solution, nous rappelons que

dans cet exemple, l'ensemble de toutes les coalitions possibles est de l'ordre de 26, comme illustré dans le tableau Tab. 6.2. Cependant, comme nous l'avons expliqué dans la section Sec. 6.3, avec les concepts que nous avons introduits, nous ne considérons que les partitions possibles comme illustrées dans le tableau Tab. 6.10. Ainsi, l'ensemble des coalitions à explorer sera alors réduit à 10, comme illustré dans le tableau Tab. 6.5.

Tâche	Partitions possibles
t_1	$P_{t_1^1} = \langle 0, 0 \rangle, P_{t_1^2} = \langle 1, 1 \rangle$
t_2	$P_{t_2^1} = \langle 0, 0, 0 \rangle, P_{t_2^2} = \langle 1, 1, 0 \rangle, P_{t_2^3} = \langle 1, 0, 1 \rangle, P_{t_2^4} = \langle 0, 1, 1 \rangle, P_{t_2^5} = \langle 1, 1, 1 \rangle$
t_3	$P_{t_3^1} = \langle 0, 0, 0 \rangle, P_{t_3^2} = \langle 1, 1, 0 \rangle, P_{t_3^3} = \langle 1, 0, 1 \rangle, P_{t_3^4} = \langle 0, 1, 1 \rangle, P_{t_3^5} = \langle 1, 1, 1 \rangle$
t_9	$P_{t_9^1} = \langle 0, 0 \rangle, P_{t_9^2} = \langle 1, 1 \rangle$

Tab. 6.10 – L'ensemble réduit des partitions possibles pour les tâches dans ν .

Les ensembles de tâches exclusives de chaque tâche sont illustrés dans le tableau Tab. 6.6. Des deux tableaux Tab. 6.10 et Tab. 6.2, nous déduisons le vecteur $\nu^{\max} = \langle 2, 5, 5, 2 \rangle$. La première possibilité sur l'espace de recherche est représentée par $\nu^{\min} = \langle 1, 1, 1, 1 \rangle$ qui signifie qu'aucune coalition n'est formée pour aucune des tâches. Donc, l'algorithme PSD ne considère pas ce cas et commence directement dans sa première itération par le vecteur $\nu^{\min} = \langle 1, 1, 1, 2 \rangle$.

Dans la suite de cet exemple, nous présentons le déroulement de l'algorithme PSD à travers 7 itérations et montrons l'évolution des différents vecteurs et ensembles utilisés pour prendre compte des changements dans les dépendances entre les tâches et leurs effets.

Itération 1 : La première itération de l'algorithme PSD correspond aux deux valeurs $\nu_1^{\min} = \langle 1, 1, 1, 2 \rangle$ et $\nu_1^{\max} = \langle 2, 5, 5, 2 \rangle$. Les partitions à sélectionner sont les premières partitions pour toutes les tâches sauf pour la tâche t_9 pour laquelle sa partition 2 est sélectionnée ($\{P_{t_1^1}, P_{t_2^1}, P_{t_3^1}, P_{t_9^2}\}$). Comme $\Phi(P_{t_1^1}) = \Phi(P_{t_2^1}) = \Phi(P_{t_3^1}) = 0$, PSD considère que les tâches seront réalisées individuellement (ou pas réalisées). Par contre, comme $\Phi(P_{t_9^2}) = 2$, la tâche t_9 sera considérée comme une

tâche à réaliser conjointement entre les deux agents \mathbf{a}_3 et \mathbf{a}_4 . Ainsi, l'ensemble des partitions de cette itération est équivalent à la structure de coalitions $\mathbf{cs} = \{\mathbf{c}_{10}\}$.

Après avoir déterminé \mathbf{cs} , PSD procède au calcul de sa fonction caractéristique puis compare sa valeur avec celle ayant la meilleure valeur jusqu'à l'itération courante. Dans le calcul, l'algorithme conserve la meilleure valeur possible en considérant que les tâches dans \mathcal{T} vont être réalisées individuellement (ou pas réalisées), sauf la tâche \mathbf{t}_9 qui va être réalisée conjointement entre les agents \mathbf{a}_3 et \mathbf{a}_4 .

Ensuite, PSD procède à la mise à jour des vecteurs \mathbf{v}^{\min} et \mathbf{v}^{\max} ainsi que l'ensemble \mathbf{TeA} pour tenir compte de l'évolution des dépendances entre les tâches en fonction des indices du parcours des deux vecteurs \mathbf{v}^{\min} et \mathbf{v}^{\max} . La nouvelle valeur de \mathbf{v}^{\min} serait alors $\mathbf{v}_2^{\min} = \langle 1, 1, 2, 1 \rangle$. On constate que dans \mathbf{v}_2^{\min} la première tâche ayant son indice changé est \mathbf{t}_3 . Alors, suite à cette mise à jour qui correspond au passage à la deuxième partition de la tâche \mathbf{t}_3 , il est impératif de mettre à jour l'ensemble des partitions possibles des tâches appartenant à l'ensemble $\mathbf{Rest}(\mathbf{t}_3) = \{\mathbf{t}_9\}$. Notons que les indices des deux tâches \mathbf{t}_1 et \mathbf{t}_2 sont restés à 1. Cela veut dire que la partition sélectionnée ne va rien changer sur l'ensemble des dépendances entre les tâches restantes (c'est-à-dire, sur les ensembles \mathbf{Rest} de chacune d'elles).

Mise à jour de l'ensemble des possibilités : Suite à la mise à jour de $\mathbf{v}_2^{\min} = \langle 1, 1, 2, 1 \rangle$, il faut mettre à jour l'ensemble des partitions possibles de la tâche $\mathbf{Rest}(\mathbf{t}_3) \cap \mathbf{TeT}(\mathbf{t}_3) = \mathbf{t}_9$. Pour cela, il faut calculer $\mathbf{TeA}(\mathbf{t}_3)$. Nous avons donc les partitions pointées dans $\mathbf{v}_2^{\min} = \langle 1, 1, 2, 1 \rangle$ qui sont $\mathbf{P}_{\mathbf{t}_1}^1, \mathbf{P}_{\mathbf{t}_2}^1, \mathbf{P}_{\mathbf{t}_3}^2$ et $\mathbf{P}_{\mathbf{t}_9}^1$. On observe que la seule partition qui vérifie la condition $\Phi(\mathbf{P}) \geq 2$ est $\mathbf{P}_{\mathbf{t}_3}^2 = \langle 1, 1, 0 \rangle$; qui correspond à une réalisation de la tâche \mathbf{t}_3 par les deux agents \mathbf{a}_1 et \mathbf{a}_3 . Donc, $\mathbf{TeA}(\mathbf{t}_3) = \{\mathbf{a}_1, \mathbf{a}_3\}$. En excluant $\mathbf{TeA}(\mathbf{t}_3)$ des agents qui vont former les partitions possibles de la tâche $\mathbf{t}_9 \in \mathbf{Rest}(\mathbf{t}_3)$, nous aurons le nouvel ensemble de partitions $\mathbf{P}_{\mathbf{t}_9} = \{\mathbf{P}_{\mathbf{t}_9}^1 = \langle 0, 0 \rangle\}$ qui comporte une seule possibilité, $\mathbf{P}_{\mathbf{t}_9}^1 = \{\langle 0, 0 \rangle\}$. Cela nous mène alors à mettre à jour le vecteur $\mathbf{v}_1^{\max} = \langle 2, 5, 5, 2 \rangle$ pour qu'il soit $\mathbf{v}_2^{\max} = \langle 2, 5, 5, 1 \rangle$.

Concrètement, cela veut dire que, après la sélection de la deuxième partition de la tâche \mathbf{t}_3 et compte tenu des dépendances entre les tâches, il ne reste qu'une seule possibilité pour la tâche \mathbf{t}_9 , qui est la partition vide.

Itération 2 : Pour la deuxième itération, nous avons $\mathbf{v}_2^{\min} = \langle 1, 1, 2, 1 \rangle$ et $\mathbf{v}_2^{\max} = \langle 2, 5, 5, 1 \rangle$ avec de nouveaux ensembles de partitions possibles comme illustrés dans le tableau Tab. 6.11. Ainsi, les partitions sélectionnées durant cette itération sont alors : $\{P_{t_1^1}, P_{t_2^1}, P_{t_3^2}, P_{t_9^1}\}$.

Tâches	Partitions possibles
t_1	$P_{t_1^1} = \langle 0, 0 \rangle, P_{t_1^2} = \langle 1, 1 \rangle$
t_2	$P_{t_2^1} = \langle 0, 0, 0 \rangle, P_{t_2^2} = \langle 1, 1, 0 \rangle, P_{t_2^3} = \langle 1, 0, 1 \rangle, P_{t_2^4} = \langle 0, 1, 1 \rangle, P_{t_2^5} = \langle 1, 1, 1 \rangle$
t_3	$P_{t_3^1} = \langle 0, 0, 0 \rangle, P_{t_3^2} = \langle 1, 1, 0 \rangle, P_{t_3^3} = \langle 1, 0, 1 \rangle, P_{t_3^4} = \langle 0, 1, 1 \rangle, P_{t_3^5} = \langle 1, 1, 1 \rangle$
t_9	$P_{t_9^1} = \langle 0, 0 \rangle$

Tab. 6.11 – L'ensemble des partitions à l'itération 2.

Comme $\Phi(P_{t_1^1}) = \Phi(P_{t_2^1}) = \Phi(P_{t_9^1}) = 1$, PSD considère que les tâches seront réalisées individuellement. Par contre, comme $\Phi(P_{t_3^2}) = 2$, la tâche t_3 sera réalisée conjointement entre deux agents (α_1 et α_3). Alors, l'ensemble des partitions de cette itération est équivalent à la structure de coalitions suivante : $cs = \{c_{10}\}$.

Après calcul de la fonction caractéristique, l'algorithme PSD procède à la mise à jour des vecteurs \mathbf{v}^{\min} et \mathbf{v}^{\max} ainsi que des ensembles TeA . À cette étape, la tâche t_9 possède une seule possibilité dans son ensemble de partition, donc, la prochaine valeur de \mathbf{v}^{\min} sera $\mathbf{v}_3^{\min} = \langle 1, 1, 3, 1 \rangle$.

Mise à jour de l'ensemble des possibilités : Suite à la mise à jour de $\mathbf{v}_3^{\min} = \langle 1, 1, 3, 1 \rangle$, nous avons $Rest(t_3) \cap TeT(t_3) = t_9$. Pour calculer $TeA(t_3)$, nous avons les partitions correspondantes à $\mathbf{v}_3^{\min} = \langle 1, 1, 3, 1 \rangle$ qui sont $P_{t_1^1}, P_{t_2^1}, P_{t_3^3}$ et $P_{t_9^1}$ avec $P_{t_3^3} = \langle 1, 0, 1 \rangle$ qui correspond à une réalisation de la tâche t_3 par les deux agents α_1, α_4 . Donc, $TeA(t_3) = \{\alpha_1, \alpha_4\}$. En excluant $TeA(t_3)$ des agents qui vont former les partitions possibles de la tâche t_9 , nous aurons l'unique possibilité $P_{t_9^1} = \langle 0, 0 \rangle$. Cela nous mène alors à mettre à jour le vecteur $\mathbf{v}_2^{\max} = \langle 2, 5, 5, 1 \rangle$ pour qu'il soit $\mathbf{v}_3^{\max} = \langle 2, 5, 5, 1 \rangle$ (il reste inchangé).

Concrètement, cela veut dire que, après sélection de la deuxième partition de la tâche t_3 , et compte tenu des dépendances entre les tâches, il ne reste qu'une seule possibilité pour la tâche t_9 , qui est la partition vide.

Itération 3 : Pour la troisième itération, nous avons $\mathbf{v}_3^{\min} = \langle 1, 1, 3, 1 \rangle$ et $\mathbf{v}_3^{\max} = \langle 2, 5, 5, 1 \rangle$. Les partitions sélectionnées durant cette itération sont alors : $\{P_{t_1^1}, P_{t_2^1}, P_{t_3^3}, P_{t_9^1}\}$.

L'ensemble des partitions de cette itération est équivalent à la structure de coalitions $cs = \{c_7\}$. La prochaine valeur de \mathbf{v}^{\min} sera alors $\mathbf{v}_4^{\min} = \langle 1, 1, 4, 1 \rangle$.

Mise à jour de l'ensemble des possibilités : Suite à la mise à jour de $\mathbf{v}_4^{\min} = \langle 1, 1, 4, 1 \rangle$, nous avons $\text{Rest}(t_3) \cap \text{TeT}(t_3) = t_9$. Pour calculer $\text{TeA}(t_3)$, nous avons la partition correspondante à $\mathbf{v}_4^{\min} = \langle 1, 1, 4, 1 \rangle$ est $P_{t_3^4} = \langle 1, 1, 1 \rangle$ qui correspond à une réalisation de la tâche t_3 par les trois agents $\mathbf{a}_1, \mathbf{a}_3, \mathbf{a}_4$. Donc, $\text{TeA}(t_3) = \{\mathbf{a}_1, \mathbf{a}_3, \mathbf{a}_4\}$. D'où l'unique possibilité pour t_9 , qui est $P_{t_9^1} = \langle 0, 0 \rangle$. Cela nous donne $\mathbf{v}_4^{\max} = \langle 2, 5, 5, 1 \rangle$.

Itération 4 : Pour la quatrième itération, nous avons $\mathbf{v}_4^{\min} = \langle 1, 1, 4, 1 \rangle$ et $\mathbf{v}_4^{\max} = \langle 2, 5, 5, 1 \rangle$. Les partitions sélectionnées sont alors : $\{P_{t_1^1}, P_{t_2^1}, P_{t_3^4}, P_{t_9^1}\}$.

Le déroulement de cette itération reste le même que celui de l'itération 3, et donne à la fin $cs = \{c_8\}$ et $\mathbf{v}_5^{\min} = \langle 1, 1, 5, 1 \rangle$ et $\mathbf{v}_5^{\max} = \langle 2, 5, 5, 1 \rangle$.

Itération 5 : Pour la cinquième itération, nous avons $\mathbf{v}_5^{\min} = \langle 1, 1, 5, 1 \rangle$ et $\mathbf{v}_5^{\max} = \langle 2, 5, 5, 1 \rangle$. Les partitions sélectionnées sont alors : $\{P_{t_1^1}, P_{t_2^1}, P_{t_3^5}, P_{t_9^1}\}$. La structure de coalitions correspondante est $cs = \{c_5\}$.

Mise à jour de l'ensemble des possibilités : Suite à la mise à jour de $\mathbf{v}_6^{\min} = \langle 1, 2, 1, 1 \rangle$, nous avons $\text{Rest}(t_2) \cap \text{TeT}(t_2) = \{t_3, t_9\}$. Pour calculer $\text{TeA}(t_2)$, nous avons la partition correspondante à \mathbf{v}_6^{\min} est $P_{t_2^2} = \langle 1, 1, 0 \rangle$ qui correspond à une réalisation de la tâche t_2 par les deux agents $\mathbf{a}_1, \mathbf{a}_2$. Donc, $\text{TeA}(t_2) = \{\mathbf{a}_1, \mathbf{a}_2\}$. Alors, les possibilités pour la tâche t_3 sont $P_{t_3} = \{P_{t_3^1} = \langle 0, 0 \rangle, P_{t_3^1} = \langle 1, 1 \rangle\}$ et l'unique possibilité pour t_9 est $P_{t_9^1} = \langle 0, 0 \rangle$. Cela nous donne $\mathbf{v}_6^{\max} = \langle 2, 5, 2, 1 \rangle$.

Itération 6 : Pour la sixième itération nous avons $\mathbf{v}_6^{\min} = \langle 1, 2, 1, 1 \rangle$ et $\mathbf{v}_6^{\max} = \langle 2, 5, 2, 1 \rangle$. Parmi les partitions possibles (*cf.* Tab. 6.12), les partitions sélectionnées suivant \mathbf{v}_6^{\min} sont : $\{P_{t_1^1}, P_{t_2^2}, P_{t_3^1}, P_{t_9^1}\}$.

Tâche	Partitions possibles
t_1	$P_{t_1^1} = \langle 0, 0 \rangle, P_{t_1^2} = \langle 1, 1 \rangle$
t_2	$P_{t_2^1} = \langle 0, 0, 0 \rangle, P_{t_2^2} = \langle 1, 1, 0 \rangle, P_{t_2^3} = \langle 1, 0, 1 \rangle, P_{t_2^4} = \langle 1, 1, 1 \rangle$
t_3	$P_{t_3^1} = \langle 0, 0 \rangle, P_{t_3^2} = \langle 1, 1 \rangle$
t_9	$P_{t_9^1} = \langle 0, 0 \rangle$

Tab. 6.12 – L'ensemble des partitions à l'itération 6.

L'ensemble des partitions de cette itération est équivalent à la structure de coalitions suivante : $cs = \{c_6\}$. La prochaine valeur de v_7^{\min} serait alors $v_7^{\min} = \langle 1, 2, 2, 1 \rangle$.

Mise à jour de l'ensemble des possibilités : Suite à la mise à jour de $v_7^{\min} = \langle 1, 2, 2, 1 \rangle$, nous avons $\text{Rest}(t_2) \cap \text{TeT}(t_2) = \{t_3, t_9\}$. Pour calculer $\text{TeA}(t_2)$, nous avons la partition correspondante à v_7^{\min} est $P_{t_2^2} = \langle 1, 1, 0 \rangle$ qui correspond à une réalisation de la tâche t_2 par les deux agents a_1, a_2 . Donc, $\text{TeA}(t_2) = \{a_1, a_2\}$. D'où alors, les possibilités pour la tâche t_3 sont $P_{t_3} = \{P_{t_3^1} = \langle 0, 0 \rangle, P_{t_3^2} = \langle 1, 1 \rangle\}$. Compte tenu de la partition à sélectionner pour t_3 , qui est $P_{t_3^2}$, l'unique possibilité restante pour t_9 serait alors $P_{t_9^1} = \langle 0, 0 \rangle$. Cela nous donne $v_7^{\max} = \langle 2, 5, 2, 1 \rangle$.

Les valeurs du déroulement complet de ce scénario sont illustrées dans les tableaux Tab. 6.13, Tab. 6.14 et Tab. 6.15

V_T	t_1	t_2	t_3	t_9	cs
v_1^{\max}	2	5	5	2	\emptyset
v_1^{\min}	1	1	1	1	
v_2^{\max}	2	5	5	2	$\{c_{10}\}$
v_2^{\min}	1	1	1	2	
v_3^{\max}	2	5	5	1	$\{c_7\}$
v_3^{\min}	1	1	2	1	
v_4^{\max}	2	5	5	1	$\{c_8\}$
v_4^{\min}	1	1	3	1	
v_5^{\max}	2	5	5	1	$\{c_5\}$
v_5^{\min}	1	1	4	1	
v_6^{\max}	2	5	5	1	$\{c_6\}$
v_6^{\min}	1	1	5	1	
v_7^{\max}	2	5	2	2	$\{c_3\}$
v_7^{\min}	1	2	1	1	
v_8^{\max}	2	5	2	2	$\{c_3, c_{10}\}$
v_8^{\min}	1	2	1	2	
v_9^{\max}	2	5	2	1	$\{c_3, c_7\}$
v_9^{\min}	1	2	2	1	
v_{10}^{\max}	2	5	1	1	$\{c_4\}$
v_{10}^{\min}	1	3	1	1	
v_{11}^{\max}	2	5	2	1	$\{c_5\}$
v_{11}^{\min}	1	4	1	1	
v_{12}^{\max}	2	5	2	1	$\{c_5, c_7\}$
v_{12}^{\min}	1	4	2	1	
v_{13}^{\max}	2	5	1	1	$\{c_2\}$
v_{13}^{\min}	1	5	1	1	
v_{14}^{\max}	2	5	5	2	$\{c_1\}$
v_{14}^{\min}	2	1	1	1	
v_{15}^{\max}	2	5	5	2	$\{c_1, c_{10}\}$
v_{15}^{\min}	2	1	1	2	
v_{16}^{\max}	2	5	5	1	$\{c_1, c_7\}$
v_{16}^{\min}	2	1	2	1	
v_{17}^{\max}	2	5	5	1	$\{c_1, c_8\}$
v_{17}^{\min}	2	1	3	1	
v_{18}^{\max}	2	5	5	1	$\{c_1, c_5\}$
v_{18}^{\min}	2	1	4	1	
v_{19}^{\max}	2	5	5	1	$\{c_1, c_6\}$
v_{19}^{\min}	2	1	5	1	
v_{20}^{\max}	2	5	2	2	$\{c_1, c_3\}$
v_{20}^{\min}	2	2	1	1	
v_{21}^{\max}	2	5	2	2	$\{c_1, c_3, c_{10}\}$
v_{21}^{\min}	2	2	1	2	
v_{22}^{\max}	2	5	2	1	$\{c_1, c_3, c_9\}$
v_{22}^{\min}	2	2	2	1	
v_{23}^{\max}	2	5	1	1	$\{c_1, c_4\}$
v_{23}^{\min}	2	3	1	1	
v_{24}^{\max}	2	5	2	1	$\{c_1, c_5\}$
v_{24}^{\min}	2	4	1	1	
v_{25}^{\max}	2	5	2	1	$\{c_1, c_5, c_7\}$
v_{25}^{\min}	2	4	2	1	
v_{26}^{\max}	2	5	1	1	$\{c_1, c_2\}$
v_{26}^{\min}	2	5	1	1	

Tab. 6.15 – Les structures de coalitions parcourues suivant l'évolution des vecteurs v^{\max} et v^{\min} .

6.5 Évaluation expérimentale

Pour l'ensemble des expériences que nous avons mené, chaque agent possède plusieurs alternatives (un plan) composées de plusieurs tâches interdépendantes. Nous avons utilisé pour les simulations une machine Intel i7 (4 GHZ) avec 4GB de RAM. Pour obtenir des valeurs moyennes, nous avons répété 30 fois chaque simulation. Nous avons effectué nos simulations en utilisant le langage JAVA et l'environnement de développement Eclipse et des populations d'agents avec des fonctions d'utilités non linéaires. Dans le système que nous avons développé, les agents transmettent leurs plans à un agent central qui s'occupe de parcourir l'espace de recherche formé par l'ensemble des structures de coalitions possibles des agents puis leur communique le résultat sur la structure de coalitions optimale. L'agent central se charge également de générer des temps de calculs des fonctions d'utilité des agents d'une manière aléatoire suivant une loi uniforme.

Rappelons que l'algorithme PSD consiste à parcourir l'espace de recherche \mathcal{S} . En tenant compte des externalités à chaque itération, la taille de \mathcal{S} peut considérablement varier. En effet, une mise à jour des ensembles de tâches possibles ainsi que des groupes d'agents qui peuvent les réaliser est effectuée (si nécessaire) en fonction de l'évolution dans l'ensemble des externalités lors de la formation de coalitions. Une itération dans PSD correspond donc à une nouvelle structure de coalitions possible en actualisant à la fois l'ensemble de tâches et les groupes d'agents qui peuvent former une coalition. Par conséquent, il faut calculer les fonctions d'utilité des agents à chaque nouvelle itération ainsi que les nouvelles coalitions possibles en éliminant les tâches conflictuelles. La mise à jour régulière des ensembles de tâches et groupes d'agents engendre alors un temps de calcul à ne pas négliger dans l'évaluation de l'algorithme PSD. Ce temps est nécessaire pour recalculer les structures de coalitions possibles à chaque itération si l'ensemble des externalités change d'une manière qui affecte l'ensemble des tâches possibles ainsi que celui des groupes d'agents. Cependant, ces ensembles de structures de coalitions peuvent être de taille réduite à cause de la réduction des ensembles de tâches possibles ainsi que des groupes d'agents.

Il est donc clair pour que notre algorithme soit efficace, le nombre d'itérations pour parcourir \mathcal{S} doit être suffisamment inférieur au nombre d'itérations pour le parcourir d'une manière exhaustive. Cependant, nos expérimentations ont montré que la différence entre le nombre d'itérations nécessaires pour faire un parcours exhaustif est beaucoup plus élevé que le nombre d'itérations parcourues par l'algorithme PSD. À titre d'exemple, dans une première étude, nous avons étudié le nombre de structures de coalitions parcourues (nombre d'itérations) en fonction du nombre de dépendances existantes³ entre les tâches. Le tableau Tab. 6.16 illustre

3. Le nombre de dépendances est la somme de toutes les dépendances existantes entre toutes

Tâches	Dépendances	Itérations avec la recherche exhaustive	Itérations avec PSD
3	4	125	30
4	8	625	50
5	16	3125	54
6	16	15625	266
7	26	78125	286
8	36	390625	338
9	50	1953125	358
10	64	9765625	410
11	64	48828125	2046
12	82	244140625	2098
13	100	1220703125	2214
14	122	1808548329	2266
15	144	30517578125	2382
16	170	152587890625	2434
17	198	762939453125	2486
18	230	3814697265625	2506
19	264	19073486328125	2526
20	282	95367431640625	3058
21	318	476837158203125	3110
22	340	2384185791015625	3594
23	380	11920928955078125	3646
24	384	59604644775390625	9426
25	390	298023223876953125	15206
26	428	1490116119384765625	15642
27	436	7450580596923828125	29982

Tab. 6.16 – Le nombre d’itérations pour parcourir l’espace de recherche \mathcal{S} dans les deux cas : parcours exhaustif et parcours par PSD avec 3 agents ayant le même plan, suivant le nombre de tâches et des dépendances.

les résultats avec 3 agents ayant le même plan. On remarque que plus le nombre de dépendances et de tâches augmentent plus le nombre d’itérations augmente d’une manière très significative dans le cas d’un parcours exhaustif de \mathcal{S} par rapport au parcours par l’algorithme PSD.

les tâches.

Notons que pour PSD, il est possible d'utiliser tout autre algorithme de partitionnement pour calculer les structures de coalitions possibles à chaque itération. Dans nos simulations, nous utilisons une approche de partitionnement classique qui consiste à générer toutes les partitions possibles sans aucune optimisation. Cela bien évidemment pénalise grandement notre algorithme, mais malgré cela, nous obtenons des résultats meilleurs qu'un parcours exhaustif sur \mathcal{S} . Ces résultats seront certainement meilleurs avec l'utilisation d'algorithmes efficaces connus dans la littérature comme dans [62, 60, 63, 59].

6.5.1 Paramètres

Le nombre d'agents dans le système est choisi aléatoirement sur l'intervalle [3, 8]. Pour chaque exécution, nous générons premièrement l'ensemble des agents puis nous affectons aléatoirement à chacun un plan sur un ensemble de 1000 plans générés aléatoirement avec le générateur utilisé en section 5.2.2. Chaque plan est composé d'un nombre de tâches dans l'intervalle [3, 27]. Pour générer le temps de calcul des fonctions d'utilité des agents ($\mathbf{temps}(\mathbf{u}_i)$), nous générons des valeurs (en millisecondes) de manière aléatoire suivant une loi uniforme sur l'intervalle [50ms, 500ms]. Nous avons effectué une évaluation comparative de l'algorithme PSD avec un parcours exhaustif de l'espace de recherche. Le choix d'une comparaison avec un parcours exhaustif se justifie par le fait que notre algorithme met à jour à chaque itération deux ensembles qui sont : l'ensemble des tâches possibles et l'ensemble des agents qui peuvent les réaliser. Il en résulte qu'il est possible d'utiliser n'importe quel autre algorithme existant dans la littérature pour générer les structures de coalitions possibles en fonction des nouveaux ensembles mis à jour, à savoir, celui des tâches et des agents.

Nous entendons par évaluation de PSD l'étude du gain de temps de son exécution. L'idée est de ne pas se considérer le temps de parcours de l'espace de recherche \mathcal{S} , mais plutôt la différence de temps entre un parcours exhaustif et un parcours avec PSD⁴. En se basant sur des éléments initiaux qui caractérisent l'ensemble des plans des agents, qui sont i) le nombre des agents dans le système, ii) le nombre de leurs tâches et iii) les dépendances entre les tâches, nous définissons deux indicateurs qui caractérisent les plans dans chaque simulation : *Poids additif des agents* (\mathcal{W}_A), *Distance de dépendances* ($\Delta_{>}$) et un indicateur *Performance d'itérations* (I) qui concerne les itérations de parcours de l'espace de recherche.

4. En effet, l'objectif avec PSD est de réduire le nombre de structures de coalitions explorées sur un espace de recherche. Le calcul de cet espace de recherche peut être effectué par un autre algorithme de la littérature.

6.5.1.1 Poids additif des agents

L'indicateur *Poids additif des agents* mesure le degré d'implication des agents sur l'ensemble des tâches dans $\vec{\nu}$. Il est calculé en fonction de toutes les tâches dans $\vec{\nu}$ et du nombre d'agents qui peuvent former des coalitions pour chaque tâche dans $\vec{\nu}$:

$$\mathcal{W}_A = \sum_{j=1}^{|\vec{\nu}|} (|\vec{\nu}_{t_j}|) \quad (6.1)$$

6.5.1.2 Distance de dépendances

L'indicateur *Distance de dépendances* (Δ_{\succ}) donne une mesure sur l'ensemble des dépendances qui caractérisent un ensemble de plans, en fonction du nombre des agents impliqués dans le processus de formation de coalitions et les ensembles TeT des tâches concernées par la formation de coalitions (les tâches dans $\vec{\nu}$) :

$$\Delta_{\succ} = \sum_{j=1}^{|\vec{\nu}|} (|\vec{\nu}_{t_j}| * \text{TeT}_j) \quad (6.2)$$

6.5.1.3 Performance d'itérations

L'indicateur *Performance d'itérations* (I) représente la performance enregistrée par l'algorithme PSD, exprimée par le rapport entre le nombre d'itérations I_0 nécessaires pour un parcours exhaustif de \mathcal{S} et le nombre d'itérations I_f nécessaires pour le parcourir avec l'algorithme PSD :

$$I = \log \left(\frac{I_0}{I_f} \right) \quad (6.3)$$

6.5.2 Gain de temps

Avec le temps de calcul de chaque fonction d'utilité des agents ($\text{temps}(u_i)$), il est possible d'estimer le temps temps_0 nécessaire pour faire un parcours exhaustif de l'espace de recherche :

$$\text{temps}_0 = \sum_{j=1}^{I_0} \left(\sum_{a_i \in \Lambda(\text{cs}_j)} \text{temps}(u_i) \right) \quad (6.4)$$

cs_j est la structure de coalitions à chaque itération j et $\text{temps}(u_i)$ est le temps de calcul de la fonction d'utilité d'un agent $a_i \in \Lambda(\text{cs}_j)$.

Notons par \mathbf{temps}_f le temps de parcours des itérations avec l'algorithme PSD :

$$\mathbf{temps}_f = \sum_{j=1}^{I_f} \left(\sum_{\alpha_i \in \Lambda(cs_j)} (\mathbf{temps}(u_i) + \mathbf{temps}_j^*) \right) \quad (6.5)$$

avec \mathbf{temps}_j^* le temps nécessaire pour mettre à jour l'ensemble des dépendances à chaque itération j et de calculer les partitions nécessaires.

Maintenant que nous pouvons calculer le temps nécessaire pour parcourir l'espace de recherche initial et le temps pour calculer la structure de coalitions optimale avec l'algorithme PSD en prenant compte le temps de mise à jour de l'ensemble des externalités, nous définissons alors la métrique $\Delta_{\mathbf{temps}}$ avec :

$$\Delta_{\mathbf{temps}} = \mathbf{temps}_0 - \mathbf{temps}_f \quad (6.6)$$

$\Delta_{\mathbf{temps}}$ représente alors le gain de temps obtenu en appliquant l'algorithme PSD par rapport au temps nécessaire pour effectuer un parcours exhaustif de \mathcal{S} .

6.5.3 Résultats des expérimentations

Dans cette section, nous nous intéressons principalement à l'évolution de $\Delta_{\mathbf{temps}}$ suivant la distance de dépendances (Δ_{\succ}) et la *Performance d'itérations* (I). Nous présentons également l'évolution des différents indicateurs sur les plans des agents ($I, \Delta_{\succ}, \mathcal{W}_A$) et la relation qui existe entre eux. Mais avant de présenter les résultats que nous avons obtenus concernant ces deux évaluations, nous présentons les graphiques sur les figures Fig. 6.4, Fig. 6.5 , Fig. 6.7 et Fig. 6.6 qui illustrent la configuration de nos expérimentations. En effet, les deux figures Fig. 6.4 et Fig. 6.5 illustrent la relation entre Δ_{\succ} et le nombre d'agents présents dans le système et le nombre de tâches respectivement, et les figures Fig. 6.7 et Fig. 6.6 illustrent la relation entre \mathcal{W}_A et le nombre d'agents et celui des tâches respectivement.

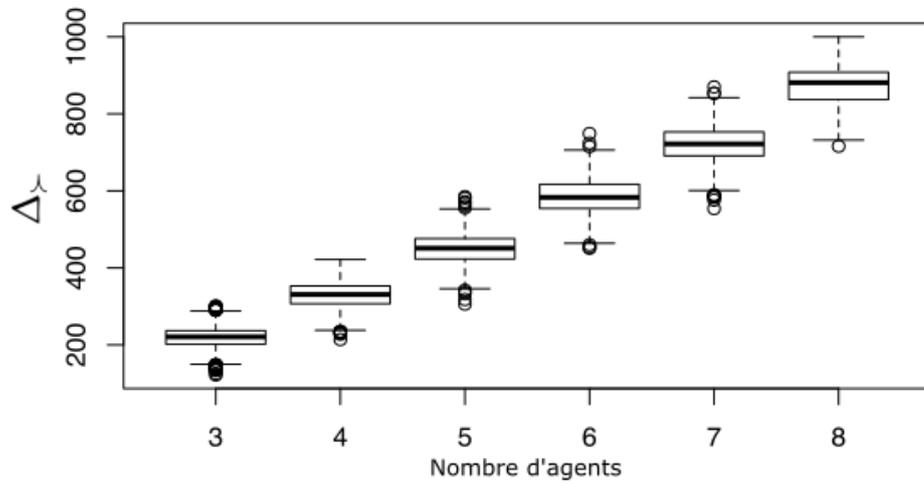


Fig. 6.4 – La distance de dépendances ($\Delta_{>}$) suivant le nombre d'agents.

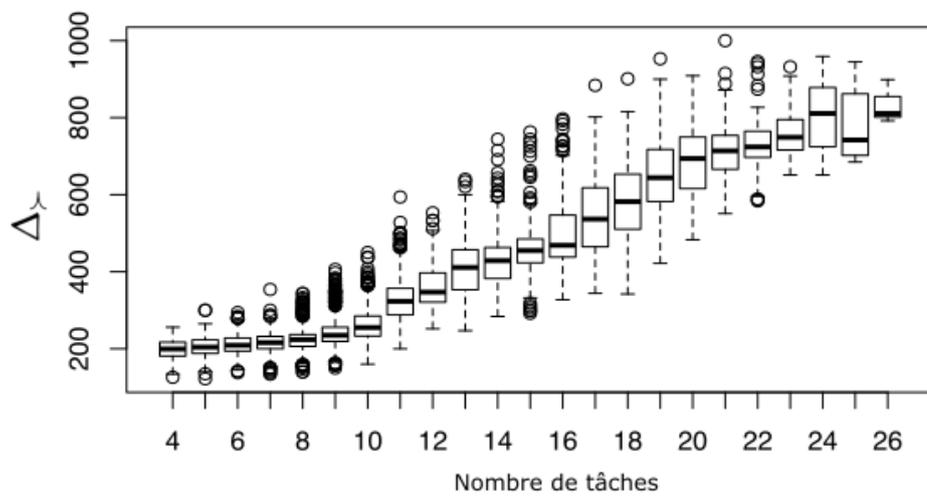


Fig. 6.5 – La distance de dépendances ($\Delta_{>}$) suivant le nombre de tâches.

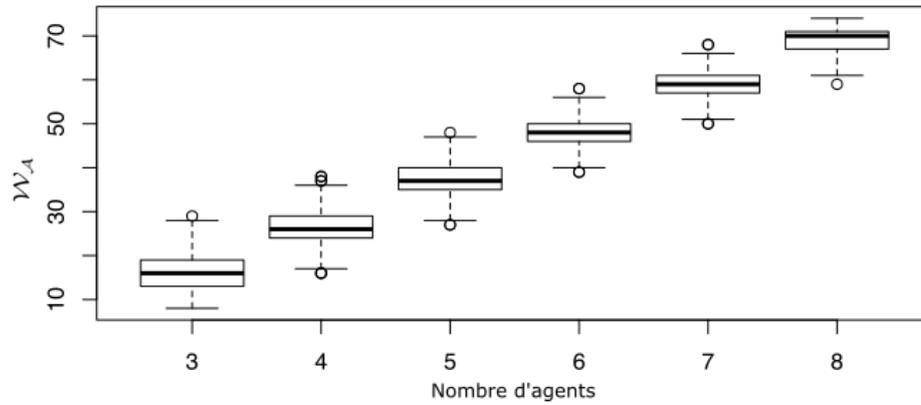


Fig. 6.6 – Le poids additif des agents (\mathcal{W}_A) suivant le nombre d'agents.

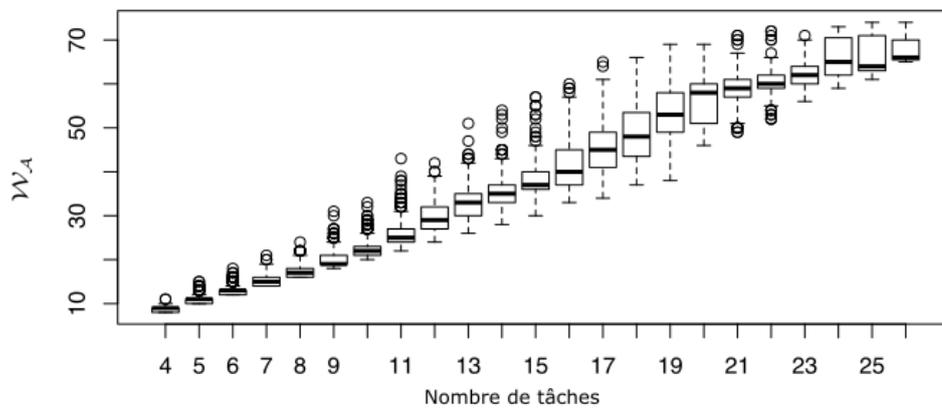


Fig. 6.7 – Le poids additif des agents (\mathcal{W}_A) suivant le nombre de tâches.

Dans la première partie des expérimentations, nous avons d'abord analysé l'évolution de Δ_{temps} suivant la distance de dépendances (Δ_{\succ}). La figure Fig. 6.8 illustre les résultats obtenus. Nous remarquons que plus Δ_{\succ} augmente plus Δ_{temps} augmente. On comprend donc que l'algorithme PSD est particulièrement efficace dans

des situations où il existe un grand nombre de dépendances entre les tâches des agents. En effet, dans des situations où nous avons des plans avec des tâches ayant beaucoup de dépendances, lors des itérations où il y a nécessité de mettre à jour les ensembles des tâches et les groupes d'agents, l'ensemble des structures de coalitions qui restent possibles peut être très réduit à cause des externalités d'impossibilités qui apparaissent sur les coalitions. Ainsi, le nombre d'itérations pour parcourir l'espace de recherche \mathcal{S} devient aussi réduit et, en conséquence, on obtient des valeurs de Δ_{temps} élevées.

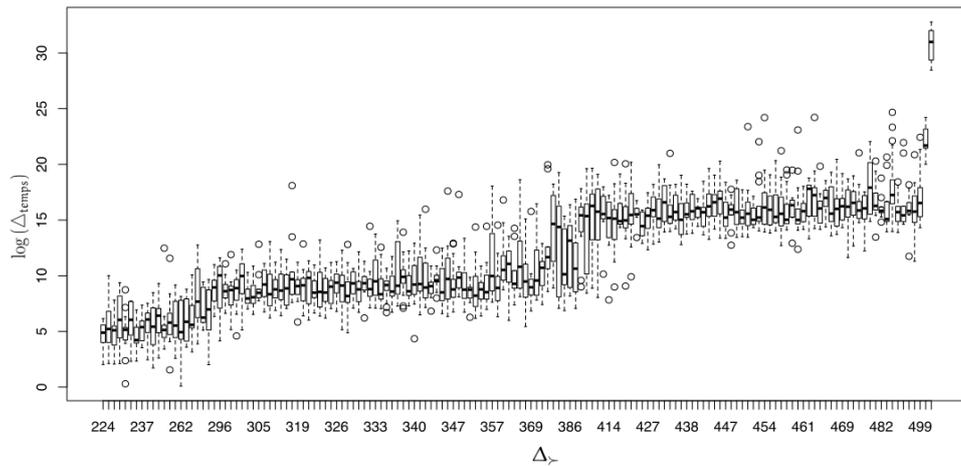


Fig. 6.8 – Évolution du gain de temps (Δ_{temps}) suivant la distance de dépendances ($\Delta_{>}$).

Dans le second cas, nous nous sommes intéressés à l'évolution de Δ_{temps} suivant I. Nous observons sur la figure Fig. 6.9 que Δ_{temps} prend une augmentation significative suivant I. Nous constatons que l'algorithme PSD enregistre de meilleurs gains de temps dans des situations où cela nécessite un nombre élevé d'itérations pour parcourir l'espace de recherche.

Nous enregistrons le même résultat pour la figure Fig. 6.10 concernant l'indicateur \mathcal{W}_A . Nous constatons alors, plus le nombre d'agents par tâche est important ainsi que le nombre total de tâches dans les plans des agents, plus le gain de temps Δ_{temps} devient important.

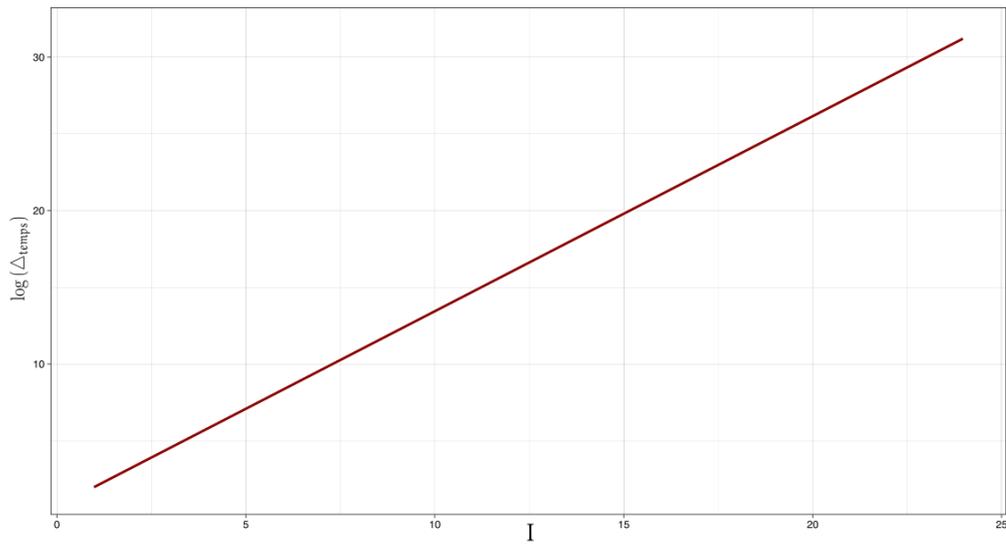


Fig. 6.9 – Évolution du gain de temps (Δ_{temps}) suivant l'indicateur performance d'itérations (I).

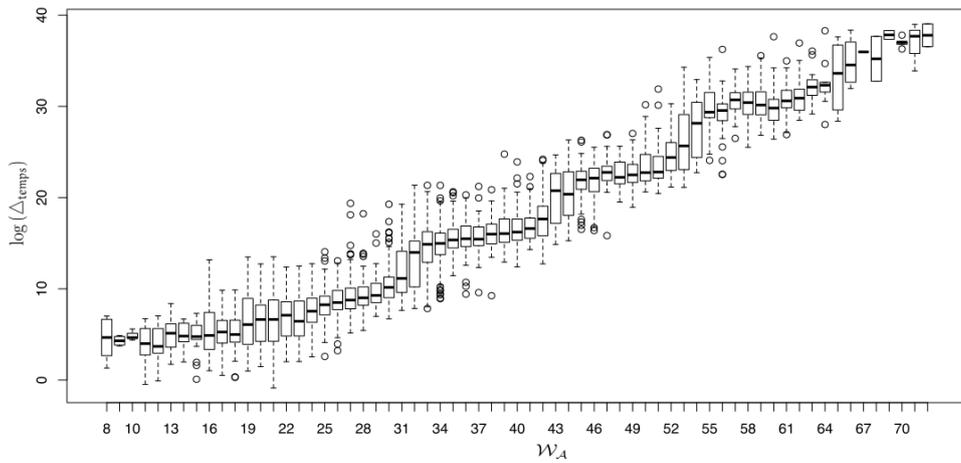


Fig. 6.10 – Évolution du gain de temps (Δ_{temps}) suivant le poids additif des agents (\mathcal{W}_A).

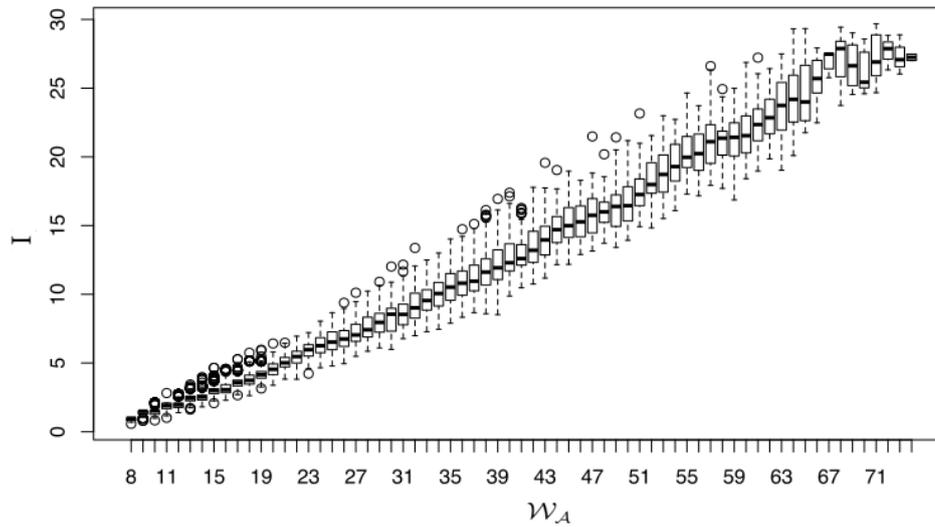


Fig. 6.11 – Évolution de l'indicateur performance d'itérations (I) suivant l'indicateur poids additif des agents (\mathcal{W}_A).

Dans la deuxième partie des expérimentations, nous avons étudié l'évolution des différents indicateurs ($I, \Delta_\succ, \mathcal{W}_A$) et la relation entre eux. Nous avons tout d'abord étudié l'évolution de I suivant \mathcal{W}_A (*cf.* Fig. 6.11). Nous remarquons une augmentation constante d'une manière linéaire de l'indicateur I avec l'augmentation de \mathcal{W}_A . Nous observons également la même évolution entre I et Δ_\succ sur la figure Fig. 6.12 et entre Δ_\succ et \mathcal{W}_A sur la figure Fig. 6.13.

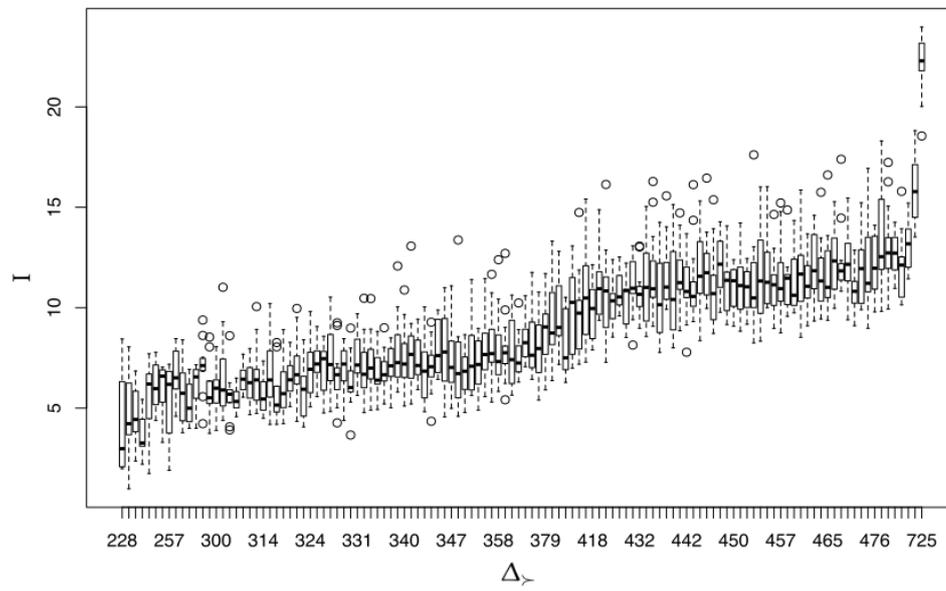


Fig. 6.12 – Évolution de l'indicateur performance d'itérations (I) suivant la distance de dépendances ($\Delta_{>}$).

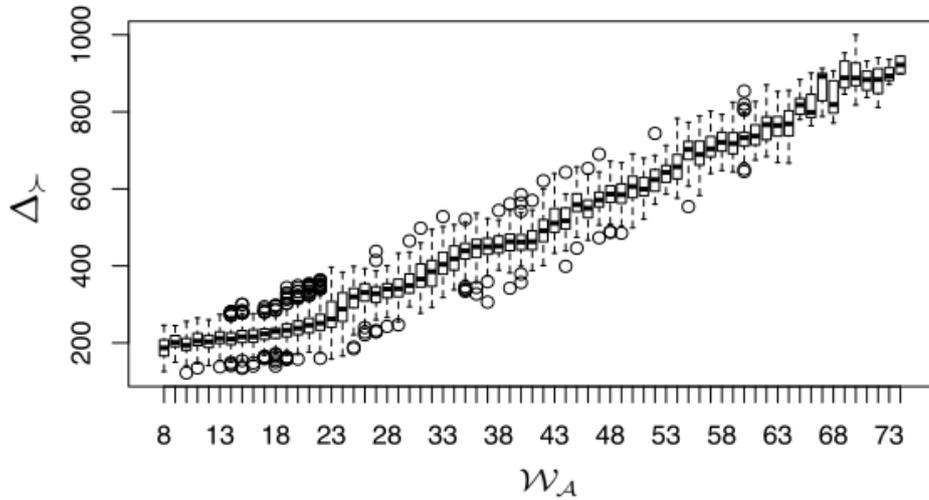


Fig. 6.13 – Évolution de la distance de dépendances (Δ_γ) suivant l'indicateur poids additif des agents (\mathcal{W}_A).

Nous concluons donc que les résultats obtenus montrent que l'algorithme PSD est efficace surtout dans des situations où d'une part, il y a un nombre important de tâches dans les plans des agents, et d'autre part, il y a un nombre important de dépendances entre les tâches (Δ_γ). PSD est aussi efficace avec un nombre important d'agents pour former des coalitions pour chaque tâche (\mathcal{W}_A); ce qui implique un nombre important d'itérations pour parcourir l'espace de recherche (I).

6.6 Conclusion du chapitre

Nous avons présenté dans ce chapitre l'algorithme PSD de formation de coalitions avec des externalités dynamiques dans sa version centralisée. Nous avons décrit le problème et l'avons formalisé en introduisant de nouveaux concepts et nouvelles définitions afin de tenir compte de ses caractéristiques particulières. Ainsi, nous avons présenté l'algorithme PSD de résolution conçu suivant une démarche de parcours sélectif de l'espace de recherche, guidé par les dépendances existantes entre les tâches. Un parcours qui se fait par itération où à chaque itération l'ensemble des dépendances entre les tâches est mis à jour pour déterminer l'ensemble des tâches restantes qui peuvent être réalisées conjointement et exclure celles qui

ne le peuvent pas à cause des dépendances conflictuelles avec les tâches déjà sélectionnées. À l'issue de ce chapitre, nous avons présenté l'évaluation expérimentale de l'algorithme **PSD** et nous avons montré son efficacité, notamment dans des situations où il y a un nombre important, à la fois, de tâches dans les plans des agents et des dépendances entre les tâches.

CONCLUSION ET PERSPECTIVES

Pour conclure ce travail de thèse, nous reprenons les questions posées et les contributions proposées. Dans la dernière section, nous évoquerons les perspectives éventuelles pour approfondir davantage les travaux déjà réalisés.

Conclusion générale et perspectives

Nous avons abordé, dans cette thèse, le problème de la coordination par formation de coalitions dans les systèmes multi-agents. Plus précisément, nous nous sommes focalisés sur le problème de la formation de coalitions avec des externalités dans les systèmes composés d'agents compétitifs. Le problème considère un ensemble d'agents possédant chacun un ensemble de tâches et un but à atteindre. Chacun dispose de plusieurs alternatives (sous-ensemble de tâches) qui lui permettent d'atteindre son but. Cependant, il existe des relations de dépendance au sein des tâches d'un même agent. Chaque tâche ayant un coût de réalisation, les agents sont intéressés par la réalisation de chaque tâche (quand il est possible) de manière conjointe pour réduire le coût global nécessaire pour atteindre chaque but. Il s'agit alors pour les agents de pouvoir proposer des coalitions, de répondre aux sollicitations et de s'engager dans les meilleures coalitions.

Nous avons, tout d'abord, décrit les concepts généralement utilisés dans le cadre des travaux sur la coordination d'agents par formation de coalitions. Des concepts relatifs aux notions d'agent, interaction multi-agents, systèmes multi-agents et quelques notions caractérisant les agents selon différents points de vue ; comme les agents coopératifs ou compétitifs. Par ailleurs, nous avons présenté quelques formes d'interaction entre agents pour assurer la coordination, la coopération ou

la négociation d'agents. Toutes ces notions fondamentales sont nécessaires pour la compréhension des problématiques relatives aux systèmes multi-agents, en général, et aux systèmes se composant d'agents compétitifs en particulier.

Nous avons présenté des travaux sur la formation de coalitions d'agents en détaillant les notions fondamentales et les problématiques existantes. Nous nous sommes intéressés aux notions théoriques où nous avons considéré le cas où les agents sont coopératifs et le cas où les agents sont compétitifs. Enfin, nous avons présenté quelques applications de la formation de coalitions.

Dans le cadre de nos travaux, nous avons introduit un nouveau problème de formation de coalitions où il existe des dépendances (que nous appelons des *externalités d'existence*) entre les coalitions possibles. Nous avons formalisé le problème comme un problème de formation de coalitions avec des externalités dynamiques, que nous avons abordé dans deux contextes : distribué et centralisé.

Dans le contexte distribué, nous avons proposé un nouveau mécanisme de formation de coalitions qui s'appuie sur des redéfinitions de quelques concepts déjà existants et l'introduction de nouveaux concepts tels que la structure de profil, le profil d'alternative, la vue d'alternative, la désirabilité d'alternative, etc. Nous avons introduit un nouvel algorithme de formation de coalitions multi-agents (CFDEA) - *Coalition Formation with Dynamic Externalities Algorithm* - qui est basé sur l'idée que les agents doivent maintenir une vue globale sur leurs alternatives tout au long du processus de formation de coalitions afin de négocier sur leurs ensembles préférés de coalitions. CFDEA permet à chaque agent d'exploiter les alternatives des autres agents et les dépendances entre les tâches pour générer des structures de coalitions.

Pour faciliter cela, nous avons introduit un *Protocole de Négociations Multilatérales (MNP)* qui nécessite un partage minimal des informations privées. Plus précisément, les agents partagent d'une manière totalement implicite uniquement leurs préférences à l'égard d'autres alternatives des autres agents. Au cours des négociations, les agents examinent les propositions qui comprennent des suggestions de tâches et de coalitions pour les exécuter. En fonction de leur évaluation privée et des préférences de leurs pairs, ils établissent leurs propres propositions ou en acceptent d'autres. Les agents utilisent une stratégie d'ordonnancement basée sur une heuristique : *Iterative Search & Converge (IS&C)* qui tente de minimiser la distance estimée entre l'alternative souhaitée d'un agent et les coalitions proposées par les autres. Avec cette heuristique, les agents adoptent un comportement de recherche en deux phases successives et répétitives ; à savoir une phase de recherche et une phase de convergence.

Nous avons traité le problème de la communication entre les agents par l'envoi direct de messages entre eux, via une entité qui assure le rôle de boîte aux lettres. Notre méthode impose une structure appropriée aux messages et fournit un protocole d'interaction imposant une sélection de réponses autorisées pendant

l'échange de messages. Cependant, si un agent juge qu'une proposition n'est pas intéressante, il est libre de ne pas répondre. L'évaluation de notre approche a montré que notre algorithme permet aux agents de converger vers une solution globale (plusieurs solutions individuelles disjointes) et de réduire considérablement le coût de la réalisation de leurs tâches et d'améliorer les performances du système.

Dans le contexte centralisé, nous avons présenté une version centralisée du problème de la formation de coalitions multi-agents dans un contexte à externalités dynamiques $C - \text{CFDCE}$ (*Centralized Coalition Formation with Dynamic Externalities*). Après avoir présenté une description formelle du problème, nous avons présenté une approche de résolution centralisée. Nous avons proposé un algorithme de formation de coalitions (PSD) par une démarche de **P**arcours **S**électif de l'espace de recherche, guidé par les **D**épendances existantes entre les tâches. L'exploration se fait par itération où à chaque itération l'ensemble des dépendances entre les tâches est mis à jour chaque fois qu'une tâche est sélectionnée pour la réaliser conjointement. Ainsi, l'algorithme détermine l'ensemble des tâches restantes qui peuvent être réalisées conjointement et exclut celles qui ne le peuvent pas à cause des dépendances conflictuelles avec les tâches déjà sélectionnées.

Nous avons procédé à une évaluation expérimentale de l'algorithme PSD qui a démontré son efficacité. L'algorithme PSD est dit *anytime*. C'est-à-dire, il peut retourner une solution à n'importe quel moment de son exécution et la qualité de sa solution s'améliore continuellement jusqu'à sa terminaison. Notre évaluation expérimentale de l'algorithme PSD a montré qu'il permet de calculer la solution optimale en un temps inférieur à l'approche exhaustive. En effet, le nombre d'itérations nécessaires pour calculer la solution est nettement inférieur avec PSD.

Perspectives de recherche

L'approche multi-agents pour la coordination de composants autonomes représente une solution appropriée pour appréhender la complexité des systèmes ambiants. Dans le cadre des perspectives de recherche de ces travaux, il est question de prendre en compte plusieurs facteurs relatifs aux utilisateurs du système. Il s'agira alors de proposer d'autres mécanismes capables à la fois de s'adapter au niveau d'autonomie variable octroyée à chacun de ses composants en fonction des situations et des recommandations de l'utilisateur ou de l'opérateur de contrôle du système ambiant et de supporter leurs interactions collectives.

Nous avons par ailleurs commencé à étudier la prise en compte de contraintes temporelles exprimées par chaque utilisateur sur les coalitions possibles entre agents. Il est question alors de prendre en considération dans les négociations entre agents les contraintes temporelles sur le début de la réalisation des coali-

tions et de la durée souhaitée. Un autre type de contraintes qui peut être ajouté est la contrainte sur les ressources dont disposent les agents pour une meilleure exploitation de celles-ci.

Cela pourra être considéré dans un cadre applicatif sur la mobilité urbain et la régulation du trafic. En effet, un des problèmes liés au transport dans les grandes villes est l'optimisation des déplacements. Il s'agira, notamment, pour ce modèle d'architecture de proposer un système coopératif de trafic avec des agents compétitifs. C'est-à-dire un système dans lequel tous les acteurs du transport (véhicules particuliers, transports en commun, intersections...) sont capables de coopérer en temps réel pour garantir aux utilisateurs un système de transport efficace, en particulier pour les moyens de transport privés avec un meilleur gain possible. Dans ces systèmes, l'itinéraire d'un véhicule de transport sera déterminé en fonction des personnes qui veulent se déplacer sur le réseau de transport. Les décisions sont prises de manière autonome par chaque agent et chaque entité (véhicule, contrôleur des feux de circulation) afin d'anticiper les événements pouvant mener à des congestions et d'optimiser l'exploitation des ressources pour réduire la pollution. Il est question alors de réorganiser les lignes de transport en temps réel en fonction des déplacements souhaités et exprimés par les utilisateurs qui arrivent sur le système.

BIBLIOGRAPHIE

- [1] S. Abdallah and V. Lesser. Organization-based cooperative coalition formation. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004. (IAT 2004).*, pages 162–168, Sept 2004.
- [2] Messan Agbaglah and Lars Ehlers. Overlapping coalitions, bargaining and networks. In *Proceedings of the Behavioral and Quantitative Game Theory : Conference on Future Directions*, page 49. ACM, 2010.
- [3] Samir Aknine and Philippe Caillou. Agreements without disagreements. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 3–7, 2004.
- [4] Samir Aknine and Onn Shehory. Reaching agreements for coalition formation through derivation of agents’ intentions. In *Proceedings of the European Conference on Artificial Intelligence ECAI*, page 180. IOS Press, 2006.
- [5] Bo An, Victor Lesser, and Kwang Mong Sim. Strategic agents for multi-resource negotiation. *Autonomous Agents and Multi-Agent Systems*, 23(1) :114–153, 2011.
- [6] S. Arib and S. Aknine. A plan based coalition formation model for multi-agent systems. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 365–368, Aug 2011.
- [7] Tolga Bektas. The multiple traveling salesman problem : an overview of formulations and solution procedures. *Omega*, 34(3) :209–219, 2006.
- [8] Filippo Bistaffa, Alessandro Farinelli, Georgios Chalkiadakis, and Sarvapali D. Ramchurn. A cooperative game-theoretic approach to the social ridesharing problem. *Artificial Intelligence*, 246 :86 – 117, 2017.

-
- [9] Gregory Bonnet and Catherine Tessier. Multi-agent collaboration : A satellite constellation case. In *Proceeding of the 2008 conference on STAIRS*, pages 24–35, 2008.
- [10] Roger Mailler Jiaying Shen Raphen Becker Kyle Rawlins Bryan Horling, R. Vincent and V. Lesser. Distributed sensor network for real time tracking. In *Proceedings of the fifth international conference on Autonomous agents*, pages 417–424, 2001.
- [11] S. Bussmann and J. Muller. A negotiation framework for cooperating agents. In *Proceedings of the CKBS-SIG, Dake Centre, University of Keele*, 1992.
- [12] Pieter Buzing, Adriaan ter Mors, Jeroen Valk, and Cees Witteveen. Coordinating self-interested planning agents. *Autonomous Agents and Multi-Agent Systems*, 12(2) :199–218, 2006.
- [13] McArthur D Camarata S and Steeb R :. Strategies of cooperation in distributed problemsolving. In *Proceedings of the 8th Int. Joint Conf on Artificial Intelligence*, pages 767–770, 1983.
- [14] J. Cao, H. Wang, and X. Wang. A distributed algorithm for agent coalition formation with complex tasks. In *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 127–132, June 2013.
- [15] Georgios Chalkiadakis, Edith Elkind, Evangelos Markakis, Maria Polukarov, and Nicholas R. Jennings. Cooperative games with overlapping coalitions. *Journal of Artificial Intelligence Research*, 39(1) :179–216, September 2010.
- [16] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. *Computational Aspects of Cooperative Game Theory*. Morgan & Claypool, 2011.
- [17] Georgios Chalkiadakis, Gianluigi Greco, and Evangelos Markakis. Characteristic function games with restricted agent interactions : Core-stability and coalition structures. *Artificial Intelligence*, 232 :76 – 113, 2016.
- [18] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. 2001.
- [19] Jeffrey Cox and Edmund Durfee. Efficient and distributable methods for solving the multiagent plan coordination problem. *Multiagent Grid Syst.*, 5(4) :373–408, December 2009.
- [20] Viet Dung Dang, Rajdeep K. Dash, Alex Rogers, and Nicholas R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1, AAAI’06*, pages 635–640. AAAI Press, 2006.
- [21] Viet Dung Dang and Nicholas R. Jennings. Generating coalition structures with finite bound from the optimal guarantees. In *Proceedings of the 3rd In-*

- ternational Joint Conference on Autonomous Agents and Multiagent Systems*, pages 564–571, 2004.
- [22] Dave de Jonge and Carles Sierra. Nb 3 : a multilateral negotiation algorithm for large, non-linear agreement spaces with limited time. *Autonomous Agents and Multi-Agent Systems*, 29(5) :896–942, 2015.
- [23] Edmund Durfee. Scaling up agent coordination strategies. *Computer*, 34 :39–46, 08 2001.
- [24] Edmund Durfee and Victor Lesser. Partial global planning : A coordination framework for distributed hypothesis formation. *Systems, Man and Cybernetics, IEEE Transactions on*, 21 :1167–1183, 10 1991.
- [25] Edmund H. Durfee. Distributed problem solving and planning. Technical report, Artificial Intelligence Laboratory, EECS Department, University of Michigan, 1999.
- [26] Lesser V. R. Durfee E. H and Corkill D. D. Coherent cooperation among communicating problem solvers. *IEE Trans Comput*, pages 1275–1291, 1987.
- [27] A.H.G. Rinnooy Kan E.L. Lawler, J.K. Lenstra and D.B Shmoys. *The traveling salesman problem*. John Wiley & Sons, 1985.
- [28] Edith Elkind, Talal Rahwan, and Nicholas Jennings. Computational coalition formation. *Multiagent Systems*, pages 329–380, 01 2013.
- [29] Alessandro Farinelli, Manuele Bicego, Filippo Bistaffa, and Sarvapali D. Ramchurn. A hierarchical clustering approach to large-scale near-optimal coalition formation with quality guarantees. *Engineering Applications of Artificial Intelligence*, 59 :170 – 185, 2017.
- [30] F. Feng, Y. Pang, G. Lodewijks, and W. Li. Agent-based negotiation and decision-making for efficient hinterland transport plan. In *Proceedings of the IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 395–400, May 2015.
- [31] Jacques Ferber. *Les Systemes Multi Agents : vers une intelligence collective*. 1995.
- [32] Zlotkin G. and Rosenschein J. S. The nature of deception in behaviour : Blocks, lies, and postal freight. In *Proceedings of the 10th Int Workshop on DAI*, 1990.
- [33] ed. G. Weiss. Multiagent systems : A modern approach to distributed artificial intelligence. *MIT Press, Cambridge, Mass.*, 1999.
- [34] T. Genin and S. Akinine. Coalition formation strategies for multiagent hedonic games. In *Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 465–472, Oct 2010.

-
- [35] Michel Grabisch. On preference representation on an ordinal scale. In *Proceedings of the 6th Eur. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty(ECSQUARU'2001)*, pages 18–28, 2001.
- [36] Gianluigi Greco and Antonella Guzzo. Constrained coalition formation on valuation structures : Formal framework, applications, and islands of tractability. *Artificial Intelligence*, 249 :19 – 46, 2017.
- [37] L Lee H S Nwana and N R Jennings. Coordination in software agent systems. *BT Technology Journal*, 1996.
- [38] Meirav Hadad, Sarit Kraus, Irith Ben-Arroyo Hartman, and Avi Rosenfeld. Group planning with time constraints. *Annals of Mathematics and Artificial Intelligence*, 69(3) :243–291, November 2013.
- [39] Bryan Horling and Victor R. Lesser. A survey of multi-agent organizational paradigms. *Knowledge Eng. Review*, 19(4) :281–316, 2004.
- [40] Karel Durkota Antonin Komenda Jan Tozicka, Jan Jakub. Multiagent planning by iterative negotiation over distributed planning graphs. In *Proceedings of the 2nd ICAPS Distributed and Multi-Agent Planning workshop (ICAPS DMAP-2014)*, 2014.
- [41] James P. Kahan and Amnon Rapoport. *Theories of coalition formation*. L.Erlbaum Associates, 1984.
- [42] Byung-Kook Kang. Optimal stopping problem with double reservation value property. *European Journal of Operational Research*, 165(3) :765–785, 2005.
- [43] Kate Larson and Tuomas Sandholm. Anytime coalition structure generation : an average case study. *Journal of Experimental and Theoretical Artificial Intelligence*, 12(1) :23–42, 2000.
- [44] Chao-Feng Lin and Shan-Li Hu. Multi-task overlapping coalition parallel formation algorithm. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '07*, pages 211 :1–211 :3, New York, NY, USA, 2007. ACM.
- [45] Nerea Luis and Daniel Borrajo. Plan merging by reuse for multi-agent planning. In *Proceedings of the 2nd ICAPS Distributed and Multi-Agent Planning workshop (ICAPS DMAP-v)*, 2014.
- [46] Georgeff M. A theory of action for multi-agent planning. In *Proceedings of the National Conf Artificial Intelligence*, 1984.
- [47] et R. M. Harstad. M. H. Rothkopf, A. Pekec. Computationally manageable combinatorial auctions. *Management Science*, 44 :1131–1147, 1998.
- [48] Efrat Manisterski, David Sarne, and Sarit Kraus. Enhancing cooperative search with concurrent interactions. *Journal of Artificial Intelligence Research*, 32(1) :1–36, 2008.

- [49] Claudia V. Goldman Mark Sims and Victor Lesser. Self-organization through bottom-up coalition formation. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 867–874, 2003.
- [50] Cheryl Martin and K. Suzanne Barber. Adaptive decision-making frameworks for dynamic multi-agent organizational change. *Autonomous Agents and Multi-Agent Systems*, 13 :391–428, 2006.
- [51] Tomasz Michalak, Dorota Marciniak, Marcin Szamotulski, Talal Rahwan, Michael Wooldridge, Peter McBurney, and Nicholas R. Jennings. A logic-based representation for coalitional games with externalities. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems : Volume 1 - Volume 1*, AAMAS '10, pages 125–132, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.
- [52] Pragnesh Jay Modi. *Distributed constraint optimization for multiagent systems*. University of Southern California, 2003.
- [53] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. Adopt : asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1) :149 – 180, 2005.
- [54] Lee L. Nwana H. S. and Jennings N. R. *Co-ordination in Multi-Agent Systems*. Lecture Notes in Computer Science, 1997.
- [55] M.J. Osborne and A. Rubinstein. A course in game theory. Technical report, MIT Press, 1994.
- [56] Gulliver P.H. Disputes and negotiations – a cross-cultural perspective. *Academic Press*, 1979.
- [57] Talal Rahwan and Nicholas R. Jennings. An algorithm for distributing coalitional value calculations among cooperating agents. *Artificial Intelligence*, 171(8) :535 – 567, 2007.
- [58] Talal Rahwan and Nicholas R. Jennings. Coalition structure generation : Dynamic programming meets anytime optimization. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 156–161, 2008.
- [59] Talal Rahwan and Nicholas R. Jennings. An improved dynamic programming algorithm for coalition structure generation. In *Proceedings of the AAMAS*, pages 1417–1420, 2008.
- [60] Talal Rahwan, Tomasz Michalak, Michael Wooldridge, and Nicholas R. Jennings. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artificial Intelligence*, 186 :95 – 122, 2012.

-
- [61] Talal Rahwan, Tomasz Michalak, Michael Wooldridge, and Nicholas R. Jennings. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artificial Intelligence*, 186(Supplement C) :95 – 122, 2012.
- [62] Talal Rahwan, Tomasz P Michalak, Michael Wooldridge, and Nicholas R Jennings. Coalition structure generation : A survey. *Artificial Intelligence*, 229 :139–174, 2015.
- [63] Talal Rahwan, Sarvapali D. Ramchurn, Nicholas R. Jennings, and Andrea Giovannucci. An anytime algorithm for optimal coalition structure generation. *J. Artif. Int. Res.*, 34(1) :521–567, April 2009.
- [64] Sarvapali D. Ramchurn, Maria Polukarov, Alessandro Farinelli, Cuong Truong, and Nicholas R. Jennings. Coalition formation with spatial and temporal constraints. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems : Volume 3*, AAMAS '10, pages 1181–1188, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.
- [65] Igor Rochlin and David Sarne. Utilizing costly coordination in multi-agent joint exploration1. *Multiagent and Grid Systems*, 10(1) :23–49, 2014.
- [66] Jerrey S. Rosenschein and Gilad Zlotkin. Designing conventions for automated negotiation. *AI Magazine*, 1994.
- [67] Stuart Russell and Peter Norvig. *Artificial Intelligence : A Modern Approach*. 1995.
- [68] M. Yokoo M. Silaghi K. Hirayama et T. Matsui S. Ueda, A. Iwasaki. Coalition structure generation based on distributed constraint optimization. *AAAI*, 2010.
- [69] Tuomas W. Sandholm and Victor R.T Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, 94(1) :99 – 137, 1997.
- [70] T W. Sandholm. *Contract types for satisficing task allocation : I theoretical results*. Stanford University, 1998.
- [71] Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Confr. On AI*,, 256-262, 1993.
- [72] Tuomas. Sandholm. *Negotiation among self-interested Computationally Limited Agents*. PhD thesis, University of Massachusetts, Amherst, <https://www.cs.cmu.edu/~sandholm/dissertation.ps>, 1996.
- [73] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohmas. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1) :209 – 238, 1999.

-
- [74] Tuomas W. Sandholm and Victor R. Lesser. Coalition formation among bounded rational agents. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 662–669, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- [75] Ip Sen Sandip Sen and Partha Sarathi Dutta. Searching for optimal coalition structures. In *Proceedings of the Fourth International Conference on Multiagent Systems*, IEEE, pages 286–292, 2000.
- [76] Onn Shehory and Sarit Kraus. Task allocation via coalition formation among autonomous agents. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 655–661, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- [77] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1) :165 – 200, 1998.
- [78] Onn Shehory and Sarit Kraus. Feasible formation of coalitions among autonomous agents in nonsuperadditive environments. *Computational Intelligence*, 15(3) :218–251, 1999.
- [79] S Shehory, O. & Kraus. Formation of overlapping coalitions for precedence ordered task- execution among autonomous agents. In *Proceedings of International Conference on Multi Agent Systems*, pages 330–337, 1996.
- [80] Yoav Shoham and Kevin Leyton-Brown. Multiagent systems : Algorithmic,game-theoretic, and logical foundations. Technical report, Cambridge University Press, 2009.
- [81] Oskar Skibski, Szymon Matejczyk, Tomasz P. Michalak, Michael Wooldridge, and Makoto Yokoo. k-coitional cooperative games. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, AAMAS '16, pages 177–185, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.
- [82] Youcef Sklab, Samir Aknine, Onn Shehory, and Abdelkamel Tari. Coalition formation with dynamically changing externalities. *Engineering Applications of Artificial Intelligence*, 91, 05 2020.
- [83] Reid G. Smith. The contract net protocol : High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 1980.
- [84] Israel Sofer, David Sarne, and Avinatan Hassidim. Negotiation in exploration-based environment. *Autonomous Agents and Multi-Agent Systems*, 30(4) :724–764, 2016.
- [85] Leen-Kiat Soh and Xin Li. An integrated multilevel learning approach to multiagentcoalition formation. In *Proceedings of the Eighteenth International JointConference on Artificial Intelligence*, pages 619–624, 2003.

-
- [86] Leen-Kiat Soh and Xin Li. A learning-based coalition formation model for multiagent systems. In *In The Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1120–1121, 2003.
- [87] Leen-Kiat Soh and Costas Tsatsoulis. Reflective negotiating agents for real-time multisensor target tracking. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 1121–1127, 2001.
- [88] Leen-Kiat Soh and Costas Tsatsoulis. Satisficing coalition formation among agents. In *In The First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1062–1063, 2002.
- [89] Peter Stone and Manuela Veloso. Multiagent systems : A survey from a machine learning perspective. *Autonomous Robotics*, 8(3), July 2000.
- [90] K Sycara. Multi-agent compromise via negotiation. *Distributed Artificial Intelligence*, 1989.
- [91] E. Elkind P. Faliszewski J. Sroka M. Wooldridge et N. R. Jennings T. Rahwan, T. P. Michalak. Constrained coalition formation. *AAAI*, 2011.
- [92] Nicholas R. Jennings Michael Wooldridge and Peter McBurney Talal Rahwan, Tomasz P. Michalak. Coalition structure generation in multi-agent systems with positive and negative externalities. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence IJCAI*, 2009.
- [93] Viet D. Dang Talal Rahwan, Sarvapali D. Ramchurn and Nicholas R. Jennings. Near-optimal anytime coalition structure generation. In *Proceedings of the IJCAI*, pages 2365–2371, 2007.
- [94] Viet Dung Dang Andrea Giovannucci and Nicholas R. Jennings Talal Rahwan, Sarvapali D. Ramchurn. Anytime optimal coalition structure generation. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, AAAI 2007*, pages 1184–1190, 2007.
- [95] Jacek Sroka Andrew Dowell Michael Wooldridge Peter McBurney Tomasz Michalak, Talal Rahwan and Nicholas R. Jennings. On representing coalitional games with externalities. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 11–20, 2009.
- [96] Hans Tonino, Andr   Bos, Mathijs de Weerdt, and Cees Witteveen. Plan coordination by revision in collective agent based systems. *Artificial Intelligence*, 142(2) :121 – 145, 2002.
- [97] Maksim Tsvetovat and Katia Sycara. Customer coalitions in the electronic marketplace. In *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 263–264, 2000.
- [98] Lovekesh Vig and Julie A. Adams. Coalition formation : From software agents to robots. *Journal of Intelligent and Robotic Systems*, 50(1) :85–118, 2007.

- [99] Jesus Virseda, Susana Fernandez, and Daniel Borrajo. Multi-agent planning with agent preferences. In *Proceedings of the 2nd ICAPS Distributed and Multi-Agent Planning workshop (ICAPS DMAP-2014)*, 2014.
- [100] T. Wang, L. Song, Z. Han, and W. Saad. Overlapping coalition formation games for emerging communication networks. *IEEE Network*, 30(5) :46–53, September 2016.
- [101] D. Yun Yeh. A dynamic programming approach to the complete set partitioning problem. *BIT*, pages 467–474, 1986.
- [102] Yu Zhang and Lynne E. Parker. Considering inter-task resource constraints in task allocation. *Autonomous Agents and Multi-Agent Systems*, 26(3) :389–419, 2013.
- [103] Yair Zick, Evangelos Markakis, and Edith Elkind. Arbitration and stability in cooperative games with overlapping coalitions. *J. Artif. Int. Res.*, 50(1) :847–884, May 2014.