



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par **l'Institut Supérieur de l'Aéronautique et de l'Espace**
Spécialité : Systèmes embarqués

Présentée et soutenue par **Grégory BONNET**
le 17 novembre 2008

Coopération au sein d'une constellation de satellites

JURY

M. Philippe Mathieu, président du jury
M. Samir Aknine, rapporteur
M. Olivier Boissier
M^{me} Marie-Claire Charmeau
M^{me} Marie-Pierre Gleizes
M. René Mandiau, rapporteur
M. Serge Rainjonneau
M^{me} Catherine Tessier, directrice de thèse

École doctorale : **Systemes**

Unité de recherche : **Équipe d'accueil ISAE-ONERA CSDV**

Directrice de thèse : **M^{me} Catherine Tessier**

Remerciements

Galilée l'a dit sous la plume de Bertolt Brecht : « il n'y a pas d'œuvre scientifique qu'un homme soit seul à pouvoir écrire ». Il n'y a pas non plus de thèse que l'on peut écrire seul. Il y a tant de personnes à remercier, à commencer par Catherine Tessier qui m'a dirigé dans ce travail, Marie-Claire Charmeau, Serge Rainjonneau, tous les membres de mon jury qui ont lu et relu mon manuscrit, le Cnes et Thales Alenia Space qui ont financé cette thèse. Je crains d'en oublier car parfois un simple mot, un conseil, un regard sitôt effacés de la mémoire suffisent à orienter tout un travail. Je voudrais donc (par ordre alphabétique afin d'éviter les jalousies) remercier tout particulièrement :

Annie, pour m'avoir toujours soutenu, pour tout.

Antoine, pour m'avoir permis de mettre en perspective ADN, vie cellulaire et intelligence artificielle, pour Planète, pour sa peinture, pour la Bête Stellaire. Keep standing broo.

Bernard, pour avoir lu mon rapport de mastère, pour lire cette thèse et pour m'avoir aidé à déployer mes ailes. Sans toi, jamais je ne serais allé jusque là.

Camille, pour raconter à qui veut l'entendre que je travaille pour la Nasa (française, entendons-nous). Puisse un jour tes mers et tes océans s'étendent aux rivages de l'espace.

Catherine, pour sa confiance, sa patience, sa gentillesse, sa rigueur, son humour, son exigence qui ont transformé mes élucubrations en un travail construit. Mille merci. Tu mérites amplement la médaille d'or des directeurs de thèse. Jamais je n'oublierais ces trois années.

Charles, Florent et Olivier, pour avoir doublé mon doctorat de la mention « coinche et belote contrée ».

Damien et Adeline, pour m'avoir permis de me vautrer avec élégance sur un Corbusier©.

Emmanuel, pour m'avoir patiemment supporté durant trois ans de vie commune, pour m'avoir initié aux mystères des processus de Markov et m'avoir fait découvrir tant d'autres choses.

Eric, pour m'avoir fait réfléchir sur ma condition d'« homo faber », d'Héphaïstos à Moravec en passant par Vaucanson et Frankenstein.

Grégory, pour la bonne idée d'avoir eu le même nom que moi et d'avoir aussi travaillé sur les satellites (à moins que ce ne soit l'inverse).

Janice, pour les salles de ventes et leurs gros marchands aux chaussettes rouges qui m'ont donné l'idée - non développée dans cette thèse - de former des coalitions par étiquetage.

Jean-Baptiste, pour avoir été là au bon moment. Merci à toi camarade.

Julien, pour être venu, lui-aussi, de la faculté, pour Man O'War, pour ce petit air d'accordéon et pour son attention (bien dissimulée) envers autrui.

Marie-Claire et Serge, qui ont su me mettre la tête dans les étoiles tout en me gardant les pieds sur terre.

Michel, pour avoir résolu avec célérité et patience tous les problèmes qu'un béotien de l'Unix comme moi était incapable de traiter. Je reste persuadé que sans toi peu de doctorants survivraient au passage obligé de l'implantation.

Nicolas, pour avoir vainement tenté de m'expliquer la mu-analyse, les LFT, le contrôle anti-windup et autres élucubrations d'automaticien (qui de toute façon ne sont pas de vrais scientifiques).

Patrice, pour m'avoir rappelé les enseignements d'Atabashka et de la notice, enseignements que je n'ai malheureusement pas respecté.

Patrice, pour ses questions sans fin et son émerveillement sans cesse renouvelé face aux mystères du monde.

Philip K. Dick, pour m'avoir fait rêver de moutons électriques.

Rémi, pour avoir eu la bonne idée de soutenir sa thèse le même jour que moi (à moins que ce ne soit, encore une fois, l'inverse).

Romain, pour m'avoir conseillé de relier mes satellites avec de grands cables afin de leur permettre de communiquer à tout moment.

Romain, pour sa bonne humeur, sa convivialité, son humour et son inefficacité crasse à la belote. Continue le théâtre, continue d'écrire.

Stanislas Lem, pour la cybériade et pour avoir pensé avant tout le monde l'intelligence en essaim.

Stéphane, pour cette semaine passée à manger du poisson et ces frigos qui interagissent émotionnellement avec leurs utilisateurs.

Sylvain, pour son humour glacé et sophistiqué, pour « philosopher avec un gourdin » et pour les discussions autour de la pertinence de la rationalité limitée, de l'utilitarisme et de l'athéisme.

Thomas, pour cette nuit arrosée, au milieu de la forêt de Bohinj, les pieds dans l'eau glacée d'un lac embrumé et cette discussion sur le darwinisme, l'origine de la vie, l'intelligence, les organisations humaines et les vins de Slovénie.

Toute l'équipe du DCSD, pour son accueil et pour son ambiance de travail si sympathique. C'est avec une pointe de regret que je vous quitte.

Yoko, pour m'avoir fait découvrir Haruki Murakami qui, lui-aussi, rêve parfois de moutons.

Musiques additionnelles : Anouar Brahem, Bobby Lapointe, Chanson Plus Bifluorée, Dead Can Dance, Émilie Simon, Gérard Manset, Hubert Félix Thiéfaine, Jacques Chelon, Julien Ribot, Korn, Marilyn Manson, Metallica, Muse, Nightwish, No One Is Innocent, Rammstein, Therion, Within Temptation.

Table des matières

Table des figures	viii
Table des définitions, protocoles et propositions	ix
Notations	xiii
Introduction	xv
1 Constellations et satellites autonomes	1
1 Qu'est-ce qu'une constellation de satellites?	1
1.1 Un ensemble de satellites interconnectés	1
1.2 Missions et objectifs	4
1.3 Détails d'une mission d'observation de la Terre	6
2 Satellites autonomes	7
2.1 Une notion relative	8
2.2 Concepts d'autonomie	9
2.3 Flots de données et structures organisationnelles	11
2.4 Plusieurs générations d'agents	12
3 Comme un unique satellite virtuel	14
3.1 Architectures pour l'autonomie des satellites	14
3.2 Planification collective en ligne	15
4 Quel est notre problème?	19
4.1 Limites des approches actuelles	19
4.2 Description de la mission	20
2 Modéliser une constellation	21
1 Des agents qui s'organisent	21
1.1 Qu'est-ce qu'un agent?	21
1.2 Qu'est-ce qu'un système multiagent?	23

1.3	S'organiser pour coopérer (ou l'inverse)	24
2	Un cadre pour les interactions	25
2.1	Taxonomie des organisations	25
2.2	Délégation et mécanismes d'interaction	30
2.3	Adaptation et auto-organisation	31
3	Une congrégation de satellites	32
3.1	Modèle d'accointances fondé sur l'environnement	33
3.2	Des tâches et des contraintes	34
3.3	Des agents intentionnels	36
3.4	Vers une connaissance commune des artefacts	40
3	Communiquer pour collaborer	43
1	Diffuser les informations	43
1.1	Taxonomie des communications	43
1.2	Protocoles épidémiques	46
1.3	Protocoles en écoute flottante	46
2	Comment les satellites communiquent-ils ?	47
2.1	Une définition en fonction des accointances	47
2.2	Déplier la relation de voisinage	48
2.3	Un protocole épidémique pour diffuser l'information	51
2.4	Propriétés des communications	52
3	Qu'est-ce que la confiance ?	55
3.1	Une définition floue	55
3.2	Classification des modèles de confiance	56
3.3	Modèles et limites	57
4	Construire une connaissance commune	59
4.1	Connaissances dérivées des accointances	59
4.2	Avoir confiance dans la dernière communication	60
4.3	Calcul de confiance	61
4	Formation décentralisée de coalitions	65
1	De la coopération	65
1.1	Protocoles de négociation	66
1.2	Prise de contrat	67
1.3	Stratégies	67
2	Qu'est-ce que la formation de coalitions ?	68
2.1	Problématiques induites par la formation de coalitions	69
2.2	Quelques approches de la formation de coalitions	71
2.3	Limites des approches actuelles	73
3	Des satellites et des coalitions	74
3.1	Définitions	75
3.2	Calculer la structure de coalitions	76

3.3	Rejoindre une coalition	76
3.4	Minimiser une coalition	78
4	Propriétés de la formation de coalitions	82
4.1	Convergence et optimalité	82
4.2	Engagements et temps de réalisation	84
4.3	Utilité de la formation de coalitions	85
5	Évaluation : simulations et mesures	87
1	Évaluation des protocoles	87
1.1	Simulation par événements discrets	87
1.2	Volume des communications	88
1.3	Tâches réalisées	88
2	Comment évaluer le passage à l'échelle ?	92
2.1	Problématique	92
2.2	Trois critères	94
2.3	Performance	95
2.4	Stabilité et capacité d'extension	96
3	Application aux constellations de satellites	97
3.1	Mesures de performance	97
3.2	Mesure de stabilité	98
3.3	Mesure de la capacité d'extension	99
3.4	Une nouvelle mesure : mesure de coopération	99
4	Évaluation pour le passage à l'échelle	100
4.1	Réactivité	100
4.2	Efficacité	103
4.3	Stabilité	104
6	Conclusion et perspectives	109
1	Contrôler une constellation de satellites	109
1.1	Du modèle d'acointances à la congrégation	110
1.2	De la congrégation aux coalitions	110
1.3	Apport de notre approche	111
2	Perspectives de travail	112
A	Propositions pour les perspectives	115
1	Analyser le modèle d'acointances	115
2	Ruptures de communication	116
3	Aléas de mission	119
	Bibliographie	123

Table des figures

1.1	Une constellation de satellites	3
1.2	Fauchée d'un satellite d'observation de la Terre	6
1.3	Trace au sol et fenêtres de réalisation	7
1.4	Une requête décomposée en tâches atomiques	8
1.5	Structures hiérarchisées ou distribuées : des approches orthogonales	12
2.1	Les différents types d'agents	22
2.2	Un système multiagent	23
2.3	Comparaison des différentes formes d'organisations	26
2.4	Organisations et mode de délégation	31
2.5	Exclusion mutuelle entre les tâches atomiques t_1 et t_2	35
2.6	Les différentes modalités d'une intention	37
2.7	Architecture de contrôle d'un agent-satellite	39
3.1	Taxonomie des mécanismes de communication	44
3.2	Représentation des deux types de communications	48
3.3	Graphe de voisinage pour l'Exemple 3.3	49
3.4	Caractéristiques des différents modèles de confiance	58
3.5	Evolution de la confiance pour l'Exemple 3.21	62
4.1	Comparaison des méthodes de génération de structures de coalitions	69
4.2	Conflit souple	79
4.3	Conflit dur	79
4.4	Critère d'expertise pour la tâche t	80
4.5	Date au plus tard de réalisation	84
4.6	Communications possibles entre deux agents en conflit	86
5.1	Nombre de messages échangés au cours d'une période de 6 heures	89
5.2	Effet de la densité sur une simulation statique	90
5.3	Nombre de tâches réalisées (avec et sans redondances)	91

5.4	Gain en termes de tâches réalisées (avec et sans redondance)	93
5.5	Réactivité de la constellation	100
5.6	Différents plans orbitaux	101
5.7	Capacité d'extension (réactivité)	102
5.8	Mesure de coopération (réactivité)	102
5.9	Efficacité de la constellation	103
5.10	Capacité d'extension (efficacité)	104
5.11	Mesure de coopération (efficacité)	104
5.12	Charge de travail de la constellation	105
5.13	Stabilité de la constellation	105
5.14	Capacité d'extension (stabilité)	106
5.15	Mesure de coopération (stabilité)	106
A.1	L'agent a_1 envoie un message à l'agent a_2	117

Table des définitions, protocoles et propositions

Définitions	
2.1	Agent 21
2.2	Organisation 25
2.3	Constellation 33
2.4	Périodicité 34
2.5	Connexité 34
2.6	Tâche atomique 34
2.7	Réaliser une tâche atomique - observation 35
2.8	Exclusion mutuelle 35
2.9	Tâche complexe 35
2.10	Réaliser une tâche complexe 36
2.11	Redondance 36
2.12	Intention 36
2.13	Plan 37
2.14	Planification : définition du problème 38
2.15	Planification : intégration des intentions 38
2.16	Planification : génération du plan 39
2.17	Connaissance 40
3.1	Communication 48
3.2	Graphe de voisinage 48
3.10	Fréquence de communication 54
3.13	Propagation d'intention 59
3.14	Propriété d'intentionnalité 59
3.15	Connaissance commune 60
3.17	Date de dernière confirmation 61
3.19	Rencontres 61
3.20	Confiance 61
3.22	Seuil de confiance maximal 62
4.1	Coalition 75
4.2	Pouvoir 75

4.3	Coalition complète	75
4.4	Coalition minimale	75
4.9	Conflit	78
4.11	Type de conflit	78
4.13	Expertise	79
4.16	Valeur de la structure de coalition	82
4.17	Valeur du plan individuel	82
5.1	Densité	89
5.2	Expérience	95
5.3	Performance	95
5.4	Événement	96
5.5	Stabilité	96
5.6	Capacité d'extension	97
5.7	Réactivité	97
5.8	Efficacité	98
5.10	Activité	98
5.11	Événement	98
5.12	Stabilité	99
5.13	Mesure de coopération	99
A.1	Complexité structurelle	115
A.2	Complexité des interactions	116
A.3	État de la constellation	117
A.5	Fusion du suivi d'état	118
A.7	Graphe de voisinage local	119
A.8	Ensemble des tâches théoriquement réalisées	121
A.9	Ensemble des tâches effectivement réalisées	121
A.10	Réputation	121

Protocoles

3.5	Diffusion des connaissances	51
3.6	Mise à jour interne	52
3.7	Mise à jour externe	52
4.7	Rejoindre une coalition	77
4.14	Résolution d'un conflit dur	81
4.15	Résolution d'un conflit souple	81
A.4	Suivi de l'état d'un agent	118

Propositions

3.8	Complexité d'une communication directe	53
3.9	Vitesse de propagation	53
3.11	Ajout d'un nouvel agent	54
3.12	Extension du modèle d'accointance	54

4.8	Stabilité de l'incitation	77
4.12	Symétrie	78
4.18	Cardinalité du treillis des structures de coalitions	82
4.19	Effet de la formation de coalitions	83
4.20	Date de réalisation au plus tard	84
4.21	Effet des communications sur la formation de coalitions	85

Notations

\mathcal{S} : constellation de satellites

\mathcal{A} : ensemble des agents d'une constellation

n : nombre d'agents de la constellation

a_i : i -ème agent de \mathcal{A}

M : ressources mémoire des agents

H : horizon de planification des agents

\mathbb{T} : ensemble de dates considérées par le système

τ : date de \mathbb{T}

τ^* : date de dernière confirmation d'une intention

Voisins : modèle d'accointance d'une constellation

p_i : durée du cycle orbital de l'agent a_i

P : période du modèle d'accointance

\mathbb{T}_P : intervalle de temps de durée P

\mathcal{V} : graphe de voisinage dérivé de Voisins

(τ_i, τ_j) : communication de a_i vers a_j émise à la date τ_i et reçue à la date τ_j

$f_j^i(\mathbb{T}_P)$: fréquence de communication entre a_i et a_j sur l'intervalle de temps \mathbb{T}_P

$M_{\tau^*}^{a_i}(I_t^{a_j})$: nombre de communications reçues par a_j entre τ^* et $rea(I_t^{a_j})$ selon a_i

$\mathcal{T}_{a_i}^\tau$: ensemble des tâches atomiques connues de l'agent a_i à la date τ

\mathcal{T} : tâche complexe

t : tâche atomique

$prio(t)$: priorité de la tâche atomique t

$prio(t)'$: priorité de la tâche atomique t après une incitation

$\mathcal{I}_{a_i}^\tau$: ensemble des intentions connues de l'agent a_i à la date τ

$I_t^{a_i}$: intention de l'agent a_i envers la tâche t

\diamond : modalité de proposition

\square : modalité d'engagement

- $\diamond\neg$: modalité de désistement faible
- $\square\neg$: modalité de désistement fort
- $rea(I_t^{a_i})$: date de réalisation de l'intention $I_t^{a_i}$
- $tel(I_t^{a_i})$: date de télédéchargement de l'image associée
- $\mathcal{K}_{a_i}^\tau$: ensemble des connaissances de l'agent a_i à la date τ
 - $K_{a_i}^\tau$: connaissance de l'agent a_i à la date τ
 - $D_{K_{a_i}^\tau}$: contenu d'une connaissance (tâche ou intention)
 - $A_{K_{a_i}^\tau}$: ensemble des agents connaissant $K_{a_i}^\tau$
 - $\tau_{K_{a_i}^\tau}$: estampille temporelle de $K_{a_i}^\tau$
- \mathcal{C}_{a_i} : structure de coalitions potentielle selon l'agent a_i
 - \mathbb{M} : treillis des structures de coalitions
 - $v(\mathcal{C})$: valeur d'une structure de coalitions
 - \mathcal{C}^* : structure de coalitions optimale
 - C : coalition
 - $v(C)$: valeur d'une coalition
 - O : ensemble des tâches atomiques objectifs d'une coalition
 - A : ensemble des membres d'une coalition
 - $|A|$: taille d'une coalition
 - \mathfrak{P} : pouvoir d'une coalition
- A^* : ensemble des agents en conflit au sein d'une coalition
 - A^+ : ensemble des agents en conflit souple
 - A^- : ensemble des agents en conflit dur
 - a^* : agent expert
 - $R(a_i)$: expertise de l'agent a_i
 - rea^* : date de réalisation au plus tôt de la tâche considérée
 - tel^* : date de télédéchargement de l'agent qui réalise la tâche à la date rea^*
- \mathcal{E} : expérience
 - $G(\mathcal{E})$: performance de l'expérience \mathcal{E}
 - $\mathcal{S}(\mathcal{E}, \mathcal{E}^\delta, \tau)$: stabilité de l'expérience \mathcal{E} subissant un événement δ à la date τ
 - $X(n, p)$: capacité d'extension en passant de n à p agents
 - $A(\mathcal{E}, \tau)$: activité de l'expérience \mathcal{E} à la date τ
 - $C(\mathcal{E}_k)$: mesure de coopération pour k agents

Introduction

L'aventure spatiale d'aujourd'hui peut être mise en perspective avec l'aventure maritime de l'Antiquité et de la fin du Moyen-Âge. Toutes deux ont nécessité le développement d'outils, de techniques de navigation, de cartographie et d'observation. Elles ont aussi nécessité la construction de machines destinées à protéger et seconder l'homme au sein d'un environnement hostile. Ces machines, devenues aujourd'hui sondes, drones ou satellites, ne sont plus habitées mais programmées ou téléguidées.

Si l'on s'intéresse aux satellites d'observation de la Terre, et plus généralement au domaine spatial, deux contraintes, qui ne sont pas en soi nouvelles mais majeures, s'imposent. D'un côté, les distances mises en jeu ne permettent pas un contrôle direct et permanent des satellites, les laissant à la merci d'un certain nombre d'aléas. D'un autre côté, les coûts de développement sont tels qu'une utilisation sous-optimale de la plateforme spatiale n'est pas rentable, ce qui arrive lorsqu'un satellite est soumis à des aléas sans recevoir des instructions adaptées. C'est pour ces deux raisons que de nombreuses recherches sont menées pour accroître l'*autonomie* des satellites.

Par autonomie nous entendons d'une part, la capacité d'un satellite à prendre des décisions sans en référer à un opérateur humain et d'autre part, sa capacité à adapter son comportement aux nouveaux événements. Les travaux sur la planification de mission en ligne tentent de répondre à cette problématique. Ces travaux se heurtent cependant à de nouveaux problèmes lorsque des *constellations de satellites* sont mises en œuvre. Il s'agit d'ensembles de satellites répartis autour de la Terre qui permettent d'envisager des activités conjointes et d'assurer une certaine robustesse fonctionnelle. Parce que les satellites sont plus nombreux, les stations sol ont plus souvent des contacts avec eux et peuvent les reprogrammer plus rapidement en cas d'aléas. De plus, l'utilisation de *liaisons intersatellites* (permettant aux satellites de communiquer entre eux) accélère la transmission des commandes venues du sol.

Si l'on désire doter d'autonomie les constellations de satellites communicants, le paradigme *multiagent* est bien adapté pour modéliser ces systèmes. Dans ce contexte, les agents doivent être capables de changer un but, réaffecter des ressources et réagir aux changements d'environnement et aux choix des autres agents, ce qui suppose que les agents doivent *coopérer*. Or la plupart des techniques de coopération et de coordination s'intéressent à des agents logiciels ou des agents sociaux qui possèdent de hautes capacités

de raisonnement et de communication. Cependant, dans le cas d'agents physiques comme des robots ou des satellites, l'environnement a une influence majeure sur les mécanismes de coopération en raison des contraintes physiques qui pèsent sur les agents. D'un côté, un agent ne peut pas toujours communiquer avec un autre ou alors ces communications sont restreintes à de brefs intervalles de temps. D'un autre côté, un agent ne peut pas toujours attendre qu'un processus de coopération ou de coordination s'achève pour agir.

Dans la littérature consacrée aux constellations de satellites, plusieurs architectures multiagents ont été proposées afin de construire de manière centralisée un plan commun. Cependant, cette centralisation implique de fortes capacités de communication, voire des liaisons permanentes entre les satellites. Dans un cadre applicatif où les communications sont intermittentes et soumises à des délais, une approche *décentralisée* est intéressante à mettre en œuvre.

C'est pourquoi nous nous proposons d'étudier dans ce manuscrit la problématique du contrôle décentralisé d'une constellation de satellites d'observation de la Terre communiquant entre eux par liaisons intersatellites. Une telle constellation peut être modélisée par un système multiagent où la dynamique des interactions, c'est-à-dire des communications, est déterministe grâce aux orbites spatiales. Nous nous fonderons sur cette seule connaissance publique pour montrer de quelle manière une connaissance commune peut être construite au sein de la constellation et comment les satellites peuvent s'organiser dynamiquement pour coopérer.

Le manuscrit est structuré comme suit :

Au Chapitre 1, nous décrirons ce que sont les constellations de satellites, les problèmes de contrôle qu'elles induisent et les approches qui ont été développées afin de les rendre autonomes. Ceci nous conduira à nous intéresser au paradigme multiagent et aux approches décentralisées.

Au Chapitre 2, nous présenterons ce paradigme ainsi que la notion d'organisation. Nous proposerons ensuite une modélisation d'une constellation de satellites à l'aide d'un système multiagent.

Au Chapitre 3, nous aborderons le problème des communications et proposerons une définition de la connaissance commune au sein de la constellation et une notion de confiance en cette connaissance fondée sur les capacités de communication des agents.

Au Chapitre 4, nous proposerons, à partir de cette connaissance commune, une méthode décentralisée de formation de coalitions afin de permettre aux agents de coopérer.

Au Chapitre 5, les performances et les capacités d'extension de nos propositions seront évaluées. Nous proposerons pour cela une formalisation de la notion d'expérience ainsi que de notions de performance, stabilité et capacité d'extension pour un système multiagent.

Au Chapitre 6, nous conclurons sur les travaux que nous avons menés et présenterons des perspectives d'études, notamment en termes de gestion des ruptures de communication.

Chapitre 1

Constellations et satellites autonomes

Afin de pouvoir situer le cadre applicatif de notre travail, nous présentons dans ce premier chapitre une description de ce que sont les satellites et, plus particulièrement, les constellations de satellites. L'exposé des missions des constellations et des besoins d'autonomie qui en résultent nous conduira à considérer différentes architectures de contrôle et de planification existant dans la littérature.

1 Qu'est-ce qu'une constellation de satellites ?

Un satellite est un robot téléopéré depuis le sol qui possède la particularité, contrairement à d'autres machines, de ne plus pouvoir être directement manipulé une fois en fonctionnement. Des contre-exemples existent, comme la désormais célèbre réparation du télescope Hubble, mais d'une manière générale une fois un satellite lancé, il n'est plus possible d'y accéder.

1.1 Un ensemble de satellites interconnectés

Un système spatial classique peut être décomposé en deux parties :

1. le *segment sol* (ou sol) représente l'ensemble des stations, des machines et des opérateurs sur terre. Sur les missions scientifiques, le sol produit généralement deux fois par semaine des plans de commandes qui sont envoyés au segment bord. Sur les missions d'observation, ces contacts sont souvent quotidiens. Malgré tout, la reprogrammation des satellites n'est pas assez rapide pour réagir à de nouveaux événements et les compétences des satellites sont souvent sous-employées. Par exemple, dans le cas des instruments optiques, 80 % des images sont inutilisables à cause des nuages ;
2. le *segment bord* (ou bord) représente le satellite, c'est-à-dire la plate-forme et la charge utile. Cependant les contraintes spatiales sont telles que les composants élec-

troniques doivent remplir certaines conditions sévères. Ainsi, des systèmes redondants doivent être mis en place en cas de panne, des éléments comme les processeurs et la mémoire de masse doivent être protégés des radiations et les communications sont bien souvent limitées. Il en résulte une puissance inférieure aux calculateurs terrestres et les algorithmes doivent être composés en fonction (les processeurs des satellites européens sont cadencés à environ 30 MHz tandis que les processeurs américains atteignent les 100 MHz).

Si les missions spatiales sont la plupart du temps menées par des satellites isolés et dépendants du segment sol, l'utilisation d'un satellite unique n'est pas toujours possible ou souhaitée. Par exemple, un seul satellite d'observation sur une orbite GEO¹ ne couvre que 30 % du globe et il peut être nécessaire de disposer d'une couverture bien plus importante comme dans le cas du GPS (**G**lobal **P**ositioning **S**ystem) qui est composé de 31 satellites sur 6 plans orbitaux quasi circulaires. De plus, il est parfois impossible d'embarquer l'ensemble des instruments nécessaires à la réalisation d'une mission sur un seul satellite, en raison de la masse et de la dimension totale du matériel. On réalise alors ces missions par l'intermédiaire d'une constellation de satellites.

Une constellation de satellites, comme représentée sur la Figure 1.1, est un ensemble de satellites répartis autour de la Terre dont les activités sont combinées. Traditionnellement, les constellations forment un ensemble de fonctions de guidage, de navigation et de contrôle (GNC) indépendantes car chaque satellite est contrôlé séparément par le segment sol.

Cependant, afin d'accroître la réactivité et les performances du système, les satellites d'une même constellation peuvent être reliés par *liaisons intersatellites* (ou ISL comme **I**nter**S**atellite **L**inks). Cela conduit à une conception plus rigoureuse et à ajouter des équipements nécessaires à l'intercommunication. Ainsi, le GNC ne peut plus être considéré comme un ensemble d'éléments indépendants et de nouvelles perturbations, celles qui peuvent affecter l'ensemble des satellites de la constellation, doivent être prises en compte. Dans ce cadre (et c'est celui sur lequel porte ce manuscrit), une constellation est une architecture distribuée qui agit comme un *unique satellite virtuel* : un ensemble de fonctions de guidage, de navigation et de contrôle (GNC) interconnectées par des capteurs distants.

Les liaisons intersatellites peuvent être de deux types :

1. *radioélectriques* : les satellites utilisent un signal hertzien pour communiquer. Chaque signal est caractérisé par une largeur de bande (plage de fréquences). Les liaisons intersatellites radioélectriques utilisent généralement une bande V (60 GHz) ;

1. Geostationary Earth Orbit : orbite circulaire située à environ 36000 kilomètres et inclinée à environ 0° par rapport à l'Équateur. Depuis la Terre, un satellite placé sur cette orbite paraît fixe. Il avance à la même vitesse que celle de la rotation de la Terre. C'est l'orbite des satellites de communication et de télévision essentiellement mais également des satellites météorologiques.

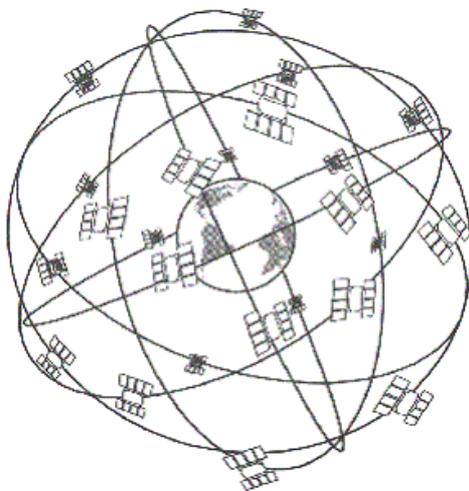


FIGURE 1.1 – Une constellation de satellites

2. *optiques* : les satellites utilisent un faisceau laser pour se transmettre de l'information². Les liaisons optiques permettent de communiquer plus d'information avec un faisceau plus étroit, ce qui réduit les interférences. En contrepartie, les satellites doivent être beaucoup plus précis dans leur synchronisation.

Bien que l'on puisse répartir les satellites à différents paliers d'altitude, il est préférable de concevoir des constellations régulières, où les satellites partagent une orbite et une inclinaison identiques par rapport à l'Équateur, afin de faciliter les communications [Wang, 1993, Werner *et al.*, 1993]. En effet, en raison de la courbure de la Terre et des risques d'interférence, les constellations régulières sont beaucoup plus efficaces pour acheminer des messages. Ces constellations sont structurées en plusieurs plans orbitaux sur 360° qui comprennent chacun un nombre identique de satellites régulièrement répartis. Elles sont caractérisées en *catégories de Walker* [Walker, 1984] et définies par un tuple $\langle T, P, F, i \rangle$ où T est le nombre total de satellites, P le nombre de plans orbitaux, F le déphasage entre deux plans orbitaux et i l'inclinaison des plans par rapport à l'Équateur.

Enfin si le nombre de satellites dans la constellation est une contrainte en termes d'architecture de contrôle et de communication, un grand nombre de satellites assure la redondance, ce qui réduit la fragilité du système, et permet de distribuer les compétences. Par conséquent, un grand nombre de satellites hétérogènes permet d'accepter un risque

2. Cette technologie peut aussi servir aux liaisons interorbites (IOL) grâce auxquelles un satellite en orbite basse peut transmettre des données avec un débit élevé (par exemple jusqu'à plusieurs centaines de mégabits par seconde) à un satellite en orbite géosynchrone. Le débit des données de la liaison descendante est beaucoup plus faible, par exemple de l'ordre de plusieurs mégabits par seconde. La portée de ce type de liaison s'étend jusqu'à 45000 kilomètres environ.

plus élevé de défaillance individuelle. De plus, si de plus petits satellites sont suffisants, cela réduit les coûts de développement.

Ainsi, une constellation de satellites est un segment bord multiple permettant de :

1. distribuer la charge utile ;
2. accroître la couverture et, par conséquent, la réactivité du système ;
3. améliorer la robustesse fonctionnelle du système spatial.

1.2 Missions et objectifs

De manière générale, les constellations actuelles sont utilisées pour trois types de missions : des missions de télécommunication, des missions d'exploration planétaire et des missions d'observation de la Terre. Ces dernières sont les plus nombreuses et ce sont elles qui nous intéresseront dans la suite de ce manuscrit. Avant de détailler ce type de missions, considérons quelques constellations qui ont été envisagées (et parfois mises en œuvre).

Télécommunications et exploration planétaire

Des constellations comme *Iridium* [Maine *et al.*, 1995], *Globalstar* [Dietrich *et al.*, 1998], *Teledesic* [Sturza, 1994] ou *Skybridge* [Sorre et Sourisse, 1997] ont été envisagées pour servir de relais dans les réseaux de télécommunication mondiaux. Si des satellites géostationnaires sont couramment utilisés en raison d'une durée de vie accrue³, ces derniers (comme nous l'avons vu précédemment) ne peuvent être en visibilité que d'une partie de la surface du globe. Utiliser un plus grand nombre de satellites répartis au sein d'une constellation en orbite LEO⁴ est alors avantageux [Wood, 2003]. Non seulement une couverture complète du globe est possible mais les capacités individuelles des satellites sont augmentées car moins dégradées par les radiations ; de plus, les temps de transfert sont réduits par rapport à une orbite GEO par l'utilisation de fréquences radio terrestres pour la communication.

Un autre type de mission concerne l'exploration planétaire. L'un des objectifs est la ceinture d'astéroïdes gravitant entre Mars et Jupiter. Cette ceinture est totalement inexplorée et doit contenir de nombreuses ressources en métal, glace ou gaz rares. L'utilisation de satellites autonomes pourrait permettre une exploration ou une exploitation. Cependant il s'agit de projets très futuristes, voire irréalistes. Par exemple, la mission *Prospecting ANTS* [Curtis *et al.*, 2003] vise à utiliser une constellation de 1000 satellites d'un kilogramme, appelés pico-satellites. La mission se déroule en deux phases. La

3. La durée d'un satellite en orbite GEO est généralement bien plus longue que celle d'un satellite en orbite LEO. En effet, bien que soumis à de plus fortes radiations, les satellites GEO ne subissent pas une usure de l'orbite aussi forte que les satellites LEO.

4. Low Earth Orbit : orbite basse située entre 350 et 1400 kilomètres environ, sous la première ceinture de Van Allen. Une orbite basse ayant une inclinaison qui lui permet un survol des régions proches des pôles est appelée orbite polaire.

première est le trajet entre un point de Lagrange terrestre où sera développé l'essaim et la ceinture d'astéroïdes. Le voyage à voile solaire durera entre 2 et 3 ans. La seconde phase est l'exploration de la ceinture et la confrontation avec des objets inconnus (lunes d'astéroïdes, blocs en mouvement, etc.)

Surveillance et observation

Les missions de surveillance terrestre nécessitent souvent l'utilisation de constellations. En raison du temps de revisite et de la précision demandée, une constellation en orbite LEO est la plus efficace. En effet, un unique satellite en orbite GEO pourrait être utilisé mais, du fait de la distance avec le sol, la dimension des instruments de détection rend l'opération coûteuse. Une constellation en orbite MEO⁵ pose les mêmes problèmes, d'ordre restreint cependant, en ce qui concerne la couverture et la précision.

La mission Fuego [Escorial *et al.*, 2003], qui a pour but de fournir aux utilisateurs un outil puissant pour la détection et le suivi des feux de forêts, est une formation de Walker de 12 satellites de 64 kilogrammes en orbite LEO. Ce principe peut s'étendre aux activités volcaniques et toute autre source de points chauds. La formation TechSat-21 [Sherwood *et al.*, 2001] est un autre exemple à plus petite échelle de ce genre de mission : 3 satellites volant en formation à une distance relative de 100 mètres à 5 kilomètres.

La constellation SAR-Lupe [Penne *et al.*, 2005] est composée de 5 satellites radar bande-X (comme les satellites TerraSAR-X [Buckreuss *et al.*, 2003], X-SAR [Evans *et al.*, 1993] et SRTM [van Zyl, 2001]) qui peuvent prendre des images précises pour une analyse fine aussi bien que des images sur une large fauchée pour une analyse globale. Les satellites sont répartis sur 3 plans orbitaux. Leur spécificité vient du fait qu'ils sont équipés de liaisons intersatellites radioélectriques bande-S afin de se communiquer des commandes venant de la station sol.

Le programme Pléiades⁶, orchestré par le CNES (Centre National d'Études Spatiales), vise à prendre la succession du programme SPOT d'observation de la Terre [Beaumet *et al.*, 2008, Bouveret, 2007]. Pléiades est une constellation de satellites d'observation de la Terre à vocation commerciale ou de service public capable de couvrir l'ensemble des besoins européens, civils et de défense. Le premier lancement devrait avoir lieu à l'horizon 2010. L'un des objectifs principaux de Pléiades est de fournir aux divers participants du programme un accès partagé aux prises de vue.

5. Medium Earth Orbit : orbite intermédiaire entre les orbites LEO et GEO située entre 10000 et 20000 kilomètres environ, entre la première et la seconde ceinture de Van Allen.

6. Nom donné à un amas ouvert d'étoiles jeunes, chaudes et brillantes dans la constellation du Taureau à environ 410 années-lumière de la Terre. Dans la mythologie grecque, les Pléiades étaient les filles d'Atlas et de Pléione et sœurs des Hyades. Au nombre de sept, elles se prénommaient Électre, Maïa, Stérope, Taygète, Mérope, Alcyone et Céléno, et, poursuivies par Orion, elles furent changées en étoiles.

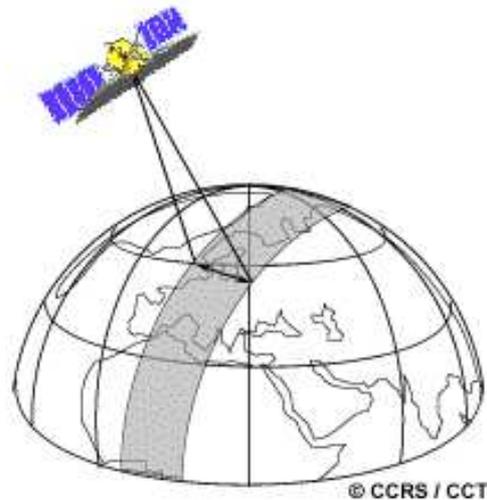


FIGURE 1.2 – Fauchée d'un satellite d'observation de la Terre

1.3 Détails d'une mission d'observation de la Terre

Que fait plus précisément un satellite d'observation de la Terre ? Il dispose de capteurs lui permettant de prendre des images du sol. Ces capteurs peuvent être de deux types :

1. *passifs*, ils enregistrent l'énergie solaire renvoyée par la surface terrestre (ultraviolet, visible, infrarouge) ou émise (infrarouge thermique, micro-ondes). Les capteurs passifs qui peuvent enregistrer plusieurs bandes spectrales sont appelées capteurs multispectraux, voire hyperspectraux lorsqu'ils ont accès à plusieurs centaines de longueurs d'ondes ;
2. *actifs*, ils fournissent leur propre rayonnement pour illuminer la zone à observer, le plus répandu des systèmes étant le radar qui peut ainsi être utilisé de jour comme de nuit.

Lorsqu'un satellite est en orbite autour de la Terre, les capteurs peuvent observer une certaine partie de la surface, appelée *fauchée*, dont la largeur varie généralement entre une dizaine et une centaine de kilomètres (Figure 1.2). Le mouvement combiné de la rotation de la Terre et de l'orbite du satellite lui permet d'observer une nouvelle région à chacun de ses passages consécutifs. Cela permet une couverture complète de la surface de la planète après un cycle orbital⁷ complet.

Chaque observation est associée à une ou plusieurs fenêtres temporelles de visibilité. Par conséquent, un satellite ne peut observer deux zones en même temps ou deux zones trop proches l'une de l'autre avec le même capteur. Il existe un type de satellite, appelée

7. On appelle cycle orbital le nombre entier de révolutions qu'un satellite doit accomplir pour se retrouver dans la même direction au-dessus d'un même point de la Terre.

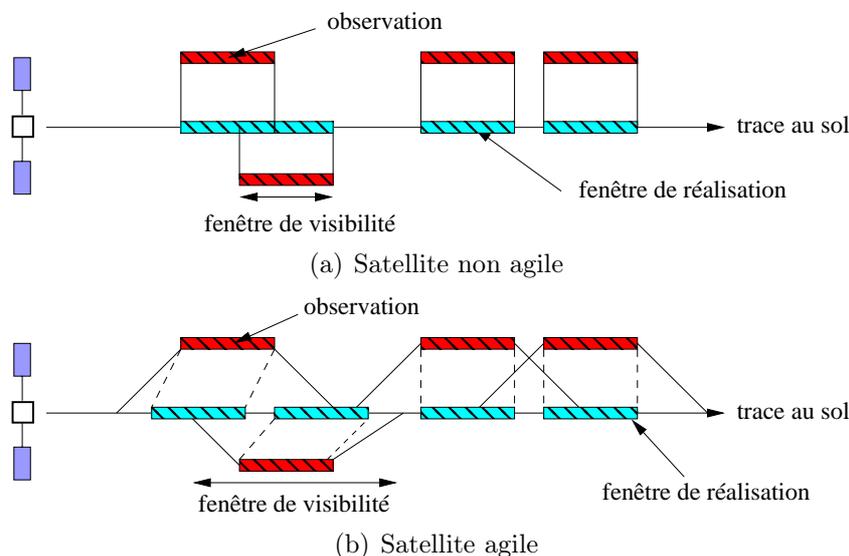


FIGURE 1.3 – Trace au sol et fenêtres de réalisation

satellite agile, capable de manœuvrer librement et rapidement selon les trois axes sur son orbite et donc d'observer des zones qui seraient inaccessibles à un satellite non agile. La Figure 1.3 représente les fenêtres de réalisation pour ces deux types de satellites.

Pendant chaque observation, le satellite enregistre l'image résultante dans sa mémoire de masse puis il la conserve jusqu'à être en visibilité d'une station au sol. À ce moment, il *téléchargera* les images ainsi que des informations de télémétrie⁸ en direction du centre de mission qui lui transmettra de nouvelles commandes sous formes de télécommandes ou de requêtes de haut niveau.

Ces requêtes correspondent à des zones au sol à observer sur lesquelles des contraintes de priorité sont définies. Cependant elles ne peuvent pas la plupart du temps être observées en une seule fois en raison de la taille de zone ou du nombre de bandes spectrales à utiliser. Ainsi les requêtes peuvent être décomposées en tâches d'observation atomiques (ou primitives). La Figure 1.4 illustre cette décomposition.

2 Satellites autonomes

L'opération d'un satellite en orbite doit prendre en considération les ressources énergétiques du segment bord, sa mémoire de masse restante, le fait que les capteurs passifs

8. Formée à partir des racines grecques *tele* (à distance) et *metron* (mesure), la télémétrie correspond au fait de journaliser des informations d'intérêt provenant d'un système vers un opérateur. L'opération inverse de la télémétrie est la télécommande.

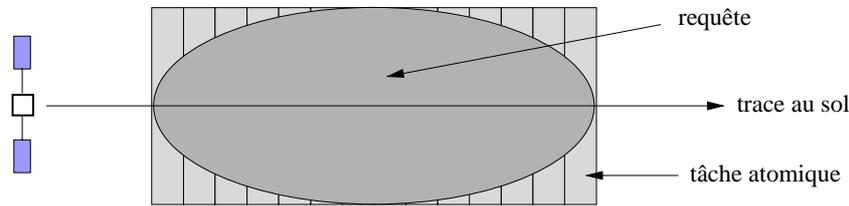


FIGURE 1.4 – Une requête décomposée en tâches atomiques

ne peuvent pas être utilisés de nuit, les incompatibilités entre diverses requêtes ainsi que leur urgence.

Cette programmation peut être faite par l'intermédiaire d'une station sol mais chaque contact avec le satellite est coûteux. Si l'on désire diminuer les opérations (et donc le coût), la programmation doit se faire sur une longue durée, ce qui la rend plus complexe, moins optimisée et peu réactive. Si l'on désire améliorer la réactivité, il faut multiplier les contacts (et donc le coût) ou déporter une partie des opérations au segment bord.

Cela est d'autant plus vrai avec une constellation de satellites. C'est pour cela que de nombreuses recherches sont menées pour accroître l'autonomie des satellites.

2.1 Une notion relative

Un satellite non autonome doit en référer au sol lorsqu'il s'agit de prendre des décisions par rapport à un événement inattendu. Dans le pire des cas, il passe en mode *Survie*⁹ et abandonne de ce fait la mission. Le satellite autonome, lui, est programmé pour prendre des décisions par lui-même. En lui permettant d'identifier et résoudre par lui-même les problèmes qui peuvent survenir au cours d'une mission, d'adapter son comportement aux nouveaux événements et en transférant la planification¹⁰ à bord, les performances et les possibilités de mission sont accrues (au prix d'une augmentation du coût de développement).

Dans la littérature consacrée à l'autonomie, en fonction de l'ensemble des compétences d'un satellite ou d'un robot, est défini le *degré d'autonomie* de ce dernier sous forme de niveaux ordonnés [Parasuraman *et al.*, 2000].

[Schetter *et al.*, 2003] utilise par exemple les catégories suivantes pour les satellites, allant de la plus autonome à la moins autonome :

9. Le mode *Survie* est le dernier mode de sauvegarde du satellite (orientation des panneaux solaires en direction du Soleil, stabilisation orbitale, tentatives de communication, économie de ressources) dont la sortie nécessite de nombreuses opérations au sol avant de pouvoir reprendre la mission.

10. Nous entendons par *planification* la génération d'activités afin de satisfaire un certain nombre de buts et nous entendons par *ordonnancement* l'association de ces activités pour satisfaire des contraintes temporelles.

Niveau 1 : le satellite est autonome dans le sens où il est capable de planifier globalement (pour lui-même et pour autrui), d'interagir avec d'autres satellites (prendre des engagements, demander des informations), de recevoir des commandes du sol ou d'un autre satellite et de les exécuter ;

Niveau 2 : le satellite est capable de planifier localement (pour lui-même uniquement) mais peut aussi interagir avec d'autres satellites, recevoir des commandes du sol ou d'un autre satellite et les exécuter ;

Niveau 3 : le satellite possède une faible autonomie en étant capable de planifier localement, de recevoir des commandes du sol et de les exécuter ;

Niveau 4 : le satellite est complètement téléopéré, seulement capable de recevoir des commandes du sol et de les exécuter ;

Le CNES utilise l'échelle d'autonomie suivante, présentée dans [Rey, 2007] :

Niveau 1 : le satellite possède une haute autonomie car il est capable de créer ou modifier un plan de tâches fourni par le sol ainsi que de prendre en compte des événements inattendus. En cas d'anomalie, le satellite est capable de se reconfigurer pour poursuivre l'opération ;

Niveau 2 : le satellite possède une autonomie moyenne en étant capable de recevoir des commandes du sol et de les exécuter sans devoir maintenir une boucle de retour avec le sol. En cas d'anomalie, le satellite abandonne l'opération en cours et passe en mode **Attente**¹¹ ;

Niveau 3 : le satellite possède une faible autonomie en n'étant capable de recevoir des commandes du sol et de les exécuter qu'en maintenant une boucle de retour avec le sol lors d'étapes spécifiques de l'opération. En cas d'anomalie, le satellite abandonne l'opération en cours et passe en mode **Survie**.

Cependant ces approches (comme toutes les approches à base de niveaux statiques) sont très restrictives car elles ne saisissent pas la différence entre *autonomie* et *automatisation*. Un satellite peut être autonome pour certaines fonctions mais laisser par exemple les calculs d'orbite aux stations sol : l'autonomie est une notion relative. C'est pour cette raison qu'il est préférable de parler de **concepts d'autonomie**, comme le font [Polle, 2002, Charneau et Bensana, 2005, Truskowski et Karlin, 2000], que d'autonomie globale.

2.2 Concepts d'autonomie

Chaque concept est indépendant des autres. Un seul d'entre eux ne suffit pas à garantir l'autonomie d'un satellite mais une combinaison, en fonction de la mission, de plusieurs

11. Le mode Attente consiste pour le satellite à se mettre en position d'attente (pointage géocentrique, mise en marche de la charge utile) et se préparer à recevoir de nouvelles télécommandes.

d'entre eux, en augmente considérablement le niveau. On peut les diviser en deux catégories :

1. les concepts liés aux commandes (par essence la planification) ;
2. les concepts liés au suivi de situation (par essence la FDIR¹² et la télémétrie).

Planification long terme : c'est la manière traditionnelle d'envoyer un plan à un satellite. L'autonomie est directement liée à la période de temps entre deux plans. Si un événement inattendu se produit, le centre de contrôle doit replanifier la tâche et communiquer les commandes au satellite. Cela fonctionne pour des missions simples et il est inefficace de planifier sur plus d'une semaine.

Génération automatique au sol : ce concept permet de mettre en place une boucle courte au sol. Les informations de télémétrie sont envoyées au centre de contrôle qui génère automatiquement les commandes et les transmet au satellite. Ce concept est suffisant dans le cas de satellites en orbite GSO¹³ car ils ont un lien permanent avec le centre de contrôle. Dans d'autres cas, la restriction des communications diminue d'autant la puissance du concept.

Planification événementielle : un plan valide est envoyé au satellite en ne se fondant plus sur le temps mais la position spatiale. Les commandes ou séquences de commandes sont associées à des événements. Le satellite observe l'environnement pour capter ces événements et déclencher la réponse adéquate. Cela peut être l'entrée dans une zone spatiale particulière ou la détection d'un niveau de ressources donné à bord. Des niveaux de priorité doivent être associés aux événements afin d'éviter les conflits. Les procédures de contrôle bord sont bien adaptées à ce type de planification.

Planification bord : les limites du concept précédent sont atteintes lorsque la mission devient complexe ou nécessite une optimisation en raison de la rareté des ressources par exemple. Ici le centre de contrôle envoie une mission de haut niveau au satellite que les planificateurs bord décomposent pour contruire un plan valide. Chaque nouvelle requête relance la planification. Un tel concept a été utilisé pour l'architecture RemoteAgent de Deep Space One [Mussettola *et al.*, 1998]. Cependant, la planification nécessite de lourds calculs qui peuvent prendre beaucoup de temps.

Planification bord réactive : réagir se fait couramment lorsque les événements inattendus déclenchent des alarmes ou provoquent des erreurs. Généralement, la réaction est d'arrêter la mission et de se placer en mode *Attente*. Une replanification rapide en fonction des événements variables permet d'éviter ces interruptions ainsi que de tenir compte du résultats des actions précédentes. Le planificateur CASPER (Continuous

12. Fault Detection, Isolation and Recovery : ensemble des techniques et des technologies destinées à détecter des défaillances en ligne et à y remédier

13. Geosynchronous Orbit : orbite géocentrique à 35800 kilomètres sur laquelle un satellite se déplace dans le même sens que la Terre (d'ouest en est) et dont la période orbitale est égale à la période de rotation de la Terre.

Activity Scheduling, Planning, Execution and Replanning) [Knight *et al.*, 2001] raffine par exemple un plan d'une semaine en actions journalières et adapte ainsi en permanence le plan global. D'autres techniques comme la planification contingente ou *anytime* s'approchent aussi de ce concept.

Accès direct : habituellement, les centres de missions envoient leurs requêtes au centre de contrôle qui prévient les conflits et commande le satellite. On peut supposer que, si le satellite possède des planificateurs bord et qu'un accès direct entre satellite et centre de mission est possible, les centres de mission envoient directement leurs requêtes au satellite. Ce dernier évitera le besoin de coordination au sol en traitant lui-même les conflits.

Contrôle d'orbite autonome : l'autonomie n'est pas possible sans ce contrôle dorénavant présent dans plusieurs démonstrateurs. Cela réduit les opérations sol et procure de la précision en pouvant procéder à des modifications d'orbites plus fréquentes et plus fines.

Télémesure réduite : l'autonomie suppose des contacts moins fréquents avec le sol. La télémesure doit donc être condensée afin d'envoyer une information pertinente au sol avec moins de données.

FDIR avancée : la planification bord doit être couplée avec la FDIR afin de pouvoir reprendre des missions interrompues sans intervention du sol. Cela suppose des systèmes de diagnostic capables de prendre en compte des ressources détériorées. Peu d'architectures peuvent le faire mais Deep Space One en était doté.

2.3 Flots de données et structures organisationnelles

Traditionnellement, les satellites non autonomes sont contrôlés par le sol. Ce dernier génère les plans pour l'ensemble de la constellation et les transmet à chaque satellite. Le processus de communication peut être accéléré si les satellites sont dotés de liaisons intersatellites.

Au sein d'une constellation de satellites autonomes en termes de planification, des données doivent être échangées afin de se passer du segment sol tout en garantissant une planification correcte. Ces flots de télémesures et de télécommandes circulent suivant la structure organisationnelle de la constellation, c'est-à-dire la manière dont les satellites interagissent. Pour un ensemble de n satellites formant une constellation, quatre grandes structures peuvent être dégagées à partir des travaux de [Barrett, 1999, Botelho, 2000, Campbell et Schetter, 2002, Johnson *et al.*, 1999, Schetter *et al.*, 2003].

Les structures **totale**ment **centralisées** sont des instanciations du paradigme *maître-esclave*. Il y a 1 unique satellite *dirigeant* qui reçoit les commandes du sol et produit les plans pour $n - 1$ satellites *subordonnés*. Ces derniers ne sont donc pas indépendants et possèdent une autonomie relativement faible alors que le satellite dirigeant doit être autonome. L'implantation de ce type de structure est relativement aisée mais possède un

	Hiérarchique	Distribuée
Implantation	Aisée	Difficile
Optimalité	Oui	Pas toujours
Poids des calculs	Lourds	Légers
Robustesse	Un point de fragilité	Dégradation progressive

FIGURE 1.5 – Structures hiérarchisées ou distribuées : des approches orthogonales

fort point de vulnérabilité. Si pour une raison ou une autre, le satellite dirigeant venait à tomber en panne, la constellation entière le serait aussi.

Les structures **partiellement centralisées** sont fort semblables aux structures totalement centralisées dans le sens où il y a toujours 1 unique satellite dirigeant et $n - 1$ satellites subordonnés. Cependant, ces derniers possèdent une autonomie plus élevée et déchargent le dirigeant d'une partie de son travail en produisant des plans partiels qui n'ont plus qu'à être fusionnés et mis en cohérence.

Les structures **décentralisées hiérarchisées** sont des ensembles où le pouvoir décisionnel est distribué à un sous-ensemble de satellites qui agissent comme autant de processus centraux pour de petits systèmes centralisés et interagissant entre eux. Ainsi, il y a x satellites dirigeants (où $x < n$) travaillant de façon décentralisée et $n - x$ satellites subordonnés. On peut donc considérer que la constellation est composée de x sous-constellations.

Les structures **décentralisées distribuées** sont les cas extrêmes de décentralisation. Chaque satellite de la constellation possède un haut niveau d'autonomie et interagit avec tous les autres. Il n'y a plus de processus centralisateur mais un processus global de négociation et de délibération. Ce sont les structures les plus difficiles à mettre en œuvre mais les plus robustes en termes de planification aux défaillances d'un satellite.

La Figure 1.5 met en lumière les différences entre structures hiérarchisées et structures distribuées. Une *approche hiérarchisée* est plus simple à implanter mais une panne du satellite dirigeant provoquera une panne du système dans son entier, tandis qu'une *approche distribuée* est plus difficile à mettre en œuvre mais est plus robuste. Les deux approches ont leurs avantages et leurs défauts. D'ailleurs, un compromis a souvent été fait : une structure partiellement centralisée avec un satellite dirigeant et au moins un, parmi les subordonnés, qui servira de sauvegarde et prendra le relais en cas de panne [Bornschlegl *et al.*, 2001, Schetter *et al.*, 2003].

2.4 Plusieurs générations d'agents

Mettre en œuvre l'autonomie des satellites passe par la définition d'architectures logicielles. Ces architectures se décomposent en couches ou modules, chacun contrôlé par

un agent. En effet, l'agentification des systèmes bord permet de considérer les différents concepts d'autonomie. Les bénéfices en sont les suivants :

1. décomposition des traitements complexes ;
2. adaptabilité en vol par le chargement dynamique d'agents ;
3. robustesse en incorporant des FDIR à la fois de haut et bas niveau ;
4. réduction du contrôle au sol.

La première utilisation d'agents pour satellite a été testée en 1999 avec RemoteAgent pour Deep Space One [Muscettola *et al.*, 1998]. Il s'agit d'une architecture en trois couches sur un système spatial traditionnel. La couche de planification raisonne par construction de buts. À partir d'un ensemble de buts de haut niveau, le planificateur de RemoteAgent utilise la programmation par contraintes pour bâtir un plan sur un horizon limité. RemoteAgent replanifie dès que cet horizon est atteint ou qu'une erreur se produit à l'exécution du plan.

En 2001, une architecture appelée ObjectAgent [Surka, 2001], qui est une extension de RemoteAgent, est définie pour les satellites de TechSat-21 [Sherwood *et al.*, 2001]. Il s'agit d'une architecture modulaire hétérogène fondée sur l'échange de messages au travers d'une *boîte aux lettres*. Chaque module est contrôlé par un agent. Ce dernier est un processus en multi-threading possédant un ensemble de *compétences*. Ces compétences peuvent incorporer des outils comme de la logique floue, des réseaux de neurones, des procédures d'apprentissage et des systèmes de traitement des exceptions. Toutes nécessitent des entrées, pouvant provenir d'autres agents, et produisent des résultats de sortie, pouvant à leur tour être demandés par d'autres agents.

Cette approche a été généralisée par l'architecture ACT (**A**gent **C**oncepts **T**estbed) [Truszkowski *et al.*, 2004]. Cette dernière reprend les concepts d'ObjectAgent mais définit les modules fondamentaux qui communiquent entre eux dans un cadre de publication-souscription¹⁴ :

1. le *percepteur* reçoit des données de capteurs et des antennes ;
2. le *modélisateur* utilise ses données pour mettre à jour les connaissances du satellite ;
3. le *communicateur* assure la communication entre les modules ;
4. le *raisonneur* définit les buts du satellite en fonction du modèle ;
5. le *planificateur* construit le plan nécessaire à la satisfaction de ces buts ;
6. l'*exécuteur* réalise le plan.

Plus récemment, [Lemaître et Verfaillie, 2007, Charneau et Bensana, 2005] et [Rodriguez-Moreno *et al.*, 2007, Aschwanden *et al.*, 2006] ont proposé deux architectures relativement proches, respectivement AGATA (**A**utonomy **G**eneric **A**rchitecture **T**ests and

14. Méthode de distribution et d'accès aux données dans laquelle les applications (modules, agents ou logiciels) publient leurs données auprès d'un greffier et s'abonnent à celui-ci pour en obtenir.

Applications) et IDEA (Intelligent Distributed Execution Architecture). Ces architectures sont structurées en deux couches où interagissent un module réactif et un module délibératif. Le module réactif décide pour l'action suivante en fonction d'heuristiques mais, s'il en a le temps, il peut demander au module délibératif un plan construit par un algorithme *anytime*.

3 Comme un unique satellite virtuel

Dans la littérature, l'idée principale en matière de contrôle d'une constellation de satellites autonomes est, pour les opérateurs au sol, de ne considérer qu'un *unique satellite virtuel* [Zetocha, 2000, Johnson *et al.*, 1999], c'est-à-dire une entité virtuelle maintenant l'état de la constellation, assistant lors des optimisations de ressources et servant d'interface avec le sol. Or RemoteAgent, ObjectAgent, AGATA et IDEA sont définis pour contrôler un unique satellite.

Dans cette section, nous allons dans un premier temps détailler les architectures qui ont été proposées pour permettre l'autonomie des constellations de satellites. Dans un deuxième temps, nous nous intéresserons à la manière dont les activités d'une constellation sont planifiées.

3.1 Architectures pour l'autonomie des satellites

Une extension d'ObjectAgent, appelée TeamAgent [Mueller *et al.*, 2001], a été proposée pour contrôler une constellation. Un ensemble d'agents représente un satellite autonome de la constellation et l'ensemble du système multiagent représente l'unique satellite virtuel. Les tâches sont réparties par protocole ContractNet et chaque agent possède les accointances nécessaires pour savoir à quelle autre entité s'adresser. Par exemple, le **SensingAgent** maintient en permanence une représentation de l'état d'un satellite donné. Le module chargé de représenter l'état global de la constellation va alors s'adresser à lui afin d'obtenir les informations nécessaires. Un autre exemple est celui du **DispatcherAgent** qui se charge d'envoyer les messages créés par les autres modules. La seule différence entre ObjectAgent et TeamAgent est le fait que certains liens de communication entre deux modules sont en fait assurés par une liaison intersatellite.

Cette première approche a été appliquée pour une formation¹⁵ de trois satellites interconnectés par ISL [Mueller *et al.*, 2001]. Les trois satellites sont hiérarchiquement organisés autour d'un *dirigeant* qui reçoit toutes les commandes de haut niveau et fait une planification globale avec un planificateur CASPER. Dans cette application :

15. Une formation de satellites, contrairement à une constellation, est composée d'un petit nombre de satellites très proches les uns des autres (entre 100 mètres et 10 kilomètres) qui doivent maintenir des orbites relatives très précises.

1. le satellite dirigeant calcule la trajectoire optimale en ressource pour chaque satellite en fonction des buts de haut niveau et la transmet ensuite aux autres satellites. Le planificateur calcule les états potentiels de chaque satellite et un algorithme de simplexe calcule le contrôle optimal pour chaque état ;
2. chaque satellite individuel est responsable de son maintien sur la trajectoire définie par le dirigeant. Pour cela une boucle de contrôle rétroactive *locale* à chaque agent utilise les données orbitales de chaque satellite pour définir sa position et sa vitesse relative. Ces résultats permettent ensuite de déduire les états futurs possibles et ajuster la trajectoire en fonction.

L'architecture ANTS (**A**utonomous **N**ano-**T**echnology **S**warm) [Truszkowski *et al.*, 2006, Curtis *et al.*, 2003] s'inspire d'une métaphore biologique pour contrôler une constellation de 1000 satellites. Dans une fourmilière, chaque individu est hautement spécialisé. Les pico-satellites de la constellation sont donc des experts dans un domaine de compétence bien que tous possèdent les compétences nécessaires à leur survie comme le contrôle d'orbite autonome ou la FDIR. Ainsi, les satellites sont subdivisés en trois groupes :

1. *ouvriers*, ils possèdent une spécialisation ;
2. *messagers*, ils sont chargés de la communication ;
3. *dirigeants*, ils planifient globalement pour la constellation.

Dans cette approche, les ouvriers se subdivisent en équipes élastiques tandis que les messagers et les dirigeants se positionnent pour créer le réseau de communication interne à la constellation afin de permettre aux informations de circuler en permanence. Les compétences sont mises en commun au sein de ces équipes pour former des instruments scientifiques virtuels sur lesquels les tâches sont réparties par les dirigeants. Cette architecture souffre cependant de plusieurs limites :

1. elle est hautement prospective et aucun des concepts fondamentaux n'a été formalisé ;
2. elle est destinée à contrôler un essaim de satellites hétérogènes, ce qui est très éloigné des constellations qui sont aujourd'hui mises en œuvre.

Remarquons qu'aucune de ces deux architectures n'a été effectivement mise en œuvre : le projet TechSat-21 pour lequel TeamAgent était développé a été abandonné en 2003 et la constellation associée à l'architecture ANTS n'est pas réalisable à l'heure actuelle.

3.2 Planification collective en ligne

Le segment sol reçoit des utilisateurs des requêtes de mission. Afin d'être réalisées par les satellites, ces dernières sont traduites en plans d'activités pour les satellites. Dans le cas de satellites non autonomes, cette planification est faite au sol et leur est ensuite transmise. Cependant, cette phase monopolise des ressources et comme les fenêtres de visibilité entre

satellites et stations sol sont limitées, la réactivité face aux nouveaux événements est relativement faible.

Dans le cadre de missions où les satellites possèdent une certaine autonomie, l'opérateur sol ne leur communique que les requêtes de mission sous forme de buts de haut niveau et de dates butoirs d'exécution. La planification et l'ordonnancement sont faits à bord. On peut distinguer trois principales méthodes de planification :

1. la **construction de buts** (GA comme **Goal Achievement**) [Damiani, 2005, Damiani *et al.*, 2005, Bornschlegl *et al.*, 2001, Guettier et Poncet, 2001, Bornschlegl *et al.*, 2000, Mueller *et al.*, 2001, Muscettola *et al.*, 1998] : à partir d'un état initial et d'un ensemble d'actions modifiant l'état courant, une suite d'actions menant à un état but est construite de manière incrémentale. Cette première méthode est simple d'implantation car elle ne nécessite qu'une connaissance des actions, de leurs préconditions et de leurs effets ;
2. la **construction de réseaux hiérarchiques de tâches** (HTN comme **Hierarchical Task Network**) [Brambilla *et al.*, 2005, Knight *et al.*, 2001, Das *et al.*, 1999, Das *et al.*, 1995] : les tâches non primitives sont décomposées en tâches primitives suivant des méthodes données *a priori* puis les conflits entre tâches sont détectés et résolus avant de réappliquer cette méthode sur le plan ainsi obtenu. Cette deuxième méthode est plus efficace en ligne car elle réduit l'espace de recherche. En contrepartie, elle nécessite l'utilisation d'heuristiques de décomposition ;
3. la **métaheuristique évolutionniste** (EA comme **Evolutionary Algorithmic**) [Carrel et Palmer, 2006, Carrel et Palmer, 2005, Globus *et al.*, 2003] : l'ordonnancement entre les tâches est fait par une approche stochastique inspirée des théories évolutionnistes au travers de la notion de *sélection naturelle* et de *reproduction*. Si cette méthode est *anytime*, sa lenteur de convergence l'empêche de faire de la planification réactive.

[Brambilla *et al.*, 2005, Das *et al.*, 1999, Das *et al.*, 1995] ont travaillé sur un planificateur trois couches nommée DIPS (**D**istributed **I**ntelligent **P**lanning and **S**cheduling). Chaque couche est un module fonctionnel. Seule la première, la couche **Interface**, a accès aux perceptions venant de l'environnement : elle gère les perceptions, actions et communications au sein du système à l'aide de scripts et de messages KQML. La seconde couche, la couche **Inférence**, gère les connaissances et accointances ainsi que la décomposition des tâches et la planification des tâches primitives nécessitant des ressources locales. La troisième couche, la couche **Contraintes**, gère l'ordonnancement des tâches en maintenant une base de contraintes.

Si la planification dans [Das *et al.*, 1999, Das *et al.*, 1995] se faisait par de classiques réseaux de tâches hiérarchiques, [Brambilla *et al.*, 2005] proposent une amélioration de ce processus de planification pour les constellations de satellites. À l'aide d'une décomposition hiérarchique partiellement ordonnée des tâches (HD-POP comme **H**ierarchical

Decomposition **P**artial-**O**rdre **P**lanning), ils proposent un algorithme qui part d'un plan partiel et l'étend à chaque itération en ajoutant des liens de causalité pour satisfaire les préconditions des actions. En cas de menace ou d'inconsistance, l'algorithme fait un *backtrack*. Les tâches non primitives sont décomposées en sous-tâches et ainsi de suite. Toutes les préconditions qu'un agent peut créer sont construites. Celles qui restent seront générées par une phase de coordination entre agents qui est en fait une centralisation. Chaque satellite envoie son plan partiel à un satellite central qui fusionne les plans avant de renvoyer un plan local à chaque satellite.

[Dungan *et al.*, 2002] proposent un algorithme centralisé HBSS (**H**euristic **B**iased **S**tochastic **S**ampling) modifié qui combine la recherche stochastique, la recherche heuristique et la propagation de contraintes pour planifier les activités d'une constellation de satellites. Résoudre ce problème consiste à trouver une séquence d'observations pour chaque satellite de manière à maximiser l'utilité globale des observations et résister à un fonctionnement dégradé. Tant qu'il reste des observations, le planificateur en sélectionne une par un biais stochastique et l'ajoute au plan courant en l'assignant à un satellite, puis les contraintes induites sont propagées. L'heuristique qui sélectionne l'observation se fonde sur une mesure de conflit temporel (une observation qui n'a pas assez d'opportunités de réalisation en fonction de sa priorité) et de conflit matériel (une observation qui réclame de nombreuses ressources). L'algorithme choisit alors :

1. l'observation qui possède la plus grande priorité et la plus grande mesure de conflit ;
2. l'observation qui possède la plus petite mesure de conflit temporel.

[Bornschlegl *et al.*, 2001, Guettier et Poncet, 2001, Bornschlegl *et al.*, 2000] ont proposé une planification des trajectoires pour une constellation de satellites à l'aide de programmation par contraintes. Les auteurs modélisent le problème à l'aide d'un *graphe de navigation*¹⁶ et un *modèle temporel* qui dénote pour chaque arête un temps de transition associé dépendant de l'arête précédente. Les contraintes dépendent ensuite de la mission : dans le cadre d'une mission de surveillance terrestre un chemin doit au moins contenir une arête dont le trajet passe par la zone à surveiller. Cependant cette approche souffre de deux limitations :

1. un plan partiel global est calculé par un satellite centralisateur ;
2. chaque agent peut recalculer son plan localement pour prendre en compte plus finement les ressources du satellite et ajouter les actions nécessaires à sa sécurité. Or ceci peut créer des plans locaux contradictoires avec le plan global.

[Globus *et al.*, 2003] ont comparé différentes approches évolutionnistes afin de planifier et ordonnancer la réalisation d'observations au sein d'une constellation de satellites. Trois types d'algorithmes ont été utilisés : (1) algorithme glouton stochastique (avec une

16. Il s'agit d'un graphe dont les sommets sont des points de navigation et les arêtes des trajets dans l'espace. L'utilisation d'un tel graphe pour un problème monosatellite avait déjà été proposé par [Gabrel, 1994].

heuristique **Hill Climbing**¹⁷); (2) algorithme génétique; (3) méthode de recuit simulé. Si [Globus *et al.*, 2003] s'intéressent à l'effet des différents opérateurs évolutionnistes (nombre de descentes et de générations, mutation, recombinaison, température) ces méthodes possèdent deux inconvénients majeurs :

1. elles sont statiques, ce qui signifie qu'aucune nouvelle tâche ne peut être introduite dans le système sans devoir recommencer l'ensemble du calcul. Cette limite a été palliée par l'algorithme NEAT (**N**ear-optimal **E**volutionary **A**utonomous **T**ask-manager) de [Carrel et Palmer, 2006, Carrel et Palmer, 2005] qui ajoute ou retire des gènes à chaque individu lorsqu'une nouvelle tâche est introduite dans le système ou lorsqu'une tâche est réalisée;
2. elles sont centralisées et planifient pour l'ensemble de la constellation en même temps, signifiant qu'aucune nouvelle tâche ne doit arriver avant de pouvoir communiquer le plan courant à chaque satellite.

[Damiani, 2005, Damiani *et al.*, 2005] s'intéressent à une constellation de satellites qui ne communiquent pas entre eux. Dans cette approche, une première planification centralisée sans incertitude est faite au sol pour répartir les requêtes de mission entre les satellites (avec éventuellement des redondances). Ces ensembles de requêtes sont ensuite transmis individuellement aux satellites qui les planifient. De plus, les satellites peuvent détecter par eux-même des zones intéressantes à observer et générer une requête de mission. Par la suite un mécanisme de répartition à bord de chaque satellite est prévu, qui leur permet d'abandonner les tâches qu'ils croient avoir été effectuées par d'autres. En effet, au sein de la constellation, les satellites sont capables de connaître les fenêtres de télédownload des autres et considèrent qu'une observation est affectée à un satellite si et seulement si ce dernier peut la télédownload au plus vite, c'est-à-dire qu'il est le premier à survoler une fenêtre de visibilité après avoir fait l'observation. Ainsi chaque observation affectée à un autre satellite est déclassée en réduisant sa priorité à la priorité nulle. Les limites de cette approche sont les suivantes :

1. elle ne considère pas des satellites pleinement autonomes en termes de planification car une partie de la planification est déléguée aux stations au sol;
2. elle n'envisage pas les liaisons intersatellites et définit la coopération comme opportuniste et non pas proactive.

Les communications entre satellites sont prises en compte par [Florio *et al.*, 2006, Florio *et al.*, 2005] et le planificateur SCOOP (**S**atellite **C**onstellations **O**ptimal **O**perations **P**lanner). Il s'agit cependant d'un planificateur centralisé au sol qui replanifie chaque fois qu'un satellite de la constellation passe en visibilité. Pour chaque observation à planifier, SCOOP détermine quels satellites peuvent recevoir des télécommandes par ISL et ne considèrent que ces derniers, fixant le plan des autres. Cependant, le planificateur ne possède

17. Cette méthode consiste à se doter d'une métrique heuristique et de ne considérer que les opérateurs qui améliorent cette métrique.

que la télémétrie du satellite en visibilité : son information sur l'état de la constellation est partielle et possiblement erronée. Ainsi une défaillance locale peut avoir d'importantes conséquences sur le plan global.

4 Quel est notre problème ?

4.1 Limites des approches actuelles

Nous avons vu dans la section précédente que les méthodes de contrôle en termes d'architecture et de planification pour constellations de satellites appartiennent à l'une des deux catégories suivantes :

1. elles sont centralisées dans le sens où un satellite dirigeant assure la plus grande partie de la planification [Brambilla *et al.*, 2005, Florio *et al.*, 2006, Bornschlegl *et al.*, 2001, Mueller *et al.*, 2001, Schetter *et al.*, 2003]. Ces approches assurent la stabilité du plan généré mais sont très sensibles aux défaillances locales, peu réactives en cas d'arrivée de nouvelles tâches (voire statiques) et nécessitent de plus importantes capacités de calcul individuel ;
2. elles ne sont pas centralisées [Damiani, 2005, Damiani *et al.*, 2005] et assurent la sécurité du plan mais elles ne tirent pas parti des liaisons intersatellites pour améliorer la qualité des plans locaux ;

Un compromis peut toutefois être trouvé avec par exemple l'utilisation de planification contingente [Dearden *et al.*, 2003, Dearden *et al.*, 2002, Meuleau et Smith, 2003], dérivée de la méthode d'ordonnancement JIC (**J**ust **I**n **C**ase **S**cheduling) pour les télescopes automatisés. L'idée centrale est de déterminer les points d'un plan initial où les risques d'échec sont les plus importants et de construire de façon incrémentale des branches alternatives à ces situations. Cependant, plus on accroît le nombre d'insertions, plus la complexité temporelle du problème grandit, ce qui est un handicap important au niveau des systèmes embarqués.

Au vu de cela, nous nous proposons de développer dans ce manuscrit une approche décentralisée du contrôle d'une constellation de satellites autonomes en tirant parti des liaisons intersatellites.

Une coordination doit donc se faire entre les stations au sol et les satellites pour avoir une réelle efficacité. En effet, les n satellites d'une même constellation peuvent au pire cas faire n fois la même observation s'ils ne se coordonnent pas. Plus généralement, nous considérons la coordination et la coopération comme des concepts d'autonomie fondamentaux pour tirer parti de l'autonomie individuelle des satellites d'une constellation.

4.2 Description de la mission

Nous nous proposons d'étudier une mission d'observation de la Terre inspirée de la mission Fuego [Escorial *et al.*, 2003] et d'envisager les liaisons intersatellites au sein d'une constellation de satellites **non agiles** et **autonomes** en termes de **planification bord**, **télémesure réduite**, **contrôle d'orbite**. Ainsi donc le contrôle individuel des satellites se trouve hors de notre champ de réflexion.

Les constellations que nous considérons sont composées de 3 à 16 satellites tels que leurs orbites se croisent aux pôles. Un positionnement judicieux des satellites sur les orbites permet d'envisager que deux (ou plus) satellites passent à proximité l'un de l'autre dans les régions polaires, et puissent donc communiquer entre eux *via* une liaison intersatellite sans intervention du sol. Intuitivement, la communication intersatellite augmente la réactivité de la constellation puisque chaque satellite n'est en visibilité du sol (et n'a donc de possibilité de communication avec ce dernier) que 10 % du temps.

Le sol envoie à la constellation des requêtes d'observation asynchrones et de priorités variables. Les satellites sont équipés d'un unique instrument d'observation dont un miroir peut décaler en roulis la ligne de visée. Une durée minimum étant nécessaire pour actionner le miroir, deux requêtes d'observation trop proches géographiquement peuvent ne pas être réalisables par un même satellite. De même, chaque satellite est contraint en ressources et ne peut réaliser qu'un nombre limité d'observations avant de devoir télédécharger. Les satellites sont aussi équipés d'un instrument de détection pointé vers l'avant qui permet de détecter une zone intéressante à la volée et de générer à bord une requête d'observation.

Notre problème est un problème d'affectation décentralisée de tâches dans un système multiagent où de nouvelles tâches arrivent de manière asynchrone et où les communications sont intermittentes. La constellation doit donc construire et réviser un plan de tâches tel que :

1. le nombre de tâches réalisées par la constellation soit le plus élevé possible ;
2. les tâches soient réalisées le plus rapidement possible ;
3. le nombre de tâches prioritaires non réalisées et le nombre de redondances soient le plus bas possible.

Dans ce chapitre, nous avons vu que le paradigme multiagent était adapté au contrôle d'une constellation de satellites autonomes d'observation de la Terre, d'autant plus lorsque les satellites communiquent par liaisons intersatellites. Nous proposons donc une approche décentralisée du contrôle d'une constellation en tirant parti des communications. Il s'agit alors de se poser les questions suivantes : comment définir l'état global de la constellation et comment prendre une décision et générer des commandes ? Ces deux questions reviennent à définir l'autonomie en termes de commandes (et donc de planification) et en termes de suivi de situation (comment les satellites échangent des informations pour partager une information commune). Il nous faut en premier lieu modéliser la constellation de satellites.

Chapitre 2

Modéliser une constellation

Nous avons vu dans le chapitre précédent qu'un système multiagent était adapté pour contrôler une constellation de satellites autonomes. Dans ce chapitre, nous présentons en premier lieu ce qu'est un agent et un système multiagent. Nous traiterons alors la question de la coordination et de l'organisation au sein du système. À partir de ces considérations, nous proposerons une modélisation d'une constellation de satellites.

1 Des agents qui s'organisent

1.1 Qu'est-ce qu'un agent ?

Depuis une vingtaine d'années la notion d'agent et plus particulièrement d'agent autonome est présente dans la littérature. À partir de la définition de [Franklin et Graesser, 1996], nous définissons ainsi un agent :

Définition 2.1 (Agent)

Un agent est une entité naturelle ou artificielle plongée au sein d'un environnement, capable de percevoir localement ce dernier à l'aide de capteurs et d'agir sur celui-ci à l'aide d'effecteurs.

Nous pouvons considérer plusieurs types d'agents, selon leurs caractéristiques :

- capacité de communiquer, directement ou non, avec d'autres agents ;
- capacité de se reproduire ou d'être reproduit ;
- vue locale de l'environnement ;
- représentation partielle de cet environnement ;
- comportement téléonomique (visant à satisfaire des buts modélisés) ;
- comportement réflexe (répondant à une perception directe).

	Téléonomique	Réflexe
Réactif	Pulsionnel	Tropique
Délibératif	Intentionnel	Module

FIGURE 2.1 – Les différents types d’agents

Classiquement, on distingue dans la littérature quatre types d’agents selon son architecture (réactive ou délibérative) et selon son comportement (téléonomique ou réflexe) comme le présente le tableau de la Figure 2.1.

Agents réactifs

Les premiers sont dits agents réactifs et sont inspirés par les travaux de Rodney Brooks sur la subsumption [Brooks, 1986]. Les agents ne construisent pas de modèle du monde mais réagissent automatiquement à un certain nombre d’événements. Ils ont donc des comportements prédéfinis et la superposition de ces derniers fait émerger des comportements plus complexes. Cependant comme le fait remarquer [Drogoul, 1995], un tel comportement global est :

- sous-optimal en raison des interactions négatives entre les comportements locaux ;
- peut être observé mais est difficilement représenté formellement ;
- difficilement portable d’une application à l’autre.

Brooks a défini l’architecture réactive comme étant *située* dans le sens où le monde est le seul modèle possible, *incarnée* car le monde est irréductible, *intelligente* car il y a une dynamique d’interaction avec le monde et enfin *émergente* car le système œuvre intelligemment pour le regard extérieur. Trois nouvelles notions furent introduites par [Werger, 1999] : *minimaliste* signifiant que l’on cherche pour chaque comportement la configuration minimale en ressources, *sans état* car l’agent réactif ne doit avoir pas ou peu d’états internes et *tolérante* dans le sens où l’on ne tente pas de réduire l’imprécision du monde.

Un agent réactif au comportement téléonomique est dit **pulsionnel** tandis qu’un agent réactif au comportement réflexe est dit **tropique**.

Agents délibératifs

Le second type d’agent est délibératif ou cognitif, signifiant que les agents sont capables de raisonner sur leur environnement. L’agent est proactif dans le sens où il cherche à satisfaire des buts intrinsèques ou définis *a priori*. Pour cela, il est doté d’un modèle du monde qui lui permet de prédire l’évolution de ce dernier et d’un modèle d’action qui lui permet de prédire les modifications de l’environnement suite à une action donnée. On résume ceci par : percevoir - modéliser - planifier - agir. Les architectures BDI (**B**elief

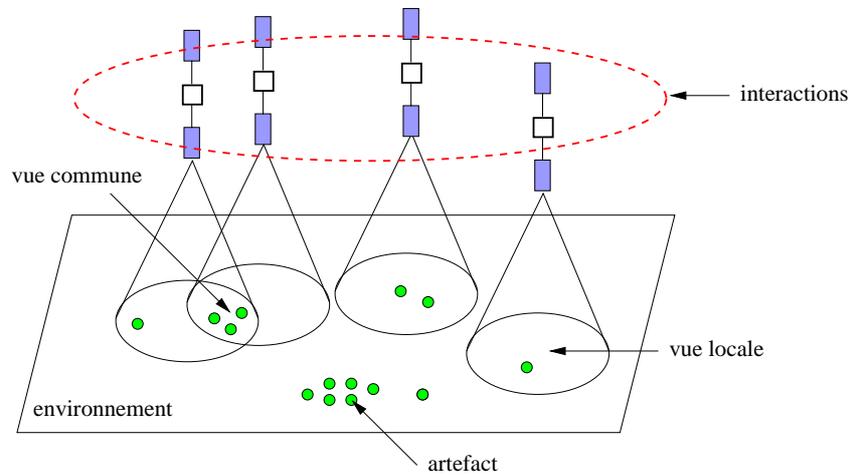


FIGURE 2.2 – Un système multiagent

Desire Intention) sont l'exemple type des architectures délibératives [Balbiani et El Fallah-Seghrouchni, 1998, Cohen et Levesque, 1990].

Les agents cognitifs sont souvent **hystérétiques**, c'est-à-dire capables de conserver des informations sur leurs états passés et sur l'environnement, par exemple en utilisant une mémoire, et de les réutiliser pour gérer leur comportement.

Un agent cognitif au comportement téléonomique est dit **intentionnel** tandis qu'un agent cognitif au comportement réflexe est dit **module**.

1.2 Qu'est-ce qu'un système multiagent ?

De par sa nature locale, un agent n'existe pas indépendamment de l'environnement dans lequel il est situé, ni des autres agents. Cet ensemble forme un système multiagent, illustré en Figure 2.2, c'est-à-dire une structure composée de :

- un environnement (espace généralement pourvu d'une métrique) ;
- un ensemble d'agents (objets actifs dans l'environnement) ;
- un ensemble d'artefacts (objets passifs pouvant être manipulés) ;

Un système multiagent est défini comme la combinaison de ses agents qui interagissent pour altérer l'environnement et les artefacts. Ces derniers peuvent à leur tour altérer les agents selon un principe de *rétroaction*.

Deux niveaux d'interactions sont présents dans un système multiagent :

1. une interaction entre un agent et le système dans sa globalité ;
2. une interaction entre deux agents.

La première est le fait de l'utilisation d'informations communes, de la réception de nouveaux buts et de reformulation des modèles généraux. La seconde est le fait de la coopération entre agents.

1.3 S'organiser pour coopérer (ou l'inverse)

Comme [Nwana *et al.*, 1996] et [Durfee, 2001], nous distinguons coordination de coopération dans le sens où il peut y avoir coordination sans coopération, comme s'écarter du chemin d'un agent pour le laisser passer, mais qu'il ne peut y avoir de coopération sans coordination. Nous parlons plutôt de *synchronisation* qui représente l'ordonnancement des agents entre eux. La coopération, comme le distingue [Legras, 2003], se décompose en *collaboration* où plusieurs agents aux buts communs interagissent et en *coordination* où plusieurs agents aux buts différents mais devant utiliser une ressource commune interagissent.

Nous pouvons alors distinguer quatre formes de coopération :

- la coopération pour éviter les conflits ;
- la coopération pour profiter de l'action d'un agent extérieur ;
- la coopération pour supprimer les redondances entre différents agents ;
- la coopération pour mener conjointement une action.

Afin de mettre en œuvre ces formes de coopération, les agents doivent structurer leurs interactions. Cette structure est appelée organisation et deux approches sont considérées dans la littérature.

Approche bottom-up

L'approche dite bottom-up signifie que les agents n'ont pas *a priori* de buts communs. Ce sont donc des agents négociants ou compétitifs. La structure organisationnelle émerge avec le temps lorsque les agents coopèrent. Les agents doivent alors négocier ces structures au travers de différents protocoles (comme le Contract Net [Smith, 1981] ou la résolution de contraintes distribuées) dont les qualités ont été définies par [Rosenschein et Zlotkin, 1994]. On distingue deux grandes classes d'organisation bottom-up fondées sur :

1. la notion d'utilité et de gain (théorie des jeux) ;
2. la complémentarité des compétences (réseaux de dépendances).

Approche top-down

L'approche top-down est celle où les agents sont initialement définis avec des buts communs. Ce sont des agents coopératifs et l'on parle alors de travail d'équipe [Fiorino, 1998, Legras, 2003]. Travail d'équipe et collaboration sont très proches : les agents partagent les mêmes buts et ne poursuivent pas d'objectif individuel. Une équipe est formée en fonction des capacités de chacun à participer à l'accomplissement des buts communs.

On cherche alors, le plus souvent, à construire des croyances, des plans et des intentions communes, généralement autour d'un *greffier* ou d'un *dirigeant*. On peut donc en dégager trois notions fondamentales :

1. l'utilisation de connaissances communes ;
2. l'échange d'information de contrôle ;
3. l'élaboration d'une conjecture commune.

2 Un cadre pour les interactions

De nombreuses définitions de la notion d'organisation ont été proposées [Corkill et Lander, 1998, Franklin et Graesser, 1996, Lemaître et Excelente, 1998, Hubner *et al.*, 2002b, Sichman *et al.*, 2005] mais toutes définissent l'organisation d'un système multiagent comme une collection de rôles, de relations et de structures de décision qui gouvernent le système (ou une sous-partie du système).

Définition 2.2 (Organisation)

Une organisation fournit un cadre pour les activités et les interactions entre agents au travers de la définition de rôles, de comportements et de relations d'autorité.

Les travaux de [Hubner *et al.*, 2002b, Hubner *et al.*, 2002a] raffinent cette définition et considèrent les modèles d'organisation comme bâtis selon trois axes principaux :

1. un *aspect structurel*, c'est-à-dire comme nous l'avons vu au Chapitre 1 Section 2.3 la manière dont l'information et les décisions circulent au sein de l'organisation ;
2. un *aspect fonctionnel* qui spécifie les politiques de coordination, d'affectation des tâches et de qualités de résultats ;
3. un *aspect déontique* qui définit les obligations et les permissions des agents.

2.1 Taxonomie des organisations

L'étude des organisations admet généralement qu'il n'existe pas une unique sorte d'organisation adaptée pour tous les problèmes bien que l'on puisse déterminer quelques processus automatiques pour de l'auto-organisation. Cependant, il faut pour cela connaître l'espace des organisations possibles, leurs avantages et leurs coûts. [Baeijs et Demazeau, 1996] distinguent trois classes d'organisations :

1. les **groupes** qui sont des organisations planes dans lesquelles nous pouvons distinguer les *équipes*, les *groupes d'intérêts* (aussi appelés coalitions) et les *communautés de pratiques* (aussi appelées congrégations) ;
2. les **hiérarchies** qui sont des organisations structurées autour du paradigme maître - esclave dans lesquelles nous pouvons distinguer les *hiérarchies simples*, les *hiérarchies uniformes*, les *hiérarchies multidivisionnelles* et les *holarchies* ;

3. les **marchés** qui sont des organisations décentralisées autour d'un mécanisme de coordination fondé sur une métaphore économique.

[Horling et Lesser, 2004] proposent une synthèse de ces différentes organisations. Le tableau de la Figure 2.3 résume les avantages et inconvénients de ces organisations.

Organisation	Avantages	Inconvénients
Hiérarchie	Décompositions facilitées Adaptation à l'échelle	Fragile Délais de communications
Holarchie	Exploite l'autonomie d'unités fonctionnelles	Structure organisationnelle complexe Faible prédictibilité des performances.
Coalition	Exploite le nombre d'agents	Bénéfices à court terme
Équipe	Cohésion accrue S'adapte aux contraintes globales	Communications accrues
Congrégation	Facilite les accointances	Ensembles d'agents parfois trop restrictifs
Société	Système ouvert Ontologie commune	Grande complexité Sophistication accrue des agents
Fédération	Délégué unique qui offre de multiples services	Délégué pouvant devenir un goulet d'étranglement
Matrice	Partage de ressources Influences multiples	Conflits potentiels accrues Sophistication accrue des agents

FIGURE 2.3 – Comparaison des différentes formes d'organisations

Hiérarchies

Les *hiérarchies* ou *organisations hiérarchiques* sont des structures d'arbre où les agents les plus élevés possèdent le point de vue le plus global. Les interactions ne sont pas transversales à la structure mais se font entre deux entités connectées. On a donc un flot d'information ascendant (les entités les plus basses font remonter l'information aux entités les plus hautes) et un flot de décision descendant (les entités les plus hautes transmettent des ordres aux entités les plus basses). Trois formes de hiérarchies peuvent être définies :

1. *simples* : un unique agent en apex prend toutes les décisions ;
2. *uniformes* : l'agent qui possède le plus d'information à un moment donné prend une décision globale ;
3. *multidivisionnelles* : l'organisation est divisée en lignes de production, chacune étant une hiérarchie autonome. L'organisation entre les lignes de production peut prendre n'importe quelle forme.

La puissance des organisations hiérarchiques vient de la décomposition naturelle des tâches. Un protocole de prise de contrat où les agents peuvent décomposer une tâche en sous-tâches leur convient parfaitement. Le principe de *diviser pour régner* qu'elles engendrent permet de maintenir un strict contrôle sur le système en ayant un faible nombre d'interactions pour la population d'agents. Cependant si l'information circule bien, il faut que l'apex de l'organisation possède suffisamment de puissance de calcul pour tirer des analyses et utiliser correctement l'information. De plus, une dégradation du centre de décision se propage à l'ensemble de l'organisation.

Comme pour toute organisation, la forme de la hiérarchie va affecter à la fois le comportement local et le comportement global. Une hiérarchie *plane* où les agents possèdent un haut degré de connectivité peut arriver à saturation si les ressources sont limitées et utilisées par les connexions. D'un autre côté, une hiérarchie *étalée* peut ralentir les performances du système en raison des délais entre les communications.

Holarchies

L'*holarchie* ou *structure holonique* est fondée sur un processus d'organisation similaire à la nidation [Fischer *et al.*, 2003]. Il s'agit d'une structure distribuée de hiérarchies groupées. Chaque groupe possède un caractère distinct des entités qui le composent et contribue au caractère du groupe dont il fait partie. Ces groupes forment la structure de base de l'organisation, que l'on appelle *holon*¹. Chaque holon est composé d'entités subordonnées qui peuvent faire partie d'holon de niveau supérieur. Il y a donc une notion hétérarchique.

Ordinairement, une structure holonique s'adapte bien aux tâches récursivement décomposables. La nature partiellement autonome et encapsulée des holons est le principal intérêt de cette organisation en plus de la flexibilité. Cependant, il est difficile de prédire le comportement du système.

Coalitions

Les *coalitions* [Sandholm *et al.*, 1999, Shehory et Kraus, 1998] sont des ensembles d'agents orientés vers un but donné et généralement avec un cycle de vie court. Elles sont formées avec un objectif précis et sont détruites lorsqu'elles n'ont plus de raison d'être. Autant les agents compétitifs que coopératifs peuvent former de telles coalitions. Bien qu'il puisse y avoir un agent dirigeant qui représente l'ensemble de la coalition, l'organisation est plane. Il n'y a pas de niveaux hiérarchiques car, une fois formée, la coalition est traitée comme une unique entité atomique. Classiquement, il n'y a pas de coordination entre deux coalitions. Dans le cas d'agents compétitifs, de nombreux travaux se sont attachés au partage des gains à la fin du cycle de vie de la coalition.

1. Ce terme est composé des racines grecques *holos* signifiant totalité et *on* signifiant partie.

L'idée fondamentale derrière la formation de coalitions est la suivante : l'utilité ajoutée par au moins un agent est super-additive suivant au moins une dimension. Par analogie, le coût des participants peut être sous-additif. Bien qu'un domaine totalement super-additif conduise les agents à former la *grande coalition*², peu de domaines réels sont totalement super-additifs.

Équipes

Une *équipe* d'agents est un ensemble d'agents coopératifs qui ont décidé de travailler à un but commun. Il s'agit d'une organisation fréquemment utilisée dans les applications multirobots pour le sauvetage urbain ou les systèmes marsupiaux³ [Murphy, 2000]. À la différence des coalitions, une équipe tente de maximiser l'utilité globale du système plutôt que l'utilité individuelle de ses membres. Un des points les plus importants est la représentation explicite du travail d'équipe ou des états mentaux globaux (croyance mutuelle, buts conjoints, etc.)

L'avantage de l'équipe est qu'elle est capable de réaliser ce qu'un agent isolé ne peut faire seul car les capacités des agents se doivent d'être complémentaires [Ott *et al.*, 1999]. De plus, elle est particulièrement bien adaptée à la satisfaction de contraintes globales. Cependant le couplage fort entre les agents mène à une augmentation des communications. De plus, déterminer comment les agents vont décomposer un problème de haut niveau, maintenir la cohérence durant l'exécution et réviser l'organisation de l'équipe sont les difficultés majeures de cette approche.

Congrégations

Similaires aux coalitions et aux équipes, les *congrégations* sont des ensembles d'agents qui sont regroupés en une organisation plane afin d'obtenir des bénéfices additionnels [Griffiths et Luck, 2003, Brooks et Durfee, 2003, Brooks et Durfee, 2002, Brooks *et al.*, 2000]. Contrairement aux deux paradigmes précédents, les congrégations ont un cycle de vie long et ne sont pas formées en vue d'un unique but à satisfaire. Les congrégations sont composées d'agents aux compétences similaires ou complémentaires formant ainsi un ensemble stable de compétences auxquels les agents peuvent faire aisément appel. On suppose que ces agents sont individuellement rationnels et maximisent leur utilité locale à long terme. Il n'y a pas de notion d'utilité collective.

L'idée principale est alors de limiter les recherches et les interactions en définissant un ensemble d'agents qui peuvent travailler ensemble si nécessaire. Par exemple [Pechoucek *et al.*, 2002] génèrent des congrégations pour simplifier la formation de coalitions à partir d'un modèle d'accointances publiques, semi-privées et privées qui sont partagées. Expérimentalement, la génération de congrégations permet de réduire le nombre de messages

2. Coalition unique comprenant l'ensemble des agents du système.

3. Une équipe marsupiale est constituée d'un gros robot qui transporte un ou plusieurs petits robots.

échangés lors de la formation de coalitions car les agents peuvent alors élarger leur recherche. Cependant, le revers de cette stratégie vient du fait que la congrégation peut être trop restrictive sur certains problèmes et ne pas contenir un ensemble d'agents optimal. En effet, plus le nombre de congrégations est élevé, plus le coût de communication est réduit mais, plus le nombre de congrégations est élevé, plus les agents doivent communiquer et partager des informations à l'extérieur pour pallier le manque de ressources.

Sociétés

Les *sociétés* sont des systèmes ouverts⁴ avec un cycle de vie long [Buzing *et al.*, 2005]. Les agents y sont hétérogènes avec différents niveaux de rationalité. La société procure en fait un domaine commun de communication et d'action. C'est un ensemble structuré dont les interactions internes sont flexibles. À l'intérieur d'une société, les agents peuvent être organisés différemment mais un ensemble de *règles sociales* ou de *normes* définissent clairement les interactions entre un agent et le système et les interactions entre les agents eux-mêmes.

Les sociétés d'agents sont donc des cadres généraux qui doivent contenir au moins une autre forme d'organisation. Une société peut avoir une organisation de contrôle, une autre pour la circulation de l'information, une troisième pour la recherche dans un espace donné, etc. Les agents peuvent endosser plusieurs rôles en réponse à différentes formes d'organisation comme dans le modèle AGR (**A**gents **G**roupes **R**ôles) [Ferber *et al.*, 2004].

La difficulté est de trouver un ensemble de règles qui minimisent les conflits et encouragent les solutions efficaces. Ces règles peuvent être données *a priori* ou apprises par renforcement. Cependant, cela n'évacue pas le problème de définition de protocoles et de rôles ainsi que la construction du mécanisme d'entrée/sortie de la société. Bien que le système soit ouvert et flexible, les agents doivent pouvoir connaître et analyser les règles qui le régissent. La description par interfaces communes est une solution potentielle.

Fédérations

Fédérations d'agents ou *systèmes fédéraux* ont tous la caractéristique suivante : un groupe d'agents cède une partie de son autonomie à un unique délégué (aussi appelé courtier ou médiateur) qui représente le groupe dans sa globalité. Les membres du groupe n'interagissent qu'avec leur délégué qui, à son tour, interagit avec le reste de l'environnement. Dans ce sens, la fédération forme une unique interface cohérente.

Les compétences mise à disposition par le délégué différencient la fédération des autres organisations. Il peut agir comme traducteur, commissaire priseur, moniteur d'exécution, etc. Cela permet aux agents d'utiliser toutes les capacités, potentiellement dynamiques,

4. On peut y adjoindre un nouvel agent sans avoir à modifier la structure d'organisation choisie ; nous dirons qu'une société est semi-fermée si les agents extérieurs doivent demander une autorisation pour en faire partie.

de la fédération sans pour autant avoir une connaissance précise des autres agents. Cela réduit la complexité en communication mais ajoute un point de fragilité au niveau du délégué qui peut devenir un goulet d'étranglement.

Organisations matricielles

Les *organisations matricielles* sont décentralisées dans le sens où les hiérarchies, les coalitions, les fédérations utilisent un médiateur ou un agent localement central pour se coordonner. Dans cette organisation, les agents peuvent être influencés par plusieurs agents différents. Les interactions peuvent venir de plusieurs côtés, apportant leur lot d'objectifs, de contraintes et d'importance. On peut représenter ces organisations par une matrice des interactions.

Les organisations matricielles permettent de spécifier explicitement comment le comportement d'un agent ou d'un groupe peut être influencé par de multiples lignes d'autorité.

2.2 Délégation et mécanismes d'interaction

Si nous avons vu au paragraphe précédent les différentes structures organisationnelles qui définissent l'aspect structurel des organisations, les aspects fonctionnels et déontiques sont tout aussi essentiels. En effet, une organisation multiagent repose sur la notion de délégation, *délégation de tâches* qui est un transfert de buts à réaliser sous forme de plans, de commandes ou de conseils et *délégation sociale* qui est un transfert d'autorité sous forme de permissions. Ces deux formes de délégation sont régulées respectivement par l'aspect fonctionnel et l'aspect déontique de l'organisation.

[Schillo *et al.*, 2002] distinguent quatre mécanismes qui président à la délégation :

1. *autorité* : il existe une relation acyclique de pouvoir entre les agents qui détermine les droits à déléguer ;
2. *échange économique* : celui auquel on délègue est payé pour son travail à l'aide d'une monnaie ;
3. *échange de biens* : un jeu de propositions et contre-propositions est instauré pour négocier des objets si le transfert de monnaie est inadapté ;
4. *vote* : les membres d'un groupe utilisent un mécanisme de vote pour élire un représentant. Les contraintes du mandat (durée, obligation et lignes d'autorité) sont définies en même temps que le mécanisme.

Le tableau de la Figure 2.4 indique pour chaque type d'organisation les mécanismes de délégation les plus adaptés et les plus fréquemment mis en œuvre. Si nous ne tenons pas compte des sociétés et des organisations matricielles qui sont efficaces avec n'importe quel type de mécanisme de délégation mais qui nécessitent toujours de disposer d'agents hautement complexes, nous pouvons définir trois classes d'organisations :

Organisation	Échange économique	Autorité	Échange de biens	Vote
Hiérarchie	×	✓	✓	×
Holarchie	×	✓	✓	×
Coalition	✓	×	✓	×
Équipe	×	×	✓	✓
Congrégation	✓	×	✓	×
Société	✓	✓	×	✓
Fédération	×	✓	×	✓
Matrice	✓	✓	✓	✓

FIGURE 2.4 – Organisations et mode de délégation

1. des structures centralisées comme les organisations de types hiérarchiques, holarchiques ou fédérales qui utilisent une relation d'autorité définie *a priori* ou un vote pour s'auto-contrôler ;
2. de larges structures décentralisées comme les congrégations. Ce sont des groupes d'agents qui fusionnent afin de mettre leurs compétences en commun pour dépasser leurs simples capacités. Cette forme d'organisation introduit une délégation sociale à long terme qui est spécifique à une tâche donnée. Chaque agent de la congrégation peut accepter des tâches venant de l'extérieur et demander de l'aide aux membres s'il ne peut l'accomplir seul. Cette organisation utilise essentiellement des échanges économiques ou des échanges de biens ;
3. de petites structures décentralisées représentées par les coalitions ou les équipes. Ces organisations se distinguent des congrégations par un **contrat** qui est passé entre les membres. Ce contrat crée une structure formelle qui détermine les conditions d'interaction. La délégation sociale est faite habituellement par le vote bien que la distribution du profit créée par la coopération s'approche de l'échange économique ou de l'échange de biens. Le contrat détermine alors la manière dont est réparti le profit.

2.3 Adaptation et auto-organisation

Une organisation statique, c'est-à-dire donnée *a priori* et immuable dans le temps, n'est pas à même de résoudre certains types de problèmes. En effet,

1. il n'existe pas une manière optimale de s'organiser ;
2. toute organisation n'a pas la même efficacité ;
3. la meilleure manière de s'organiser dépend de l'environnement.

C'est pour cela qu'une partie de la littérature consacrée aux systèmes multiagents, comme [Gleizes *et al.*, 2008, Serugendo *et al.*, 2006, Georgé *et al.*, 2003, Goldman et Rosenschein, 2002, Schillo *et al.*, 2002], s'est intéressée aux notions d'*auto-organisation*, d'*adaptation* (respectivement la capacité d'un système à modifier son organisation courante et la capacité d'un système à déterminer quelle est l'organisation la plus appropriée à un environnement donné) et de *fluidité* (facilité avec laquelle les agents peuvent modifier l'organisation). Cette approche multiagent se fonde sur les trois principes de la théorie de la stigmergie proposée par [Grassé, 1959] :

1. *intelligence en essaim* inspirée de l'éthologie et de l'entomologie, caractérisant un système qui comporte de très nombreuses interactions entre les agents qui le composent et tel que ces interactions portent essentiellement sur le renforcement de comportements par rétroactions positives ou négatives ;
2. *diminution de l'entropie*, caractérisant le fait qu'un système qui présente un comportement d'auto-organisation sous une influence extérieure doit propager et renforcer cette influence en son sein ;
3. *autopoïèse*⁵, caractérisant la propriété du système à se produire, à se définir lui-même et à se maintenir lui-même dans un état stable ou un équilibre instable (des états organisés qui restent stables pour de longues périodes).

Si les concepts d'émergence et d'auto-organisation sont suffisamment proches pour être parfois confondus, des différences notables existent comme le soulignent [Wolf et Holvoet, 2004]. En effet, l'émergence est un comportement global d'un système surgissant de l'interaction locale de ses parties tel qu'on ne peut retrouver le comportement global en observant les comportements locaux. L'auto-organisation, quant à elle, est un processus dynamique où le système peut acquérir et maintenir une structure sans intervention ou contrôle extérieur. Bien que l'émergence et l'auto-organisation soient par essence dynamiques et robustes, l'une peut exister sans l'autre car l'émergence est avant tout l'apparition d'une propriété tandis que l'auto-organisation est la réduction de l'entropie d'un système.

Cela a conduit à considérer les organisations comme étant le résultat des choix individuels des agents au sein d'une interaction, choix qui sont renforcés en rétroaction par les choix collectifs. Les organisations sont certes résultantes de l'interaction mais pas entièrement dépendantes.

3 Une congrégation de satellites

À partir de ces considérations, comment pouvons-nous modéliser une constellation de satellites ? Et quels seraient les choix d'organisation les plus pertinents ? Deux points

5. Ce terme est composé des racines grecques *auto* signifiant soi-même et *poiësis* signifiant production.

essentiels sont à prendre en compte. Une constellation de satellites est un système multiagent fermé qui possède un important besoin de robustesse en raison du coût de mise en œuvre des agents. Ainsi, les organisations fondées sur des hiérarchies, des holarchies ou des fédérations ne semblent pas convenir. En effet, ces organisations reposent sur la présence d'un agent central dont la défaillance provoque une défaillance globale du système. De plus, étant donné les contraintes de l'environnement spatial sur les capacités de calcul de satellites, des agents trop complexes comme ceux qui sont nécessaires à une société d'agents ou une organisation matricielle ne semblent pas les plus adaptés.

Notre approche sera alors de modéliser une constellation de satellites comme une **organisation composée** fondée sur :

1. une *congrégation* pour l'autonomie en termes de suivi de situation. Comme nous le verrons au Chapitre 3, nous nous fonderons sur un réseau d'accointances publiques pour bâtir une connaissance commune sur les artefacts et l'état du système ;
2. des *coalitions* pour l'autonomie en termes de planification. Comme nous le verrons au Chapitre 4, nous définirons la coopération au sein de sous-ensembles dynamiques d'agents que nous associerons à des coalitions.

Afin de mettre en œuvre cette approche, nous définissons le système multiagent comme une ensemble d'accointances publiques (la constellation), un ensemble d'artefacts (les tâches à réaliser) et un ensemble d'agents autonomes (les satellites).

3.1 Modèle d'accointances fondé sur l'environnement

Une constellation est définie comme suit :

Définition 2.3 (Constellation)

Une constellation \mathcal{S} est un triplet $\langle \mathcal{A}, \mathbb{T}, \text{Voisins} \rangle$ tel que :

- $\mathcal{A} = \{a_1 \dots a_n\}$ est un ensemble de n agents représentant n satellites ;
- $\mathbb{T} \subset \mathbb{N}$ est un ensemble de dates définissant une horloge commune ;
- $\text{Voisins} : \mathcal{A} \times \mathbb{T} \mapsto 2^{\mathcal{A}}$ est une relation symétrique et non transitive qui spécifie pour un agent donné et une date donnée l'ensemble des agents avec lesquels il peut communiquer à cette date.

La relation Voisins correspond au modèle d'accointances de notre système. Elle représente les fenêtres temporelles durant lesquelles les satellites peuvent communiquer lorsque leurs orbites se croisent. Cette relation permet cependant de représenter des modèles de communication plus complexes comme l'utilisation de liaisons interorbites (IOL)⁶.

Afin d'éviter un modèle d'accointance vide, c'est-à-dire que des agents ne puissent pas communiquer avec d'autres à partir d'une date donnée, la relation de voisinage doit

6. L'idée principale de ce type de liaison est d'utiliser un satellite en orbite GEO visible de plusieurs satellites en orbite LEO qui ne se rencontrent pas. Le satellite géostationnaire peut alors servir d'intermédiaire dans la communication.

satisfaire deux propriétés : périodicité et connexité. La *périodicité* spécifie qu'il existe toujours pour un agent donné une date future où il pourra communiquer avec un autre. La *connexité* spécifie que tout agent peut communiquer avec tous les autres agents du système.

Dans notre application, la périodicité de la relation Voisins est intrinsèque. En effet, nous supposons que les satellites communiquent lorsque leurs orbites se croisent. Or les orbites étant périodiques, la relation Voisins qui en découle l'est aussi.

Définition 2.4 (Périodicité)

Soit \mathcal{S} une constellation et $\{p_1 \dots p_n\}$ l'ensemble des durées $p_i \in \mathbb{T}$ du cycle orbital des agents $a_i \in \mathcal{A}$. La période de la relation Voisins, notée $P \in \mathbb{T}$, est le plus petit commun multiple de l'ensemble $\{p_1 \dots p_n\}$.

Dans toute la suite, nous noterons $\mathbb{T}_P \subset \mathbb{T}$ l'intervalle de temps de durée P , soit $\mathbb{T}_P = \{0 \dots P\}$.

La connexité, quant à elle, découle de la topologie spécifique des orbites.

Définition 2.5 (Connexité)

Soit \mathcal{S} une constellation. La relation Voisins est connexe si et seulement si $\forall a_i, a_j \in \mathcal{A}$, $\exists z \in \mathbb{N}$ tel que $\exists \{(a_{\tau_k}, \tau_k) \in \mathcal{A} \times \mathbb{T}, k \in [0 \dots z]\}$ où :

1. $a_{\tau_0} \in \text{Voisins}(a_i, \tau_0)$;
2. $a_{\tau_{k+1}} \in \text{Voisins}(a_{\tau_k}, \tau_k)$ et $\tau_k < \tau_{k+1}$;
3. $a_j \in \text{Voisins}(a_{\tau_z}, \tau_z)$.

La connexité se définit donc sur un intervalle temporel, entre la première et la dernière date de communication possible entre deux agents.

La constellation $\langle \mathcal{A}, \mathbb{T}, \text{Voisins} \rangle$ est une connaissance détenue par l'ensemble des agents. Quoi qu'il en soit, chaque agent dispose d'une connaissance privée.

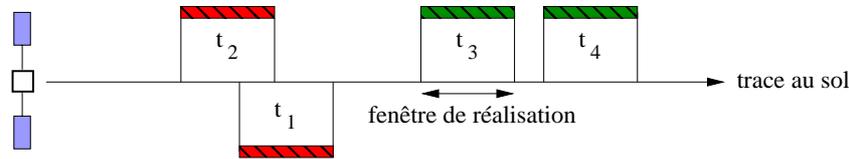
3.2 Des tâches et des contraintes

Dans notre application, le sol ou les satellites peuvent générer des requêtes d'observation. Ces requêtes sont des tâches complexes composées d'une ou plusieurs tâches atomiques.

Définition 2.6 (Tâche atomique)

Une tâche atomique t est un tuple $\langle ID_t, pos_t, prio(t), b_t \rangle$ tel que :

- ID_t est l'identifiant, unique, de t ;
- pos_t représente les coordonnées géographiques de t ;
- $prio(t) \in \mathbb{N}^*$ est une priorité ;
- $b_t \in \{\top, \perp\}$ est un booléen indiquant si t a été réalisée ou non.

FIGURE 2.5 – Exclusion mutuelle entre les tâches atomiques t_1 et t_2

Notons que dans le domaine spatial, la valeur 1 correspond à la priorité la plus haute tandis que la valeur 5 représente la priorité la plus basse. Par conséquent, plus la valeur de $prio(t)$ est faible, plus la tâche atomique t est importante à réaliser. Nous définissons formellement la réalisation comme suit :

Définition 2.7 (Réaliser une tâche atomique - observation)

Soit un agent a_i et une tâche t telle que $b_t = \perp$. Si a_i réalise t à la date $\tau \in \mathbb{T}$, noté $a_i \vdash_{\tau} t$, alors $b_t = \top$, $\forall \tau' > \tau$. Dans toute la suite, nous appellerons observation une tâche atomique réalisée.

Les tâches sont contraintes de trois manières.

(1) Comme réaliser une tâche atomique nécessite du temps, deux réalisations de tâches atomiques ne peuvent pas être effectuées en même temps. Nous pouvons ainsi définir des **exclusions mutuelles** sur un ensemble de tâches atomiques. Par exemple sur la Figure 2.5, les tâches t_1 et t_2 sont en exclusion mutuelle. Cette contrainte est une contrainte locale à un agent. En effet, les fenêtres de réalisation associées à une tâche atomique sont fonction de l'orbite de l'agent. Ainsi pour deux agents distincts, les exclusions mutuelles sur un même ensemble de tâches atomiques ne sont pas nécessairement les mêmes. De plus, il s'agit d'une contrainte dure (qui ne peut être violée) car elle est due à la mécanique spatiale et au mode d'utilisation des instruments de détection.

Définition 2.8 (Exclusion mutuelle)

Soit un agent $a_i \in \mathcal{A}$ et deux tâches atomiques t_1 et t_2 . On dit que t_1 est en exclusion mutuelle avec t_2 , noté $t_1 \bowtie t_2$, si et seulement leurs fenêtres de réalisation pour a_i se chevauchent.

(2) Une tâche complexe peut être **décomposée** en n tâches atomiques. Chacune de ces n tâches atomiques doit être réalisée pour que la tâche complexe le soit. Contrairement aux exclusions mutuelles, les compositions sont des contraintes globales. En effet, pour deux agents distincts les compositions sur un même ensemble de tâches atomiques sont identiques. De plus, il s'agit d'une contrainte souple (qui peut être violée). Formellement,

Définition 2.9 (Tâche complexe)

Une tâche complexe est un sous-ensemble \mathcal{T} de tâches atomiques.

Définition 2.10 (Réaliser une tâche complexe)

Une tâche complexe \mathcal{T} est réalisée à la date $\tau \in \mathbb{T}$ si et seulement si $\forall t \in \mathcal{T}, b_t = \perp, \forall \tau' > \tau$.

(3) Lorsqu'une tâche atomique est réalisée, l'image générée est sauvegardée dans la mémoire de masse de l'agent. Cette ressource mémoire ne se régénère qu'au moment d'un téléchargement. Chaque agent étant limité en mémoire, éviter les redondances peut être intéressant. Nous disons qu'il y a redondance lorsqu'une tâche atomique est réalisée par un agent alors qu'elle a déjà été réalisée par un autre agent. Éviter une redondance est une contrainte locale et souple. Formellement,

Définition 2.11 (Redondance)

Soit $a_i \in \mathcal{A}$ un agent tel que $a_i \vdash_\tau t$. Il y a redondance pour t si et seulement si $\exists a_j \in \mathcal{A}$ et $\exists \tau' \in \mathbb{T}$ ($\tau' \leq \tau$) tels que $a_j \vdash_{\tau'} t$.

Dans toute la suite, nous noterons $\mathcal{T}_{a_i}^\tau = \{t_1 \dots t_m\}$ l'ensemble des tâches connues par l'agent a_i à la date τ .

3.3 Des agents intentionnels

Une intention représente l'attitude d'un agent face à une tâche donnée. Nous pouvons la considérer comme une modalité de la proposition (**l'agent a_i réalise la tâche t**). Ainsi, quatre attitudes sont définies : un engagement ferme qui ne peut être remis en cause par la dynamique de l'environnement, un engagement temporaire qui peut être remis en cause, un refus ferme qui ne peut être remis en cause et un refus temporaire qui peut être remis en cause. Formellement,

Définition 2.12 (Intention)

Une intention $I_t^{a_i}$ de l'agent a_i envers la tâche t est un triplet $\langle \text{mod}(I_t^{a_i}), \text{rea}(I_t^{a_i}), \text{tel}(I_t^{a_i}) \rangle$ tel que :

- $\text{mod}(I_t^{a_i}) \in \{\square, \diamond, \square\neg, \diamond\neg\}$ est la modalité de l'intention ;
- $\text{rea}(I_t^{a_i}) \in \mathbb{T} \cup \{\emptyset\}$ est la date de réalisation de la tâche t ;
- $\text{tel}(I_t^{a_i}) \in \mathbb{T} \cup \{\emptyset\}$ est la date de téléchargement de l'image associée.

Le tableau de la Figure 2.6 présente la sémantique des différentes modalités d'une intention. Un agent peut *proposer* de réaliser une tâche (modalité \diamond) mais cette intention est révisable. L'agent peut *s'engager* à réaliser une tâche (modalité \square) : il ne révisera pas son intention à moins qu'un autre agent s'engage à son tour. À l'inverse, un agent peut se *désister faiblement* (modalité $\diamond\neg$), c'est-à-dire qu'il ne réalisera pas la tâche mais peut réviser cette intention. L'agent peut aussi se *désister fortement* (modalité $\square\neg$) : il ne réalisera pas la tâche et ne révisera pas son intention. Ainsi, l'ensemble des intentions d'un agent correspond à son plan courant. Chaque engagement ou proposition signifie que la tâche associée est planifiée tandis que les tâches associées aux désistements ne le sont pas. Formellement,

Modalité	Sémantique
\square	engagement : a_i s'engage à réaliser t
\diamond	proposition : a_i se propose de réaliser t
$\square\neg$	désistement fort : a_i ne réalisera pas t
$\diamond\neg$	désistement faible : a_i n'envisage pas de réaliser t

FIGURE 2.6 – Les différentes modalités d'une intention

Définition 2.13 (Plan)

Soit un agent a_i et $\mathcal{T}_{a_i}^\tau$ l'ensemble des tâches atomiques connues par a_i . $\forall t \in \mathcal{T}_{a_i}^\tau$, le plan de l'agent a_i à la date τ est défini ainsi :

- la tâche atomique t est planifiée $\equiv (\text{mod}(I_t^{a_i}) \in \{\square, \diamond\})$;
- la tâche atomique t n'est pas planifiée $\equiv (\text{mod}(I_t^{a_i}) \in \{\square\neg, \diamond\neg\})$.

Remarquons alors que l'équivalence entre plan et intentions signifie que les dates de réalisation et de téléchargement ne sont vides que si et seulement si l'intention associée possède une modalité de désistement (fort ou faible).

Dans toute la suite, nous noterons $\mathcal{I}_{a_i}^\tau = (I_{t_j}^{a_i})$ la matrice des intentions connues par l'agent a_i à la date τ .

Planification individuelle et génération des intentions

Dans ce manuscrit, nous nous intéressons à une approche décentralisée de la coopération. Chaque agent doit donc être autonome en termes de planification bord. C'est pourquoi nous supposons que chaque agent possède un planificateur individuel capable, à partir d'un ensemble de tâches données, de retourner un ensemble de tâches à réaliser du point de vue individuel de l'agent.

Si les algorithmes individuels de planification se trouvent hors de notre champ de réflexion, nous pouvons définir abstraitement la planification individuelle comme suit. Pour un agent donné a_i , planifier à la date $\tau \in \mathbb{T}$ est un processus en trois étapes :

1. a_i considère l'ensemble des tâches non réalisées puis calcule un plan de tâches à réaliser. Nous avons choisi d'utiliser pour cela la programmation linéaire en nombres entiers. L'ILP (**I**nteger **L**inear **P**rogramming) consiste à résoudre un problème d'optimisation où la fonction objectif et les contraintes sont toutes linéaires. D'un point de vue géométrique, les contraintes linéaires forment un polyèdre convexe. Si la fonction objectif est elle aussi linéaire, tous les optima locaux sont également des optima globaux ; cela reste vrai si elle est monotone croissante sur chaque variable considérée. Les algorithmes les plus connus pour résoudre ce type de problème sont les algorithmes du simplexe [Dantzig, 1963], celui de [Khachiyan, 1979] et celui de

[Karmarkar, 1984]. Le premier est efficace en pratique bien que son comportement au pire cas soit très mauvais ; tandis que le second est théoriquement performant mais peu efficace en pratique. Le troisième, appelé méthode projective, est efficace en théorie (polynomial) et en pratique (équivalent au simplexe). Pour la résolution de problèmes de programmation linéaire, on considère comme équivalentes les méthodes dérivées du simplexe ou de la méthode projective. De plus, pour la résolution de problèmes de grande taille, une technique comme la génération de colonnes [Dantzig et Wolfe, 1961] peut se révéler extrêmement efficace. Dans notre application, les agents vont individuellement maximiser une fonction objectif dépendante des priorités des tâches tout en satisfaisant les contraintes dures du système (exclusions mutuelles et limitation en mémoire de masse). Formellement,

Définition 2.14 (Planification : définition du problème)

Soit un agent $a_i \in \mathcal{A}$ devant planifier à la date $\tau \in \mathbb{T}$ un ensemble de tâches $\mathcal{T}_{a_i}^\tau$. Soit $M \in \mathbb{N}$ le nombre maximum de tâches que a_i peut enregistrer dans sa mémoire de masse. L'agent a_i planifie en résolvant le problème ILP suivant :

- (a) soit $\mathcal{T}^\perp = \{t_k \in \mathcal{T}_{a_i}^\tau : b_{t_k} = \perp\}$;
- (b) $\forall t_k \in \mathcal{T}^\perp$, une variable $v_k \in \{0, 1\}$ est générée ;
- (c) les contraintes sont définies comme suit :
 - contraintes d'exclusion : $\forall t_k, t_{k'} \in \mathcal{T}^\perp$ telles que $t_k \bowtie t_{k'}$, $v_k + v_{k'} \leq 1$;
 - contrainte de mémoire : $\sum v_k \leq M$.
- (d) on cherche les $\{v_k\}$ qui maximisent la fonction objectif suivante : $\sum \frac{v_k}{prio(t_k)}$.

2. a_i planifiant à la date τ tient compte de ses intentions à la date $(\tau - 1)$. En effet, un engagement (\square) ne pouvant être remis en cause, les tâches associées doivent être toujours planifiées. De même, un désistement fort ($\square \neg$) retire la tâche associée des tâches pouvant être planifiées. Formellement, deux contraintes viennent s'ajouter au problème :

Définition 2.15 (Planification : intégration des intentions)

Soit un problème ILP sur un ensemble de variables $\{v_k\}$ permettant à un agent $a_i \in \mathcal{A}$ de planifier à la date $\tau \in \mathbb{T}$. Les contraintes suivantes doivent être ajoutées au problème :

- $\forall t_k \in \mathcal{T}^\perp$ telle que $mod(\mathcal{I}_{t_k}^{a_i}) = \square$, $v_k = 1$;
- $\forall t_k \in \mathcal{T}^\perp$ telle que $mod(\mathcal{I}_t^{a_i}) = \square \neg$, $v_k = 0$.

3. après résolution du problème ILP, chaque v_k prenant une valeur de 1 représente une tâche t_k planifiée et chaque v_k ayant une valeur de 0 représente une tâche non planifiée. Le plan d'un agent a_i à la date τ modifie alors ses intentions. Chaque nouvelle tâche planifiée génère une proposition (\diamond) et chaque nouvelle tâche non planifiée génère un désistement faible ($\diamond \neg$). Formellement,

Définition 2.16 (Planification : génération du plan)

Soit un problème ILP sur un ensemble de variables $\{v_k\}$ permettant à un agent $a_i \in \mathcal{A}$ de planifier à la date $\tau \in \mathbb{T}$. Après résolution :

- $\forall v_k : v_k = 1$ et $\text{mod}(\mathcal{I}_{t_k}^{a_i}) \notin \{\square, \square\neg, \diamond\}$, $\text{mod}(\mathcal{I}_{t_k}^{a_i}) \leftarrow \diamond$;
- $\forall v_k : v_k = 0$ et $\text{mod}(\mathcal{I}_{t_k}^{a_i}) \notin \{\square, \square\neg, \diamond\neg\}$, $\text{mod}(\mathcal{I}_{t_k}^{a_i}) \leftarrow \diamond\neg$

En général, a_i ne dispose que des ressources M suffisantes à la réalisation d'un sous-ensemble de $\mathcal{I}_{a_i}^\tau$. Nous pouvons aussi remarquer que les engagements (\square) et les désistements forts ($\square\neg$) ne sont pas générés par le processus de planification individuel. En effet, un engagement n'a de sens que relativement à un collectif. Un agent seul n'a pas à s'engager. C'est pour cette raison que la planification individuelle ne produit pas d'engagements ou de désistements forts.

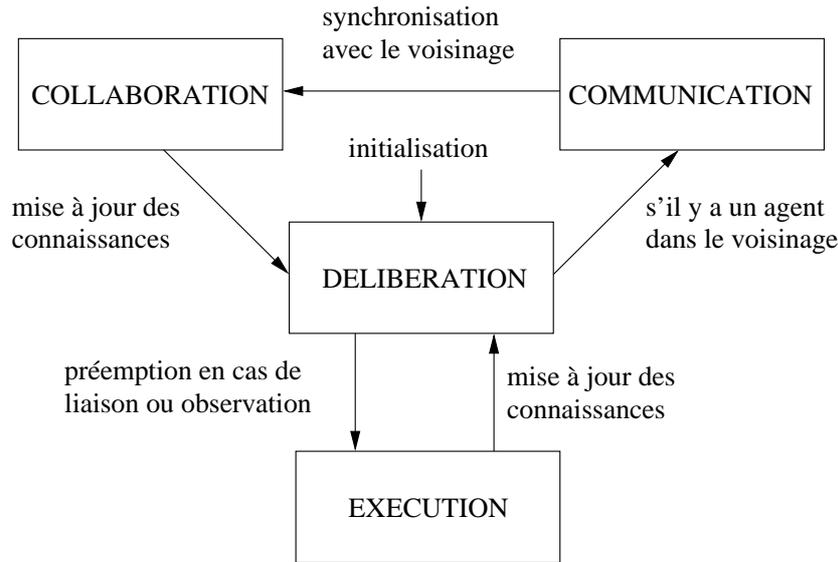
Contrôleur d'exécution

FIGURE 2.7 – Architecture de contrôle d'un agent-satellite

Afin de garantir l'autonomie des agents (comme nous l'avons précisé au Chapitre 1 Section 4), une architecture de contrôle pour chaque agent doit être définie. Représentés sur la Figure 2.7, quatre modules distincts assurent le contrôle de l'agent :

1. *délibération* permet à l'agent de planifier ;
2. *exécution* contrôle les liaisons avec le sol ou la réalisation des tâches atomiques ;
3. *communication* assure la synchronisation avec les autres agents ;
4. *collaboration* applique les algorithmes de coopération.

Initialement, un agent lance un processus de délibération (planification) qui met à jour sa base d'intentions. L'agent passe ensuite dans un état d'attente jusqu'à ce que le module de communication ou d'exécution prenne le relais (avec une priorité pour le module d'exécution).

Si une communication entre agents est possible, le module de communication va mettre en œuvre un protocole de communication spécifique à l'échange synchronisé de messages entre les agents. Suite à cet échange, le module de collaboration prend le relais, applique les algorithmes de coopération (Chapitre 4) puis rend la main au module de délibération qui relance un processus de planification, affiné par la collaboration.

Le module d'exécution prend le contrôle dès qu'une liaison avec le sol est disponible ou qu'une tâche qui a été planifiée doit être réalisée. Il rend ensuite la main au module de délibération mais peut de nouveau préempter et reprendre le contrôle.

Pour résumer, chaque agent est défini par deux boucles d'exécution entremêlées, une boucle (*re*)*planification - communication - collaboration* et une boucle *liaison au sol - observation* prioritaire. Le principe de planification que nous proposons est donc le suivant : les agents débutent une première étape de planification individuelle suivie par une phase de collaboration lorsqu'ils communiquent. Le processus est ensuite relancé, planification puis collaboration, afin d'atteindre un point fixe. Il s'agit donc de collaboration *durant* une planification incrémentale.

3.4 Vers une connaissance commune des artefacts

Ainsi que nous l'avons vu dans les paragraphes précédents, chaque agent dispose d'une connaissance locale en termes de tâches et d'intentions. Ce sont les artefacts de notre système, les éléments que les agents vont manipuler et que nous appellerons dans toute la suite connaissance. Formellement,

Définition 2.17 (Connaissance)

Une connaissance $K_{a_i}^\tau$ d'un agent a_i à la date $\tau \in \mathbb{T}$ est un triplet $\langle D_{K_{a_i}^\tau}, A_{K_{a_i}^\tau}, \tau_{K_{a_i}^\tau} \rangle$ tel que :

- $D_{K_{a_i}^\tau}$ est une tâche atomique t ou une intention $I_t^{a_k}$ de a_k à propos de t , $a_k \in \mathcal{A}$;
- $A_{K_{a_i}^\tau} \subseteq \mathcal{A}$ est le sous-ensemble des agents connaissant $K_{a_i}^\tau$;
- $\tau_{K_{a_i}^\tau} \in \mathbb{T}$ est la date de création ou de mise à jour de $D_{K_{a_i}^\tau}$.

Dans toute la suite, nous noterons $\mathcal{K}_{a_i}^\tau$ l'ensemble des connaissances de l'agent a_i à la date τ .

Dans ce chapitre, nous proposons de modéliser une constellation de satellites comme un réseau d'accointances publiques formant une congrégation où chaque agent est autonome en termes de planification individuelle. Les connaissances des agents définies comme des ensembles de tâches à réaliser et des ensembles d'intentions sont les artefacts du système. Afin de pouvoir coopérer et raisonner sur ces artefacts, les agents doivent en premier lieu en construire une connaissance commune. Intuitivement, la connaissance des artefacts va circuler en fonction du modèle d'accointances du système, c'est-à-dire la relation de voisinage.

Chapitre 3

Communiquer pour collaborer

Nous avons défini au chapitre précédent une constellation de satellites comme un système multiagent ayant un modèle d'accointances périodique et connexe. Dans ce chapitre, nous présentons dans un premier temps différents mécanismes de communication servant à diffuser de l'information puis nous présentons la notion de confiance. Ceci nous permettra de définir un protocole tirant parti des propriétés du modèle d'accointances afin de construire une connaissance commune fiable.

1 Diffuser les informations

Dans notre application, les agents ne peuvent coopérer que si leurs choix relatifs aux tâches à exécuter sont conditionnés par les choix des autres agents. Il est donc nécessaire que les agents puissent raisonner sur une connaissance commune en termes de tâches et d'intentions. Cela passe par la mise en place d'un protocole de communication efficace afin que chaque agent puisse savoir quel agent sait quelle information et ainsi construire à partir d'une connaissance distribuée (les connaissances individuelles) une connaissance commune (connaissance sur les connaissances d'autrui).

1.1 Taxonomie des communications

La communication permet aux agents de diffuser de l'information et, par conséquent de construire une connaissance commune. Cette communication peut être implicite ou explicite. Plus précisément,

- la communication implicite se réfère aux mécanismes où l'environnement sert à transmettre de l'information. Il peut s'agir d'une observation directe des actions des autres agents ou bien du dépôt d'information dans l'environnement (mécanisme de stigmergie [Grassé, 1959] ou d'étiquetage [Holland, 1993, Griffiths, 2008]);

Classification	Medium	Contenu	Fréquence	Synchronisation
Implicite	environnement	protocolaire ou cognitif	continue	non
Broadcast	centralisé	tout type	unique ou continue	non
Multicast	décentralisé	tout type	unique	oui
Point-à-point	décentralisé	tout type	unique	oui

FIGURE 3.1 – Taxonomie des mécanismes de communication

- la communication explicite se réfère aux mécanismes où les agents s’envoient et traitent des messages. Si l’on considère un environnement comprenant n agents, trois formes générales de communication explicite dépendant d’un type de liaison au sein d’un système peuvent être dégagées : on appelle *broadcast* une communication n à n , *multicast* une communication k à k où $2 < k < n$ et *point à point* une communication 2 à 2.

À partir des travaux de [Sklar *et al.*, 2006], nous pouvons caractériser les communications par :

1. le *medium*, qui indique la manière dont un message est transmis à un autre agent : l’environnement pour les communications implicites ou, pour les communications explicites, une manière centralisée (tableau noir, bus de communication, mécanisme de publication et de souscription) ou décentralisée (ensemble de communications point-à-point) ;
2. le *contenu* qui correspond au type de données que contient le message et dont [Balch et Arkin, 1994], s’inspirant de la théorie des actes de langage de [Searle, 1969], ont défini trois niveaux d’information de richesse sémantique croissante :
 - (a) un niveau protocolaire où les agents ne s’intéressent pas au contenu sémantique du message (qui peut par ailleurs être vide) mais à la présence du message lui-même (au sein d’un jeu de dialogue ou non) ;
 - (b) un niveau cognitif où une information d’état (qui n’est pas nécessairement intentionnelle) est transmise, permettant ainsi à un agent de connaître l’état interne (total ou partiel) d’un autre agent ;
 - (c) un niveau social où l’émetteur transmet intentionnellement un but (éventuellement enrichi d’arguments, de modalités ou d’assertions) pour accroître les connaissances des autres agents.
3. la *fréquence*, qui indique combien de fois les messages doivent être échangés, par exemple une unique fois ou bien en continu ;
4. la *synchronisation*, qui représente l’ordre d’échange des messages, si les agents doivent attendre un accusé de réception, s’ils communiquent à tour de rôle, etc.

Le tableau de la Figure 3.1 résume les caractéristiques des différents mécanismes de communication. Nous pouvons alors nous demander quel type de communication est le plus adapté pour diffuser de l'information, tant au niveau du contenu que du *medium*.

En termes de contenu, plus la communication protocolaire fournit de l'information moins une communication d'états ou de buts va accroître les performances. Ceci peut être mis en évidence par les travaux de [Balch et Arkin, 1994] sur des problèmes de fourragement, de consommation de type SUGARSCAPE¹ et de patrouille à l'aide d'agents réactifs qui interagissent traditionnellement de façon implicite et protocolaire. Dans le cadre du fourragement, une communication d'états augmente les performances de 16 % et une communication de buts de 19 %. Pour la consommation, une communication d'états augmente les performances de 10 % et une communication de buts de 6 %. Enfin pour la couverture, une communication d'états augmente les performances de 1 % tandis qu'une communication de buts ne les améliore pas.

Cependant ces redondances d'information peuvent aussi être complémentaires. Par exemple, dans les travaux de [Roth *et al.*, 2003] consacrés à RoboCup, les agents communiquent leurs perceptions locales en *broadcast*. Ces informations sont fusionnées à l'aide d'un filtre de Kalman² en un modèle global mais chaque agent conserve aussi son modèle local. Dans ce cadre, la position d'un agent est toujours donnée par ses propres capteurs (observation directe), la position des équipiers est toujours donnée par le modèle global puisque chaque équipier connaît précisément sa propre position (communication explicite d'état), la position des adversaires est tirée du modèle local. Cependant les images sont très souvent bruitées et l'agent, en connaissant la position de ses équipiers, peut en déduire celle des adversaires dans son champ de vision (observation directe et communication implicite). Le mécanisme est identique lorsqu'il s'agit de calculer la position de la balle.

Si le contenu de la communication est dépendant du type d'information à diffuser, le *medium* est dépendant de la structure organisationnelle du système. Dans ce manuscrit, nous nous intéressons à des constellations de satellites communicants par ISL. Dans un tel cadre, les agents ne peuvent percevoir de l'environnement que l'effet de leurs propres actions. Il ne peut pas y avoir de communication implicite. De même, en raison des distances entre les satellites, il ne peut y avoir de *broadcast*. Nous ne considérerons donc que la communication *multicast* (qui est une généralisation de la communication *point à point* et nous nous intéresserons à deux types de protocoles efficaces respectivement pour diffuser de l'information et construire une connaissance commune.

1. Il s'agit d'un environnement de simulation multiagent proposé par [Epstein et Axtell, 1996] où des agents doivent fourrager et consommer pour survivre une unique ressource (du sucre) distribuée aléatoirement sur une grille.

2. Afin de prendre en compte l'incertitude, chaque information transmise par les robots est datée. Une information ancienne aura moins de poids qu'une information récente, évitant ainsi qu'une erreur se propage dans le temps.

1.2 Protocoles épidémiques

Les protocoles fondés sur la rumeur (**Gossip Protocols**) comme ceux de [Gupta *et al.*, 2002] et [Jenkins *et al.*, 2001], aussi appelés protocoles épidémiques, sont utilisés pour partager des processus en *multicast*. Chaque agent du groupe sélectionne à intervalle de temps régulier un certain nombre d'agents pour partager de l'information. Les implantations du protocole diffèrent dans la longueur du tour de discussion et le poids des agents lors de la sélection. Par exemple, le *flat gossiping* sélectionne aléatoirement les agents avec une distribution de probabilités uniforme. Nous pouvons aussi considérer le *newcast* de [Buzing *et al.*, 2004, Eiben *et al.*, 2005] qui part d'une distribution uniforme puis incite les agents à ne communiquer par la suite qu'avec les agents qui ont interagi avec eux, formant ainsi incrémentalement des congrégations.

De manière générale, le protocole épidémique fonctionne comme suit :

1. soit un ensemble \mathcal{A} d'agents tel que $\mathcal{A} = \{a_1 \dots a_n\}$;
2. à chaque tour, tout agent a_i sélectionne k agents de \mathcal{A} et leur communique en *multicast* les informations à partager ;
3. il a été prouvé par [Pittel, 1987] que, dans un ensemble de n agents où un seul dispose d'une information donnée, un protocole épidémique a besoin de $\mathcal{O}(\log n)$ tours de discussion pour entièrement propager l'information.

L'avantage des protocoles épidémiques est d'être des protocoles *multicast* asynchrones, ce qui est utile lorsque la communication est coûteuse ou lente. Par exemple, nous pouvons nous intéresser aux problèmes de bases de données distribuées : traditionnellement, une opération de lecture/écriture synchrone sur plusieurs bases doit être lancée avant de faire une modification locale. [Holliday *et al.*, 2000] ont comparé cette approche avec une propagation des modifications de manière épidémique, ce qui entraîne une diminution du temps de réponse de 31 %. Notons qu'une estampille temporelle doit être ajoutée aux messages de mise à jour, pour ne tenir compte que des dernières modifications.

1.3 Protocoles en écoute flottante

Ce protocole de communication proposé par [Legras, 2003] s'intéresse à un principe de communication *multicast* se rapprochant du protocole Ethernet. On appelle **écoute flottante**, la capacité d'un agent à entendre un message sans que l'émetteur le sache ou ait l'intention qu'on l'entende. L'utilisation de l'écoute flottante suppose alors deux choses. La première est que l'agent doit être capable de percevoir un signal qui ne lui est pas directement destiné et comprendre qu'il s'agit d'une communication. La seconde est, au-delà de savoir qu'un message a été émis, de pouvoir le décoder et obtenir l'information qu'il contient.

Au vu de ceci, [Legras, 2003] propose un **modèle en étoile** afin d'utiliser l'écoute flottante pour la collaboration. L'idée est d'émettre *à la ronde* des messages susceptibles d'être

interceptés par d'autres agents. Plutôt que de router les messages, on route l'information. Cela peut se faire sans protocole complexe par proposition spontanée d'information : a_i perçoit un message m de a_j qui ne lui est pas destiné et estime que l'information est inexacte, il émet alors un message m' corrigé. Comme a_i a reçu m , il y a de fortes chances que a_j reçoive m' .

De ce flot d'informations qui circule, les agents peuvent construire une croyance commune. Cependant, les informations doivent être datées pour prendre en compte d'éventuelles révisions. Cela pose le **problème des horloges locales**. En effet, la datation des informations suppose un *temps unique* entre tous les agents. En fonction de deux hypothèses, ce problème peut être résolu de la manière suivante :

1. En supposant qu'aucune horloge ne se trompe de plus de δ si le temps véritable est t , le temps peut être discrétisé en une série d'intervalles de longueur 2δ centré sur t et séparés entre eux par au moins 4δ ;
2. En supposant que les agents ont des horloges synchrones sans origine du temps commune et que le temps de communication entre deux agents est une constante δ , il est possible de supprimer la date d'émission T_i et de remplacer toutes les autres dates d (locales à l'émetteur) par $T_i - d$. Le récepteur en recevant le message à d' peut alors transcrire les dates d dans son propre temps local avec la valeur $d' - (\delta + T_i - d)$.

Les deux protocoles précédents servent à la diffusion d'une information. Pour le récepteur de cette information se poseront par la suite deux questions importantes :

1. comment s'assurer à un instant donné qu'une information est partagée par un même sous-ensemble d'agents ?
2. cette information est-elle fiable ?

2 Comment les satellites communiquent-ils ?

À partir des considérations vues aux paragraphes précédents, nous proposons un protocole épidémique fondé sur une notion d'écoute flottante afin de propager les connaissances (voir Définition 2.17) de chaque agent au sein de la constellation de satellites. Nous définirons ensuite à la Section 4 la connaissance commune et la confiance associée en fonction des capacités de communication des agents.

2.1 Une définition en fonction des accointances

Dans notre application, deux agents peuvent communiquer à l'aide d'une liaison intersatellite lorsqu'ils se rencontrent. Par conséquent, la structure de la relation Voisins va déterminer l'ensemble des communications possibles. Nous pouvons donc définir, comme l'illustre la Figure 3.2, deux types de communications : une *communication directe* où deux agents se rencontrent et une *communication indirecte* où il existe une **chaîne de communications directes** d'un agent vers un autre. Formellement,

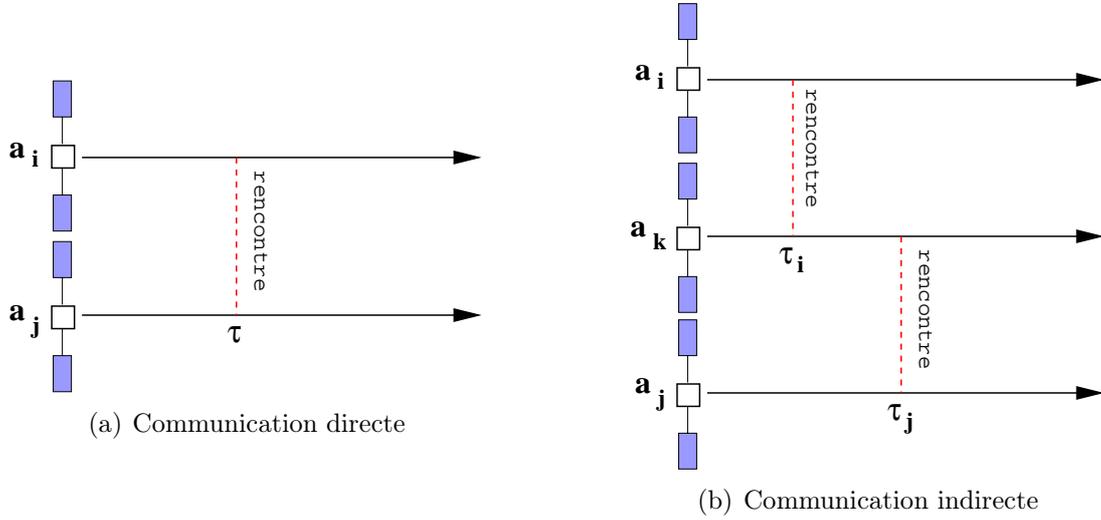


FIGURE 3.2 – Représentation des deux types de communications

Définition 3.1 (Communication)

Soit $\mathcal{S} = \langle \mathcal{A}, \mathbb{T}, \text{Voisins} \rangle$ une constellation et $a_i, a_j \in \mathcal{A}$ deux agents. a_i communique avec a_j à $(\tau_i, \tau_j) \in \mathbb{T}^2$ avec τ_i la date d'émission de a_i et τ_j la date de réception de a_j :

- directement si et seulement si $\exists \tau_i \in \mathbb{T}_P$ tel que $a_j \in \text{Voisins}(a_i, \tau_i)$. Dans ce cas, $\tau_i = \tau_j$;
- indirectement si et seulement si $\exists l \in \mathbb{N}^*$ tel que $\exists \{(a_{\tau_k}, \tau_k) \in \mathcal{A} \times \mathbb{T}, k \in [0 \dots l]\}$ avec :
 1. $a_{\tau_0} \in \text{Voisins}(a_i, \tau_i)$;
 2. $a_{\tau_{k+1}} \in \text{Voisins}(a_{\tau_k}, \tau_k)$ et $\tau_i < \tau_k < \tau_{k+1} < \tau_j$;
 3. $a_j \in \text{Voisins}(a_{\tau_l}, \tau_j)$.

D'après la définition de la relation de voisinage qui est symétrique et non transitive, si l'agent a_i peut communiquer directement avec a_j alors a_j peut communiquer directement avec a_i ; que a_i puisse communiquer indirectement avec a_j n'implique pas que a_j puisse communiquer indirectement avec a_i .

2.2 Déplier la relation de voisinage

À partir de la définition formelle d'une communication, nous pouvons représenter la relation Voisins comme un graphe multivalué \mathcal{V} connu par chaque agent (puisque le modèle d'accointance est une connaissance publique de la constellation). Formellement,

Définition 3.2 (Graphe de voisinage)

Soit $\mathcal{S} = \langle \mathcal{A}, \mathbb{T}, \text{Voisins} \rangle$ une constellation. Le graphe de voisinage \mathcal{V} est défini par $\mathcal{V} = (\mathcal{A}, \{(a_i, a_j)\}, \{\{v_{ij}\}\})$ où :

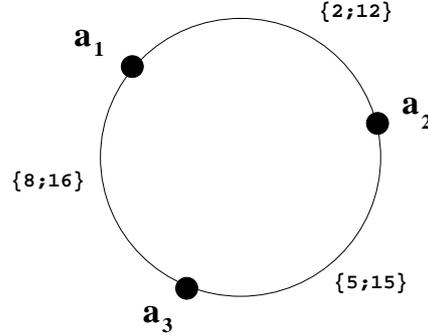


FIGURE 3.3 – Graphe de voisinage pour l'Exemple 3.3

- l'ensemble des sommets de \mathcal{V} est \mathcal{A} ;
- l'arête (a_i, a_j) existe si et seulement si $\exists \tau \in \mathbb{T}_P, a_j \in \text{Voisins}(a_i, \tau)$;
- chaque arête (a_i, a_j) est valuée par l'ensemble $v_{ij} = \{\tau \in \mathbb{T}_P : a_j \in \text{Voisins}(a_i, \tau)\}$.

Illustrons ceci sur un exemple.

Exemple 3.3

Soit a_1, a_2, a_3 trois agents. Supposons que *Voisins* est défini ainsi sur la période $P = 20$:

$$\left\{ \begin{array}{l} \text{Voisins}(a_1, 2) = \{a_2\}, \text{Voisins}(a_1, 12) = \{a_2\} \\ \text{Voisins}(a_2, 5) = \{a_3\}, \text{Voisins}(a_2, 15) = \{a_3\} \\ \text{Voisins}(a_3, 8) = \{a_1\}, \text{Voisins}(a_3, 16) = \{a_1\} \end{array} \right.$$

Le graphe de voisinage est présenté en Figure 3.3.

Intuitivement, une communication indirecte d'un agent a_i vers un agent a_j est un chemin depuis le sommet a_i jusqu'au sommet a_j et une communication directe est un cas particulier de communication indirecte. Nous pouvons alors doter chaque agent d'algorithmes pour calculer ces communications.

À partir du graphe multivalué, un graphe valué en fonction de la date courante est extrait. Pour calculer la prochaine communication entre deux agents, il suffit de calculer le chemin de poids minimal entre les deux sommets correspondants, où le poids représente le temps. Ce graphe valué est construit au fur et à mesure de l'exploration. Pour cela, nous proposons une variante d'un algorithme de Dijkstra (voir Algorithme 1) où :

1. la date courante t_i (la date initiale plus le poids du chemin courant) est associée au sommet a_i ;
2. le poids de chaque arête (a_i, a_j) est calculé comme suit : $\min v_{ij} - t_i \pmod{P}$.

Algorithme 1 : Calculer la prochaine communication entre deux agents**Données** : a_i : source ; a_j : destinataire ; τ^* : date initiale**Résultat** : (τ_i, τ_j) : prochaine communication entre a_i et a_j

```

1  début
2  |  pour  $a_k \in \mathcal{A}$  faire
3  |  |  pour  $a_l \in \mathcal{A}$  faire
4  |  |  |   $v(a_k, a_l) \leftarrow \emptyset$  ;           /* construction de l'arête  $(a_k, a_l)$  */
5  |  |  |  si  $a_k \neq a_l$  alors
6  |  |  |  |   $v(a_k, a_l) \leftarrow \{\tau : a_l \in \text{Voisins}(a_k, \tau)\}$ 
7  |  |  |  fin
8  |  |  fin
9  |  |   $d(a_k) \leftarrow \infty$  ;           /* distance courante entre  $a_i$  et  $a_k$  */
10 |  |   $p(a_k) \leftarrow a_i$  ;           /* sommet précédent de  $a_k$  */
11 |  |   $\tau(a_k) \leftarrow \tau^*$  ;       /* date courante au sommet  $a_k$  */
12 |  fin
13 |   $d(a_i) \leftarrow 0$ 
14 |   $\mathcal{V} \leftarrow \mathcal{A} \setminus \{a_i\}$  tant que  $\mathcal{V} \neq \emptyset$  faire
15 |  |   $a_k \leftarrow \arg \min_{a \in \mathcal{A}} d(a)$ 
16 |  |   $\mathcal{V} \leftarrow \mathcal{V} \setminus \{a_k\}$ 
17 |  |  pour  $a_l \in \mathcal{A}$  faire
18 |  |  |  si  $v(a_k, a_l) \neq \emptyset$  alors
19 |  |  |  |   $v^* \leftarrow \min v(a_k, a_l) - \tau(a_k) \pmod{P}$  ;           /* poids courant */
20 |  |  |  |  si  $d(a_k) > d(a_l) + v^*$  alors
21 |  |  |  |  |   $d(a_k) \leftarrow d(a_l) + v^*$ 
22 |  |  |  |  |   $p(a_k) \leftarrow a_l$ 
23 |  |  |  |  |   $\tau(a_k) \leftarrow \tau^* + d(a_k)$ 
24 |  |  |  |  fin
25 |  |  |  fin
26 |  |  fin
27 |  fin
28 |   $a_k \leftarrow a_j$ 
29 |   $\tau_j \leftarrow \tau(a_k)$ 
30 |  tant que  $a_k \neq a_i$  faire
31 |  |  si  $p(a_k) = a_i$  alors
32 |  |  |   $\tau_i \leftarrow \tau(a_k)$ 
33 |  |  fin
34 |  |   $a_k \leftarrow p(a_k)$ 
35 |  fin
36 |  retourner  $(\tau_i, \tau_j)$ 
37 fin

```

Exemple 3.4

Reprenons l'Exemple 3.3 et appliquons l'Algorithme 1 afin de calculer à la date 1 la prochaine communication de a_1 vers a_3 .

1. Considérons les arêtes à partir du sommet a_1 , (a_1, a_2) et (a_1, a_3) . Le poids des arêtes (a_1, a_2) et (a_1, a_3) sont respectivement $(\min(2-1 \pmod{20}, 12-1 \pmod{20}))$, c'est-à-dire 1, et $(\min(8-1 \pmod{20}, 16-1 \pmod{20}))$, c'est-à-dire 7. Les dates courantes alors mémorisées dans les sommets a_2 et a_3 sont respectivement 2 et 8 ;
2. Comme un chemin de a_1 à a_3 a été calculé via l'arête (a_1, a_3) , une première solution a été trouvée : une communication directe à $(8, 8)$.
3. Continuons l'exploration à partir du sommet a_2 et considérons l'arête (a_2, a_3) . Son poids est calculé comme étant $(\min(5-2 \pmod{20}, 15-2 \pmod{20}))$, c'est-à-dire 3 et la date courante mémorisée pour le sommet a_3 est 5. Un nouveau chemin de a_1 à a_3 a été calculé via les arêtes (a_1, a_2) et (a_2, a_3) . Une meilleure solution a été trouvée : une communication indirecte à $(2, 5)$.

Cette structure de graphe permet de calculer ou de vérifier d'autres propriétés comme, par exemple, le nombre de fois qu'un agent communique directement entre deux dates, la date de dernière communication entre deux agents ou si un agent a communiqué avec un autre dans un intervalle de temps donné.

2.3 Un protocole épidémique pour diffuser l'information

Afin de propager les connaissances individuelles de chaque agent, nous définissons un protocole épidémique où chaque agent utilise chaque occasion de communication pour transmettre des connaissances, même si ces dernières ne concernent pas directement les interlocuteurs. Informellement, chaque agent note tous les changements de connaissance et les propage à ses voisins qui mettent à jour leurs connaissances en se fondant sur l'estampille temporelle. Ces voisins réitèrent alors le processus.

Pour éviter de surcharger le système en messages inutiles, un agent a_i ne communique pas une connaissance si le récepteur la connaît déjà, c'est-à-dire s'il appartient à l'ensemble $A_{K_{a_i}^\tau}$ des agents connaissant $K_{a_i}^\tau$.

Formellement,

Protocole 3.5 (Diffusion des connaissances)

Soit un agent $a_i \in \mathcal{A}$. $\forall \tau \in \mathbb{T}$:

— $\forall a_j \in \text{Voisins}(a_i, \tau)$, a_i exécute :

1. $\forall K_{a_i}^\tau \in \mathcal{K}_{a_i}^\tau$ tel que $a_j \notin A_{K_{a_i}^\tau}$:

(a) a_i transmet $K_{a_i}^\tau$ à a_j ;

(b) a_i met à jour $A_{K_{a_i}^\tau}$ si a_j accuse réception de $K_{a_i}^\tau$.

- $\forall K_{a_i}^\tau \in \mathcal{K}_{a_i}^\tau$ reçue par a_j à la date τ :
1. a_j met à jour $\mathcal{K}_{a_j}^\tau$ avec $K_{a_i}^\tau$;
 2. a_j accuse réception de $\mathcal{K}_{a_i}^\tau$ à a_i .

Deux types de mises à jour existent pour un agent : une mise à jour interne de $K_{a_i}^\tau$ réalisée par a_i et une mise à jour externe de $K_{a_i}^\tau$ réalisée par a_j lorsque ce dernier la reçoit.

Protocole 3.6 (Mise à jour interne)

Soit une connaissance $K_{a_i}^\tau$. La mise à jour interne de $K_{a_i}^\tau$ par l'agent a_i à la date τ se fait en deux occasions :

1. si $D_{K_{a_i}^\tau}$ est une intention $I_t^{a_i}$ de a_i , la mise à jour se fait lorsque $\text{mod}(I_t^{a_i})$ est révisée ;
2. si $D_{K_{a_i}^\tau}$ est une tâche t , la mise à jour se fait lorsque t est réalisée par a_i .

Dans les deux cas, $\tau_{K_{a_i}^\tau} \leftarrow \tau$ et $A_{K_{a_i}^\tau} \leftarrow \{a_i\}$.

Protocole 3.7 (Mise à jour externe)

Soit une connaissance $K_{a_i}^\tau$ reçue par un agent a_j à la date τ . La mise à jour externe de $K_{a_i}^\tau$ par a_j s'effectue ainsi :

1. si $K_{a_i}^\tau$ est déjà connue de a_j , $\exists K_{a_j}^\tau$ telle que $D_{K_{a_i}^\tau} = D_{K_{a_j}^\tau}$:
 - (a) si $\tau_{K_{a_i}^\tau} > \tau_{K_{a_j}^\tau}$, la connaissance de a_j est obsolète. Il la remplace par $K_{a_i}^\tau$ et s'ajoute à l'ensemble $A_{K_{a_i}^\tau}$.
 - (b) si $\tau_{K_{a_i}^\tau} = \tau_{K_{a_j}^\tau}$, seuls les agents connaissant cette information peuvent avoir changé. Par conséquent, $A_{K_{a_i}^\tau} = A_{K_{a_j}^\tau} \cup A_{K_{a_i}^\tau}$.
2. si $K_{a_i}^\tau$ n'est pas connue de a_j , a_j l'ajoute simplement à son ensemble $\mathcal{K}_{a_j}^\tau$, et s'ajoute à l'ensemble $A_{K_{a_i}^\tau}$.

2.4 Propriétés des communications

Nous pouvons alors nous intéresser à trois propriétés. La complexité en termes de messages échangés, c'est-à-dire le nombre de connaissances communiquées, est importante car les agents ne peuvent communiquer que durant un intervalle de temps faible. Il est aussi intéressant de s'assurer qu'une connaissance donnée sera bel et bien propagée à l'ensemble de la constellation après un laps de temps donné. Enfin, nous verrons qu'ajouter de nouveaux agents permet d'accélérer le processus de communication.

Complexité du protocole

Une communication directe de a_i vers a_j suit le protocole de diffusion des connaissances (Protocole 3.5) et la mise à jour qui en résulte (Protocoles 3.6 et 3.7).

Proposition 3.8 (Complexité d'une communication directe)

La complexité en termes de connaissances d'une communication directe entre a_i et a_j à $(\tau, \tau) \in \mathbb{T}^2$ est pour l'agent a_i en $\mathcal{O}(|\mathcal{T}_{a_i}^\tau| \cdot |\mathcal{T}_{a_j}^\tau| \cdot |\mathcal{A}|^2)$

Démonstration 3.8

Au pire cas lors d'une communication directe, un agent a_k envoie $|\mathcal{T}_{a_k}^\tau|$ connaissances concernant les tâches et $|\mathcal{T}_{a_k}^\tau| \cdot |\mathcal{A}|$ connaissances concernant les intentions (une intention par tâche et par agent). Soit a_i et a_j deux agents tels que a_i communique directement vers a_j et a_j vers a_i à $(\tau, \tau) \in \mathbb{T}^2$. Pour l'agent a_i , la complexité en termes de connaissances est :

$$\mathcal{O}\left(\underbrace{|\mathcal{T}_{a_i}^\tau| + |\mathcal{T}_{a_i}^\tau| \cdot |\mathcal{A}|}_{\text{connaissances émises}} + \underbrace{|\mathcal{T}_{a_j}^\tau| + |\mathcal{T}_{a_j}^\tau| \cdot |\mathcal{A}|}_{\text{connaissances reçues}}\right)$$

Pour chaque connaissance reçue, l'agent a_i passe en revue sa base de connaissance : $|\mathcal{T}_{a_i}^\tau|$ connaissances pour chaque tâche reçue et $|\mathcal{T}_{a_i}^\tau| \cdot |\mathcal{A}|$ connaissances pour chaque intention reçue. Ainsi, la complexité de la mise à jour est :

$$\mathcal{O}(|\mathcal{T}_{a_j}^\tau| \cdot |\mathcal{T}_{a_i}^\tau| + |\mathcal{T}_{a_j}^\tau| \cdot |\mathcal{T}_{a_i}^\tau| \cdot |\mathcal{A}|^2)$$

Donc, la complexité d'une communication directe est :

$$\mathcal{O}(|\mathcal{T}_{a_i}^\tau| + |\mathcal{T}_{a_i}^\tau| \cdot |\mathcal{A}| + |\mathcal{T}_{a_j}^\tau| \cdot |\mathcal{T}_{a_i}^\tau| + |\mathcal{T}_{a_j}^\tau| \cdot |\mathcal{T}_{a_i}^\tau| \cdot |\mathcal{A}|^2)$$

Plus simplement :

$$\mathcal{O}(|\mathcal{T}_{a_i}^\tau| \cdot |\mathcal{T}_{a_j}^\tau| \cdot |\mathcal{A}|^2)$$

Propagation des connaissances

La vitesse de propagation d'une connaissance est fonction de la structure de la relation Voisins.

Proposition 3.9 (Vitesse de propagation)

Soit une constellation \mathcal{S} , un agent $a_i \in \mathcal{A}$ et une connaissance $K_{a_i}^\tau$. Cette connaissance est toujours propagée à l'ensemble des agents de \mathcal{A} après, au plus, une durée de P , la période de Voisins.

Démonstration 3.9

Soit une constellation \mathcal{S} et un agent $a_i \in \mathcal{A}$. Soit une connaissance $K_{a_i}^\tau$ de l'agent a_i à la date τ . Puisque Voisins satisfait la Définition 2.5 de la connexité, $\forall a_j \in \mathcal{A}, \exists (\tau_i, \tau_j) \in \mathbb{T}_P^2$ où a_i peut communiquer (directement ou non) avec a_j .

1. si $\tau_i > \tau$ alors $K_{a_i}^\tau$ arrive au pire cas à a_j à la date τ_j .
2. si $\tau_i < \tau$ alors $K_{a_i}^\tau$ arrive au pire cas à a_j à la date $(\tau_j + P)$.

Dans les deux cas, la connaissance a été propagée après au plus une durée P .

Influence des nouveaux agents

On peut remarquer intuitivement que plus le nombre d'agents d'une constellation est grand, plus les communications sont régulières et rapprochées. Ainsi si le nombre d'agents tend vers l'infini, les agents tendront à communiquer en permanence.

Définissons la fréquence de communication d'un agent vers un autre.

Définition 3.10 (Fréquence de communication)

Soit deux agents a_i et a_j d'une constellation \mathcal{S} . La fréquence de communication entre a_i et a_j , noté $f_j^i(\mathbb{T}_P)$, est le nombre de fois où a_i peut communiquer (directement ou non) avec a_j sur l'intervalle \mathbb{T}_P .

Proposition 3.11 (Ajout d'un nouvel agent)

Soit \mathcal{S} une constellation et un agent a_{n+1} tel que $\mathcal{A}_+ = \{\mathcal{A} \cup \{a_{n+1}\}\}$ forme une constellation \mathcal{S}_+ . Les fréquences de communication entre les agents de \mathcal{S}_+ sur l'intervalle \mathbb{T}_P sont supérieures ou égales à celles des agents de \mathcal{S} sur le même intervalle.

Démonstration 3.11

Soit l'ensemble d'agents $\mathcal{A} = \{a_1 \dots a_n\}$ d'une constellation \mathcal{S} et un agent a_{n+1} tel que $\mathcal{A}_+ = \{a_1 \dots a_{n+1}\}$ forme une constellation \mathcal{S}_+ . Notons Voisins et Voisins_+ les relations de voisinage respectivement associées à \mathcal{S} et \mathcal{S}_+ , et P et P_+ les périodes associées.

1. Par définition, $\forall a_i \in \mathcal{A}, \tau \in \mathbb{T}_P$ alors $\tau \in \mathbb{T}_{P_+}$ et $\text{Voisins}(a_i, \tau) \subseteq \text{Voisins}_+(a_i, \tau)$. Ainsi l'agent a_{n+1} ne retire aucune possibilité de communication. Les fréquences de communication de \mathcal{S}_+ sur \mathbb{T}_P ne peuvent pas être inférieures à celles de \mathcal{S} .
2. Exhibons un cas où les fréquences de communication de \mathcal{S}_+ sont supérieures à celles de \mathcal{S} . Soit a_i et $a_j \in \mathcal{A}_+$ ($i \neq j \neq (n+1)$) tels que $\exists \tau_1$ et $\tau_2 \in \mathbb{T}_{P_+}$ où $a_i \in \text{Voisins}(a_{n+1}, \tau_1)$ et $a_j \in \text{Voisins}(a_{n+1}, \tau_2)$. Dans ce cas, a_i peut communiquer avec a_j à la fréquence de \mathcal{S} et au moins une fois supplémentaire par l'intermédiaire de a_{n+1} .

Les fréquences de communication de \mathcal{S}_+ sont donc supérieures ou égales à celles de \mathcal{S} .

Ainsi, l'ajout d'un nouvel agent ne dégrade pas les occasions de communication de la constellation. Plus précisément, les fréquences de communication sont améliorées de la manière suivante :

Proposition 3.12 (Extension du modèle d'acointance)

Ajouter un agent a_{n+1} à une constellation \mathcal{S} augmente les fréquences de communication de celle-ci si et seulement si a_{n+1} participe à une communication indirecte.

Démonstration 3.12

Soit une constellation \mathcal{S} et un nouvel agent a_{n+1} . Nous pouvons distinguer deux cas :

1. si l'agent a_{n+1} ne participe qu'à des communications directes, $\exists a_i \in \mathcal{A}$ tel que $\exists \tau \in \mathbb{T}_P$ où $a_i \in \text{Voisins}(a_{n+1}, \tau)$. Si ce n'était pas le cas alors a_{n+1} communiquerait directement avec deux agents distincts et participerait à une communication indirecte entre eux deux. Ainsi cette unique rencontre sur la période de Voisins ne permet pas d'accroître le nombre de communications indirectes et donc la fréquence de communication ;
2. si l'agent a_{n+1} participe à au moins une communication indirecte, nous nous ramenons au cas 2 de la Preuve 3.11 et a_{n+1} augmente la fréquence de communication entre au moins deux agents a_i et a_j grâce à une communication indirecte supplémentaire.

Nous avons vu dans les paragraphes précédents de quelle manière les agents de notre application propageaient leurs connaissances en termes de tâches et d'intentions. Se pose alors la question de la représentation de la connaissance commune et celle, due aux délais lors des communications indirectes, de la confiance en une information susceptible d'être révisée.

3 Qu'est-ce que la confiance ?

Le développement de systèmes multiagents dont les membres doivent coopérer, comme les grilles de calcul, le commerce électronique, la toile sémantique ou les systèmes multiexperts, a conduit à l'étude de la notion de confiance. En effet, les agents doivent pouvoir modéliser les interactions qu'ils observent ou auxquelles ils participent pour juger si elles sont acceptables, c'est-à-dire si les autres agents se comportent correctement ou non et si les informations qu'ils en retirent sont fiables.

3.1 Une définition floue

Il n'y a pas de consensus sur la définition de la confiance. Par exemple, [Mayer *et al.*, 1995] définissent la confiance comme mesurant la volonté d'un agent a_i à se rendre vulnérable aux actions d'un agent a_j dans l'espoir que ce dernier accomplira en retour une action désirée par a_i . En revanche, [Azzedin et Maheswaran, 2003] considèrent la confiance comme une croyance dans la compétence d'un agent à agir d'une manière attendue, cette confiance n'étant pas caractérisée par une valeur fixe associée à l'agent mais par une valeur subjective applicable dans un contexte donné à un instant donné.

Confiance et méfiance sont couramment présentées comme complémentaires. Cependant, il n'est pas évident que l'une et l'autre soient strictement corrélées. Pour [Gans *et al.*, 2001], la confiance mesure le degré de coopération (et par conséquent le risque) qu'il est prêt à prendre tandis que la méfiance mesure le degré de contrôle qu'un agent pense nécessaire envers un autre. Ainsi un agent peut suffisamment faire confiance à un

autre pour coopérer mais être suffisamment méfiant pour désirer un contrôle sur l'exécution de cette coopération.

De manière générale, la confiance est une mesure (possiblement multidimensionnelle) interne à un agent sur un autre agent tandis que la réputation d'un agent est une agrégation des confiances des autres agents. La réputation est donc une donnée dynamique avec une évolution non monotone, graduée (il n'y a pas d'unité de mesure). De plus, comme la réputation est une donnée subjective propagée, elle est **asymétrique** et possède une propriété de **transitivité conditionnelle**.

3.2 Classification des modèles de confiance

Les deux aspects fondamentaux d'un modèle de confiance et de réputation sont donnés par [Josang et Ismail, 2002] : un *moteur* permettant de calculer une valeur de réputation d'un agent en fonction d'un certain nombre d'observations rapportées par les autres agents et un *mécanisme de propagation* qui permet aux agents d'obtenir une réputation lorsqu'ils en ont besoin. À cela [Sabater et Sierra, 2005] ajoutent différents critères permettant de classer les modèles : le modèle conceptuel, les sources d'information, la visibilité des mesures, la contextualisation, la malignité des agents impliqués et enfin la fiabilité que l'on concède au modèle.

Le **modèle conceptuel** peut être :

1. *cognitif* (C) : la confiance et la réputation, représentées par des valeurs discrètes, se fondent sur des croyances sous-jacentes et sont fonction des degrés de ces croyances ;
2. *théorie des jeux* (TJ) : la confiance et la réputation, représentées par des valeurs continues, sont des probabilités subjectives indiquant la probabilité pour l'agent a_i que l'agent a_j réalise une action donnée.

Les **sources d'information** indiquent les moyens par lesquels la confiance et la réputation se construisent. Elles sont dépendantes des capacités de perception des agents :

1. *expérience directe* (E) : les agents observent leur interaction avec les autres agents. Bien qu'il puisse y avoir un bruit sur l'observation, la source est considérée comme fiable ;
2. *témoignage* (T) : cette perception est fondée sur la transmission d'une expérience directe ou d'un autre témoignage par un agent tiers. Si ce type d'information est prépondérant, la question de la fiabilité de la source est centrale ;
3. *information sociologique* (IS) : le rôle joué par les autres agents donne une base à la confiance et la réputation. Cette information peut être fondée ou non (ne pas avoir confiance en un fonctionnaire, avoir confiance en ses parents). Ce type de source est peu traité dans la littérature ;

4. *étiquetage* (L) : une étiquette est attribuée à chaque agent en fonction de certaines propriétés. La confiance et la réputation se fondent alors sur l'analyse de ces étiquettes.

La **visibilité** se définit par deux caractéristiques :

1. *globale* ou *centralisée* (G) : chaque agent est doté d'une réputation globale et publique qui est mise à jour par les autres agents. Un grand nombre d'opinions différentes minimise le risque de non fiabilité des sources d'information. Cependant, cela nécessite que l'indicateur de confiance soit un agrégat acceptable par tous les agents ;
2. *subjective* ou *décentralisée* (S) : la confiance et la réputation d'un agent sont construites individuellement par chaque agent. Afin de tirer parti de cette multitude de sources, ce sont des modèles plutôt adaptés à des environnements de petite taille ou de taille moyenne où les agents communiquent fréquemment.

L'**indépendance au contexte** représente le fait qu'il n'y a qu'une valeur de réputation unique pour chaque agent (subjective ou globale). Dans le cas contraire, changer de contexte entraîne le calcul d'une nouvelle réputation. Par exemple, on peut avoir confiance dans un médecin lorsqu'il parle de médecine mais pas lorsqu'il parle d'informatique. Cette approche rejoint celle des sources d'information sociologiques.

La **malignité** des agents est aussi à prendre en compte :

1. *honnête* (H) : les comportements malveillants ne sont pas pris en compte. Le modèle se repose sur le grand nombre d'agents honnêtes afin de faire contrepoids aux quelques agents malveillants ;
2. *malhonnête* (MH) : le modèle suppose que les agents peuvent dissimuler ou biaiser des informations mais qu'ils ne mentent pas (par exemple des agents négociant qui ne révèlent pas leurs préférences) ;
3. *malveillant* (MV) : le modèle suppose que les agents peuvent non seulement dissimuler ou biaiser des informations mais surtout qu'ils sont capables de mentir (par exemple des agents négociants qui révèlent de fausses préférences pour induire les autres agents en erreur).

Enfin la dernière question est celle de savoir si la mesure de confiance et de réputation calculée est elle-même **fiable**.

3.3 Modèles et limites

Le tableau de la Figure 3.4 résume les caractéristiques de différents systèmes de confiance et de réputation que l'on trouve dans la littérature. Afin d'en faciliter la lecture, les lettres représentent les caractéristiques présentées au paragraphe précédent. Nous pouvons y distinguer trois approches principales :

Approche de	Modèle	Sources	Visibilité	Contexte	Malignité
[Marsh, 1994]	TJ	E	S	oui	H
[Castelfranchi et Falcone, 1998]	C	E + IS	G	non	H
[Zacharia et Maes, 2000]	C	T	G ou S	non	H
[Schillo <i>et al.</i> , 2000]	TJ	E + T	S	non	MH
[Abdul-Rahman et Hailes, 2000]	C	E + T	S	non	MH
[Esfandiari et Chandrasekharan, 2001]	TJ	E + T	S	non	H
[Sabater et Sierra, 2001]	C	E + T	G + S	non	H
[Josang et Ismail, 2002]	TJ	E + T	S	non	MH
[Yu et Singh, 2002]	C + TJ	E ou T	S	non	H
[Sen et Sajja, 2002]	C + TJ	T	S	non	MV
[Carbo <i>et al.</i> , 2002]	C	E + T	S	non	H
[Carter <i>et al.</i> , 2002]	C	T	G	oui	H
[Muller, 2006]	C	E + T	S	oui	MV

FIGURE 3.4 – Caractéristiques des différents modèles de confiance

1. les approches logiques [Castelfranchi et Falcone, 1998, Carter *et al.*, 2002, Muller, 2006] posent la confiance comme l'état mental derrière toute action de délégation ou toute satisfaction de normes. Ces approches utilisent des croyances sur les compétences des agents et sur les normes instanciées pour inférer des intentions persistantes de délégation, ce qui correspond à une confiance ;
2. les approches statistiques [Marsh, 1994, Abdul-Rahman et Hailes, 2000, Esfandiari et Chandrasekharan, 2001, Carbo *et al.*, 2002, Josang et Ismail, 2002, Sen et Sajja, 2002, Yu et Singh, 2002] agrègent les observations et les témoignages à l'aide de réseaux bayésiens, de fonctions beta, d'ensembles flous, de fonctions de croyance de Dempster-Shafer ou d'autres formalismes. Cela permet par la suite de définir un estimateur de la confiance en un autre agent par une densité de probabilités représentant la tendance à *bien* se comporter ;
3. les réseaux de confiance (**Trust Net**) [Schillo *et al.*, 2002, Sabater et Sierra, 2001, Zacharia et Maes, 2000] modélisent la confiance sous forme de graphes où les nœuds sont les sources d'information et les arcs les témoignages. La réputation (ou la malveillance) est ensuite modélisée par un processus de Bernoulli où la réputation d'un agent au niveau X est calculée récursivement à partir des témoignages du niveau $X - 1$.

Ces modèles ont cependant une limite importante : ils supposent que les agents peuvent communiquer, même au sein d'un réseau contraint d'acointances, afin de partager leurs évaluations de réputation. **Aucun des modèles présentés ne prend en compte les délais ou les impossibilités de communication, et par là même, ils négligent la dimension temporelle.** C'est pourquoi dans le paragraphe suivant, nous proposons

une définition de la confiance se fondant sur les capacités de communication et servant à construire une connaissance commune.

4 Construire une connaissance commune

Nous avons vu dans les paragraphes précédents de quelle manière les connaissances étaient propagées au sein de la constellation. Nous nous intéressons maintenant à la manière dont une connaissance commune peut être construite. Deux cas sont à distinguer :

1. en ce qui concerne les tâches, le mécanisme de mise à jour interne permet la connaissance commune sitôt que l'information sur la tâche a été diffusée à l'ensemble de la constellation. En effet, une tâche n'est mise à jour que lorsqu'elle est réalisée, ce qui n'arrive qu'une unique fois. Ainsi il y aura connaissance commune de la réalisation de la tâche après au plus une durée P (voir Proposition 3.9) ;
2. en ce qui concerne les intentions, la mise à jour se fait sur les changements de modalités qui ne forment aucun état absorbant puisque les intentions peuvent être révisées. Une situation de babillage³ est possible et nous proposons dans le paragraphe suivant une définition de la notion de connaissance commune pour les intentions.

4.1 Connaissances dérivées des accointances

En logique épistémique, une proposition φ est une **connaissance partagée** d'un groupe d'agents A si pour tout $a_i \in A$, $K_{a_i}(\varphi)$ (l'agent a_i sait φ). Plus formellement $E_A(\varphi) =_{def} \bigwedge_{a_i \in A} K_{a_i}(\varphi)$ (où \wedge est la conjonction). Donc $E_A(\varphi)$ peut se lire : φ est un fait connu de tout le groupe A , ou le groupe A sait que φ . La connaissance partagée peut s'itérer : ainsi $E_A^3(\varphi)$ signifie le groupe A sait que le groupe A sait que le groupe A sait que φ , tandis que $E_A^k(\varphi)$ signifie le groupe A sait que (répété k fois) le groupe A sait que φ . La **connaissance commune** est alors $C_A(\varphi) =_{def} \bigwedge_{k=0}^{\infty} E_A^k(\varphi)$.

Deux propriétés permettent aux agents de construire une connaissance partagée. La première, la *propagation des intentions*, indique qu'un agent a_i qui communique une intention à un agent a_j sait que a_j connaît cette intention. La seconde, la *propriété d'intentionnalité*, indique qu'un agent connaît toujours les intentions qu'il a générées. Formellement,

Définition 3.13 (Propagation d'intention)

À une date τ donnée, un agent a_i sait que l'agent a_j connaît $I_t^{a_i}$ si et seulement si a_i a communiqué avec a_j à (τ_i, τ_j) tel que $\tau_{K_{a_i}^{\tau}} \leq \tau_i$ et $\tau_j \leq \tau$, noté $K_{a_i}(K_{a_j}(I_t^{a_i}))$.

Définition 3.14 (Propriété d'intentionnalité)

À une date τ donnée, un agent a_i connaît toutes les intentions $I_t^{a_i}$ qu'il a générées à une date $\tau' \leq \tau$, noté $K_{a_i}(I_t^{a_i})$.

3. Un système babille lorsque les éléments qui le composent communiquent sans fin sans converger.

À l'aide du modèle d'accointances, un agent peut connaître l'ensemble des intentions qu'il connaît et qu'un autre agent connaît. La connaissance commune se définit alors grâce au fait que le modèle d'accointances est une connaissance publique.

Définition 3.15 (Connaissance commune)

Soit une intention $I_t^{a_k}$ et deux agents a_i et a_j . $I_t^{a_k}$ est une connaissance commune de a_i et a_j à la date τ si et seulement si a_k a communiqué avec a_i et a_j respectivement à (τ_k, τ_i) et (τ'_k, τ_j) tels que $\tau_i < \tau$ et $\tau_j < \tau$.

L'exemple suivant illustre la construction d'une connaissance commune entre deux agents.

Exemple 3.16

Soit deux agents a_i , a_j et trois dates $\tau_1 < \tau_2 < \tau_3$. Supposons que a_i a généré une proposition $I_t^{a_i}$ à la date τ_1 et que a_i a communiqué indirectement avec a_j à (τ_2, τ_3) . À la date τ_3 , la connaissance commune est construite ainsi :

1. Situons nous du point de vue de a_i . Par intentionalité, a_i connaît $I_t^{a_i}$, noté $K_{a_i}(I_t^{a_i})$. Par propagation, a_i sait que a_j connaît $I_t^{a_i}$, noté $K_{a_i}(K_{a_j}(I_t^{a_i}))$. Comme le modèle d'accointance est public et que a_j peut vérifier la propriété de propagation, a_i sait que a_j sait $K_{a_i}(K_{a_j}(I_t^{a_i}))$;
2. Situons nous du point de vue de a_j . Par intentionalité, a_j sait que a_i connaît $I_t^{a_i}$, noté $K_{a_j}(K_{a_i}(I_t^{a_i}))$. Par propagation, a_j sait que a_i sait que a_j connaît $I_t^{a_i}$, noté $K_{a_j}(K_{a_i}(K_{a_j}(I_t^{a_i})))$;

Ainsi a_i et a_j connaissent tous deux $I_t^{a_i}$, ils savent qu'autrui le sait et qu'il sait qu'ils le savent. Il s'agit d'une connaissance commune.

4.2 Avoir confiance dans la dernière communication

Lorsqu'un agent a_i communique à (τ_i, τ_j) une proposition (une intention de modalité \diamond), l'agent a_j qui la reçoit ne peut pas être certain qu'elle sera toujours vraie à la date τ' ($\tau' > \tau_i$). En effet, comme l'environnement est dynamique, un agent peut recevoir de nouvelles tâches ou de nouvelles intentions qui vont modifier son plan, c'est-à-dire ses propres propositions. Plus il y a de temps qui s'écoule entre la génération d'une proposition et la réalisation de la tâche concernée, plus la confiance en cette proposition s'érode. Cependant, une confirmation transmise par l'agent qui a généré la proposition permet de raffermir la confiance en celle-ci.

Comme nous considérons des agents honnêtes et coopératifs, une communication indirecte (qui est équivalente à un témoignage) est fiable en elle-même. C'est pourquoi un agent a_i considère qu'une proposition générée par l'agent a_j a été confirmée si a_j communique (directement ou non) avec a_i sans modifier cette proposition. Formellement, la dernière confirmation d'un agent est définie comme suit :

Définition 3.17 (Date de dernière confirmation)

Soit a_i un agent qui connaît une proposition $I_t^{a_j}$ de a_j à propos d'une tâche t . La date de dernière confirmation de la proposition $I_t^{a_j}$ pour a_i à la date τ est :

$$\tau^* = \max_{\substack{\tau_j : a_j \text{ communique } I_t^{a_j} \text{ à } a_i \text{ à } (\tau_j, \tau_i) \\ \tau_{K_{a_i}^\tau} < \tau_j \\ \tau_i < \tau}} \{ \tau_j : a_j \text{ communique } I_t^{a_j} \text{ à } a_i \text{ à } (\tau_j, \tau_i) \} \text{ et } I_t^{a_j} \text{ inchangée}$$

Exemple 3.18

Reprenons l'Exemple 3.3. Supposons que, à la date 15, a_3 calcule la confiance associée à une proposition que l'agent a_1 a générée à la date 7. a_1 a communiqué directement avec a_3 à la date 8 puis il a communiqué indirectement avec a_3 à la date (12, 15) sans modifier sa proposition. Ainsi, la dernière confirmation a eu lieu à la date 12 et a_3 sait que, entre les dates 7 et 12, a_1 a maintenu son intention. La confiance doit alors être calculée entre la date 13 et la date de réalisation de la tâche concernée par la proposition.

4.3 Calcul de confiance

Intuitivement, la confiance associée à une proposition dépend de la durée écoulée entre sa dernière confirmation et la date de réalisation de la tâche associée. Comme les agents n'ont pas de modèle de l'environnement, ils ne peuvent pas prédire l'arrivée de nouvelles tâches. Cependant, plus le temps passe, plus un agent peut en rencontrer un autre. Comme le disent [Eiben *et al.*, 2005] lorsqu'ils parlent de *vitesse de l'information*, il y a une très forte dépendance entre le temps où une information peut rester pertinente et la vitesse à laquelle les agents communiquent. Chaque rencontre est alors pour chaque agent une occasion de recevoir de nouvelles tâches et de réviser ses intentions.

Par conséquent, la confiance en une proposition d'un agent est définie par le nombre de rencontres que va faire ce dernier entre la date de sa dernière confirmation et la date de réalisation de la tâche associée. Comme ce nombre est donné par la relation Voisins, chaque agent peut calculer la confiance qu'il accorde aux intentions d'autrui.

Définition 3.19 (Rencontres)

Soit a_i un agent qui connaît la proposition $I_t^{a_j}$ à la date τ . Soit τ^* la date de dernière confirmation de $I_t^{a_j}$ pour a_i à la date τ . Le nombre d'agents $M_{\tau^*}^{a_i}(I_t^{a_j})$ que l'agent a_j rencontrera entre la date τ^* et la date $rea(I_t^{a_j})$ est donné par :

$$M_{\tau^*}^{a_i}(I_t^{a_j}) = \left| \bigcup_{\tau^* < \tau' < rea(I_t^{a_j})} Voisins(a_j, \tau') \right|$$

Finalement, un agent a confiance ou non dans une proposition. Formellement,

Définition 3.20 (Confiance)

Soit a_i un agent qui connaît la proposition $I_t^{a_j}$ à la date τ . L'agent a_i a confiance dans l'agent a_j à propos de $I_t^{a_j}$ si et seulement si $M_{\tau^*}^{a_i}(I_t^{a_j}) = 0$.

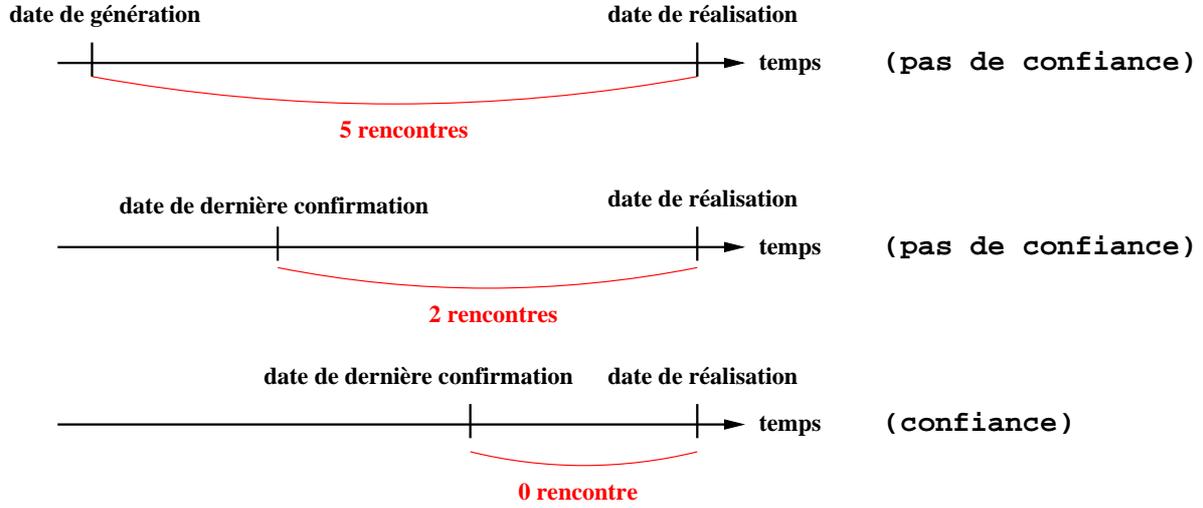


FIGURE 3.5 – Evolution de la confiance pour l'Exemple 3.21

L'exemple suivant illustre cette définition.

Exemple 3.21

Soit a_i un agent qui connaît la proposition $I_t^{a_j}$ à la date τ . Supposons que $M_{\tau^*}^{a_i}(I_t^{a_j}) = 5$. Dans ce cas, l'agent a_i n'a pas confiance en a_j au sujet de cette proposition. Supposons que a_j maintienne sa proposition suffisamment longtemps pour la confirmer deux fois. À chaque confirmation, a_i peut calculer $M_{\tau^*}^{a_i}(I_t^{a_j})$ à nouveau, respectivement 2 et 0. Ainsi à chaque confirmation, a_i a de plus en plus confiance en a_j . La Figure 3.5 illustre cette évolution.

Nous pouvons remarquer que le critère de confiance est dur : un agent n'inspire pas confiance dès qu'il rencontre un autre agent avant de réaliser la tâche associée à sa proposition ($M_{\tau^*}^{a_i}(I_t^{a_k}) = 0$). Cette hypothèse pessimiste peut être relâchée (par exemple $M_{\tau^*}^{a_i}(I_t^{a_k}) \leq 1$).

Cependant, en fonction de la structure de la relation Voisins, il existe un nombre de rencontres maximal au-delà duquel le critère de confiance est toujours optimiste (les agents ont toujours confiance).

Définition 3.22 (Seuil de confiance maximal)

Soit \mathcal{S} une constellation et $H \in \mathbb{T}$ l'horizon de planification⁴ des agents. $\forall a_i, a_j \in \mathcal{A}$, notons $f_j^i(\mathbb{T}_H)$ la fréquence de communication entre a_i et a_j sur chaque intervalle \mathbb{T}_H de longueur H tel que $\mathbb{T}_H \subseteq \mathbb{T}_P$. Le seuil de confiance maximal est :

$$F = \max_{a_i, a_j \in \mathcal{A}} f_j^i(\mathbb{T}_H)$$

4. On appelle horizon de planification l'horizon temporel sur lequel des agents peuvent planifier.

Exemple 3.23

Reprenons l'Exemple 3.3 et supposons que $H = P$. Chaque agent a_i peut rencontrer un autre agent avant de rencontrer a_j . Ainsi, le seuil de confiance maximal est de 1. Deux comportements peuvent être considérés :

1. une attitude pessimiste où un agent n'a pas confiance en un autre dès lorsqu'il rencontre un autre agent avant de réaliser les tâches associées à ses propositions ;
2. une attitude optimiste où un agent a toujours confiance en autrui.

Nous nous sommes intéressés dans ce chapitre aux notions de communication et de confiance. Nous avons proposé un protocole épidémique fondé sur l'écoute flottante afin de diffuser les connaissances des agents au sein de la constellation. À partir de la structure du modèle d'accointances et de ses propriétés (notamment le fait qu'il soit public), nous avons défini une notion de connaissance commune et une notion de confiance en cette connaissance. À partir de cette connaissance, les agents vont pouvoir coopérer.

Chapitre 4

Formation décentralisée de coalitions

Nous avons vu au chapitre précédent de quelle manière les agents construisaient une connaissance commune et comment ils pouvaient avoir confiance en cette connaissance. Dans ce chapitre, nous nous attachons à la coopération proprement dite. Nous présentons les notions de formation de coalitions ainsi que le modèle que nous avons choisi et proposons une méthode décentralisée de formation de coalitions pour permettre aux agents de coopérer.

1 De la coopération

Nous avons vu au Chapitre 2 Section 1.3 que les modèles de coopération peuvent être divisés en deux catégories : l'approche top-down et l'approche bottom-up, dont le travail d'équipe et la formation de coalitions sont respectivement représentatifs. Quelle que soit l'approche, comme le font remarquer [Dutta et Sen, 2002], il n'y a de possibilités de coopération que dans les situations où un agent peut aider un autre agent en permettant une économie globale de coût. Il est donc nécessaire de définir des mécanismes de coopération pour prendre en compte cette notion de coût.

1. Une méthode de coopération consiste à utiliser la **structure organisationnelle** du système. Les paradigmes maître-esclave où un agent central distribue et coordonne les tâches des agents qui ne font que du traitement en font partie. Il en est de même pour la négociation à l'aide d'un tableau noir où les agents lisent et écrivent sur le tableau tandis qu'un agent centralisateur gère les files d'accès. Cependant, le problème du paradigme maître-esclave est lié à l'hypothèse qu'un agent a une vue globale et à la perte du bénéfice du parallélisme. Dans cas d'un tableau noir, on suppose une sémantique identique pour tous les agents et il y a risque de goulet d'étranglement si les agents sont très nombreux ;
2. Une autre méthode est la **prise de contrat** avec, comme exemple, le célèbre protocole Contract Net [Smith, 1981]. Elle convient très bien à des tâches décomposables

hiérarchiquement mais ne prend pas en compte les demandes contradictoires. Elle ne détecte ni ne résout les conflits tout en requérant de nombreuses communications ;

3. Il est aussi possible de faire de la **planification multiagent centralisée** ou **distribuée**. Dans le premier cas, le plan local de chaque agent est envoyé à un agent central qui réalise la fusion, résout les conflits en y insérant des actions de coordination et retourne le plan ainsi formé aux agents. Dans l'approche distribuée, chaque agent possède un modèle du plan des autres. Ils communiquent entre eux pour mettre à jour leur plan et le modèle jusqu'à ce que les conflits soient résolus. La méthode centralisée partage les mêmes faiblesses que les architectures centralisées et la méthode décentralisée est difficile à mettre en œuvre, nécessitant beaucoup de calculs et de communications.

Nous pouvons remarquer qu'il n'y a pas de mécanisme de coopération générique mais des formes spécifiques parfois difficiles à exploiter dans un autre contexte. Les formes les plus génériques sont celles qui offrent le moins de flexibilité (centralisation et prise de contrat). Cependant, trois notions fondamentales se trouvent au cœur de ces mécanismes : les *protocoles de négociation*, les *prises de contrat* et les *stratégies*.

1.1 Protocoles de négociation

Les protocoles décrivent les types de relations qui existent entre les agents et les comportements qu'ils attendent des autres. Les protocoles de négociation, quant à eux, sont selon [Nwana *et al.*, 1996] les processus de communication qu'un groupe d'agents établit afin d'atteindre un accord commun sur un sujet donné. Cette définition est complétée par [Jennings *et al.*, 1998] : un protocole de négociation est un ensemble de règles qui gouvernent la coopération entre agents. Elles couvrent les participants admissibles, les états de la négociation, les événements qui font changer d'état la négociation et les actions que peuvent entreprendre les participants. Un protocole porte donc sur un *objet de négociation* qui est la structure sur laquelle porte la négociation (les contrats) et sur des *modèles de prise de décision* qui sont les mécanismes qu'emploient les agents afin de négocier (les stratégies).

Le protocole de négociation est donc un processus de recherche distribuée au sein d'un espace de compromis possibles dont la dimension et la topologie sont fonction de l'objet de négociation. Au début du processus, les participants ont un certain espace de compromis qu'ils considèrent comme acceptable. Durant le processus, cet espace peut changer. Le protocole prend fin lorsqu'un espace commun de compromis a été dégagé ou qu'il n'y a plus assez de participants pour négocier leur coopération. Ainsi, pour négocier, les participants doivent au moins être capables de faire ou répondre à des propositions et de détecter des conflits potentiels. Si les participants ne peuvent qu'accepter ou refuser les propositions, le temps pour atteindre un compromis risque d'être long car les participants n'ont aucun moyen de connaître les raisons d'un refus. C'est pourquoi enrichir les réponses

d'une *critique* ou d'une *contre-proposition* (suivant un modèle argumentatif) peut servir la négociation.

1.2 Prise de contrat

Cette méthode de coopération inspirée d'une métaphore économique se positionne dans ce que [Rosenschein et Zlotkin, 1994] ont appelé le domaine orienté-tâches. Dans un processus de prise de contrat, les agents échangent des tâches à réaliser dans un cadre d'échange économique. Une hypothèse de **rationalité individuelle limitée** des agents est faite: un agent n'accepte une tâche que s'il est payé plus que son coût marginal et un agent ne délègue une tâche que s'il paye moins qu'il n'épargne en la conservant pour lui; et les agents ne connaissent pas les tâches ou les fonctions de coût des autres. Ce mécanisme *anytime* correspond à une exploration **Hill Climbing** sur l'espace des tâches à affecter pour lequel la métrique serait de minimiser la somme des coûts des agents (aussi appelé bien-être social [Kreps, 1990, Moulin, 2003]).

Quatre types de contrats ont été définis par [Sandholm, 1998]. Un contrat de tâche ou *T-contrat* correspond à l'échange d'une unique tâche contre une valeur monétaire. Un contrat de groupe ou *G-contrat* est l'échange d'un ensemble de tâches contre une valeur monétaire. Un contrat d'échange ou *E-contrat* correspond à l'échange d'une tâche et d'une quantité de monnaie contre une autre tâche et une autre quantité de monnaie. Enfin, le contrat multiagent ou *M-contrat* est un contrat d'échange entre plus de deux agents.

[Sandholm, 1998] a prouvé que l'hypothèse de rationalité individuelle limitée des agents ne permet pas de garantir une solution optimale avec ces quatre types de contrat. Pour obtenir une solution optimale, les agents doivent pouvoir formuler des *TGEM-contrats* qui sont des contrats multiagents sur des contrats d'échange de groupe. Cependant, définir un tel contrat nécessite des agents très sophistiqués en termes de raisonnement.

1.3 Stratégies

Une stratégie est une règle employée par un agent afin de prendre une décision. Les stratégies possibles pour un choix donné dépendent de la connaissance de l'agent. Par exemple, dans les travaux de [Modi et Veloso, 2005], les agents doivent choisir entre accepter ou non une proposition qui entre en conflit avec une autre précédemment proposée. Deux stratégies sans connaissances peuvent être de toujours accepter la proposition et remettre en cause la précédente ou de ne jamais l'accepter si elle provoque un conflit. Une stratégie plus informée consisterait par exemple à n'accepter que la proposition qui concerne le plus grand nombre d'agents et remettre en cause l'autre.

[Durfee, 2001] définit l'espace des stratégies en fonction des propriétés de la population du système, des propriétés de l'environnement et des solutions que l'on désire.

- La taille de la population est fondamentale : si chacun des n agents doit choisir entre b actions alors l'espace des combinaisons possibles est b^n . De plus, des agents hétérogènes doivent avoir une ontologie commune pour communiquer ;
- L'impact de l'environnement doit être pris en compte en termes de dynamique, c'est-à-dire de ses changements au fil du temps et de la quantité d'interactions qu'il permet ;
- Une solution doit être mesurée en termes de qualité (optimalité sur une mesure de performance) et de robustesse (utilité en cas de changement dans l'environnement). Il faut aussi prendre en compte les limites intrinsèques du système comme les coûts, les délais de communication et les capacités de calcul.

Dans le Chapitre 2, les intentions des agents données en Définition 2.12 correspondent à un contrat. En effet, à une tâche et un agent donnés est associé un coût sous la forme du couple (date de réalisation, date de téléchargement). Dans le Chapitre 3, nous avons présenté le protocole de communication qui permet la diffusion de ces contrats. Dans les paragraphes suivants, nous proposons de nous servir de la notion de coalition pour définir les stratégies de coopération des agents.

2 Qu'est-ce que la formation de coalitions ?

Les coalitions sont des organisations d'agents au cycle de vie court. Elles sont formées avec un objectif précis et sont dissoutes lorsqu'elles n'ont plus de raison d'être. Au travers d'une coalition, chaque agent tente de maximiser un gain, collectif ou personnel selon que l'agent est coopératif ou compétitif. Dans la littérature, le problème de formation de coalitions, qui est un problème NP-dur, est décomposé en trois étapes [Sandholm *et al.*, 1999].

1. Générer des structures de coalitions, c'est-à-dire l'ensemble de toutes les coalitions pouvant réaliser les tâches considérées. Par exemple, supposons que le système contient trois agents a_1 , a_2 , a_3 et que les tâches à réaliser sont t_1 et t_2 . Supposons aussi que les deux agents a_1 et a_2 peuvent réaliser t_1 tandis que a_1 et a_3 peuvent réaliser t_2 . Ainsi, l'ensemble $\{\{a_1, a_2\}, \{a_1, a_3\}\}$ est une structure de coalition.
2. Choisir la structure qui sera retenue par tous les agents. Les structures générées à la 1^{re} étape sont différentes pour chaque agent en fonction de ses préférences. Il faut donc que ces agents négocient une unique solution.
3. Partager les gains obtenus par chaque coalition de la structure choisie entre les agents qui la composent. Notons que lorsque les agents sont coopératifs, le partage des gains est inutile car les agents ne s'intéressent pas à leur gain individuel. Ainsi, le processus se réduit aux deux premières étapes uniquement.

2.1 Problématiques induites par la formation de coalitions

Ces trois étapes sont profondément liées entre elles et il n'est pas aisé de les isoler. Cependant, chacune apporte ses propres problématiques.

Explorer le treillis des structures de coalitions

Différentes méthodes de génération des structures de coalitions ont été proposées dans la littérature. Ces méthodes se fondent sur quatre approches (qui peuvent être hybridées) synthétisées par [Thanh-Tung *et al.*, 2005, Thang-Tung, 2004] et dont les avantages et les inconvénients sont résumés sur le tableau de la Figure 4.1 :

1. les approches *incrémentales* [Abdallah et Lesser, 2004, Dang *et al.*, 2006, Goradia et Vidal, 2007, Ortiz et Rauenbusch, 2002, Shehory et Kraus, 1998, Shehory et Kraus, 1999, Sims *et al.*, 2003] où les agents explorent le treillis des structures de coalitions en partant des coalitions singletons (contenant un seul agent) puis incrémentent petit à petit la cardinalité des coalitions jusqu'à trouver une solution ;
2. les approches *par référence* [Kraus *et al.*, 2003] mettent en œuvre des agents qui considèrent des coalitions de taille maximale en fonction de leurs connaissances puis les raffinent en retirant les agents inutiles ;
3. les approches *aléatoires* [Scully *et al.*, 2004, Sen et Dutta, 2000] affectent aléatoirement les agents à des coalitions puis toutes les coalitions inacceptables sont fusionnées et le processus est réitéré sur ce sous-ensemble ;
4. les approches *adaptatives* [Sen et Airiau, 2006, Soh et Tsatsoulis, 2001, Soh et Tsatsoulis, 2002, Soh et Tsatsoulis, 2003, Vauvert et El Fallah-Seghrouchni, 2000] considèrent des agents hystérétiques qui conservent un historique des coalitions acceptables passées pour débiter le processus de génération dans le voisinage.

Approche	Nb. faible de tâches	Nb. élevé de tâches
Incrémentale	Très rapide Sous-optimal	Peu rapide Sous-optimal
Par référence	Peu rapide	Très rapide Réduit l'espace de recherche
Aléatoire	Anytime Exploration redondante	Rapide Configurations inhabituelles
Adaptative	Peu rapide Réduit l'espace de recherche	Peu rapide Très sensible à la dynamique

FIGURE 4.1 – Comparaison des méthodes de génération de structures de coalitions

Plus généralement, la problématique principale de la génération de structure de coalition est celle de l'exploration du treillis de ces mêmes structures (et par là-même la

réduction de l'espace de recherche). Des travaux comme ceux de [Rahwan *et al.*, 2007] ou de [Dang et Jennings, 2004] par exemple consistent à introduire de nouvelles métriques sur le treillis pour les approches incrémentales.

Choisir la structure optimale

Lorsqu'il s'agit de trouver la structure de coalitions optimale, les agents n'ont bien souvent qu'une information incertaine et (ou) incomplète sur les coûts et les préférences des autres agents. Ils doivent donc utiliser des heuristiques pour estimer la valeur d'une coalition. [Kraus *et al.*, 2003] en proposent deux : une *heuristique marginale* qui estime le coût des autres agents et cherche à maximiser le profit marginal estimé ainsi qu'une *heuristique experte* qui tente de minimiser le nombre d'agents concurrents au sein d'une même coalition (des agents ayant des compétences identiques). L'heuristique marginale tend à favoriser la maximisation de l'utilité (sans toutefois la garantir puisque les valeurs marginales ne sont que des estimations) tandis que l'heuristique experte réduit les collusions entre coalitions et permet de réaliser plus de tâches. Globalement, l'heuristique marginale est la plus adaptée en cas d'information incertaine tandis que l'heuristique experte est plus adaptée pour une information incomplète. Une notion de confiance peut être utilisée pour moduler ces heuristiques [Sen et Airiau, 2006].

Cependant la quasi-totalité des protocoles de formation de coalitions supposent la commensurabilité des fonctions d'utilité des agents. [Caillou *et al.*, 2002] puis [Aknine et Shehory, 2005, Aknine *et al.*, 2004] proposent un algorithme qui permet d'atteindre un optimum de Pareto sans agréger, ni comparer les utilités. Chaque agent partitionne les structures de coalitions en classes d'équivalence selon ses préférences. Un agent initiateur communique ensuite sa classe d'équivalence préférée. Chaque autre agent va alors réordonner les structures de coalitions en élaguant les classes d'équivalence les moins préférées. Les solutions restantes à la fin du processus correspondent à un optimum de Pareto. On remarquera cependant que la structure finale générée par ce protocole de négociation diffère suivant l'agent qui l'initie. C'est pourquoi [Aknine et Shehory, 2005, Aknine *et al.*, 2004] proposent de relancer le processus à partir d'un nouvel agent initiateur si l'ensemble des structures qui lui sont communiquées sont sous-optimales par rapport à sa situation actuelle.

Partager les gains

Les méthodes de formation des coalitions s'intéressent principalement aux deux premières phases du processus, c'est-à-dire la génération des structures de coalitions candidates et la sélection des coalitions acceptables. [Kraus *et al.*, 2004] s'intéressent à la troisième phase : le partage des gains de la coalition entre ses membres. En effet, cette dernière phase peut influencer sur la sélection des coalitions acceptables pour chaque agent car elle procure une **stabilité**, c'est-à-dire le fait qu'aucun agent n'est incité à quitter la coalition une fois celle-ci formée. Traditionnellement, la distribution du gain se fonde sur le concept de *noyau*

(ensemble de coalitions Pareto-optimales qui maximisent la valeur de Shapley¹ des agents) où le gain de chaque agent est donné par sa valeur de Shapley. Cependant cette méthode nécessite une information certaine et complète concernant les coûts et les capacités des agents. [Kraus *et al.*, 2004] proposent alors se servir des heuristiques suivantes :

- une distribution *égalitaire* où chaque agent reçoit un gain égal à son coût estimé plus une part identique du profit net de la coalition ;
- une distribution *proportionnelle* où chaque agent reçoit une part du profit proportionnelle à son coût estimé ;
- une distribution fondée sur des *compromis* où un agent peut se satisfaire d'une part α ($0 \leq \alpha \leq 1$) de son gain permettant ainsi de redistribuer une part $(1 - \alpha)$ de ce gain aux autres agents afin de rendre une coalition plus attractive.

Ces heuristiques peuvent être définies de manière hybride avec une distribution fondée sur des compromis. Cependant, si chercher des compromis est utile avec une distribution égalitaire, cela est moins utile avec une distribution proportionnelle et inutile avec la méthode du noyau qui stabilise déjà les coalitions.

2.2 Quelques approches de la formation de coalitions

Le problème canonique de la formation incrémentale de coalitions a été posé par [Shehory et Kraus, 1995a, Shehory et Kraus, 1995b]. Ce problème est celui de l'allocation de tâches parmi des groupes d'agents coopératifs. L'idée est qu'étant donné un ensemble de tâches, le système doit satisfaire toutes les tâches, ou au moins en satisfaire le plus possible, tout en maximisant son profit global et ce à l'aide de coalitions disjointes². Leur approche est une approche incrémentale où la taille des coalitions est bornée par un entier k (ce qui permet de borner la complexité de l'algorithme) et, dès qu'une coalition est formée, les agents qui la composent quittent le processus de génération. En vue de prendre en compte des coalitions recouvrantes, [Shehory et Kraus, 1998] modifient l'algorithme de manière à ne pas retirer les coalitions formées du processus mais à simplement adapter les ressources des agents.

Le problème des agents compétitifs, c'est-à-dire des agents rationnels qui tentent de maximiser une fonction d'utilité personnelle sans se soucier d'une utilité globale, a été posé par [Shehory et Kraus, 1999]. Le problème est désormais de définir une structure de coalitions *stable* où aucun agent ne désirera changer de coalition une fois qu'elles auront été calculées. Ils proposent alors un algorithme polynomial, le DNPK-CFM, qui calcule incrémentalement une approximation d'un équilibre de Nash. Tous les agents débutent comme coalition singleton et forment leurs coalitions par une succession de *propositions*

1. La valeur de Shapley, introduite par [Shapley, 1953], est pour un agent et une coalition donnés la somme pondérée de ses contributions marginales à la valeur de la coalition quelle que soit la structure de coalitions.

2. On dit qu'une structure de coalitions est *disjointe* si chaque agent n'appartient qu'à une seule et unique coalition. Dans le cas contraire, la structure de coalition est dite *recouvrante*.

acceptées ou rejetées par les autres. Ceci continue jusqu'à ce qu'il y ait un point fixe ou qu'un temps imparti soit écoulé. Un point fixe de l'environnement est un état dans lequel les agents sont organisés en une structure de coalitions optimale selon Pareto et stable, ou qu'il n'y a plus aucune proposition accordant un meilleur bénéfice pour eux. Les propositions sont ordonnées selon leur gain espéré et les agents acceptent celles qui maximisent ce gain. Cependant, la faible complexité de l'algorithme n'est due qu'à une borne sur la taille des coalitions. De plus, pour permettre une perception du point fixe, chaque coalition doit communiquer son statut (il n'y a plus de proposition à transmettre) aux autres.

Les approches de [Shehory et Kraus, 1995a, Shehory et Kraus, 1995b, Shehory et Kraus, 1998, Shehory et Kraus, 1999] recherchent une solution optimale en limitant la taille des coalitions pour assurer une faible complexité à leur algorithme. Cependant, cette heuristique ne garantit pas l'optimalité des solutions. En effet, ces dernières peuvent se trouver hors des bornes fixées. De plus, il est toujours nécessaire de calculer les structures des coalitions au sein de ces bornes, ce qui est d'une haute complexité lorsque les agents sont en grand nombre. C'est pourquoi [Sandholm *et al.*, 1999] proposent un algorithme *anytime* de génération des structures de coalitions permettant d'obtenir un résultat final au moins égal à une certaine proportion du résultat optimal, ce qui permet de diminuer la taille de l'espace de recherche. [Sandholm *et al.*, 1999] ont alors prouvé qu'aucun algorithme ne permet d'établir une borne au pire cas sur la valeur de la structure de coalitions³ en explorant moins de (2^{n-1}) nœuds où n est le nombre d'agents du système. De plus, n étant le numérateur de cette borne, elle devient de plus en plus imprécise au fur et à mesure que le nombre d'agents augmente.

S'appuyer sur la structure organisationnelle du système et sur la connaissance qu'en ont les agents peut aussi être une manière de réduire l'espace de recherche. [Abdallah et Lesser, 2004, Ortiz et Rauenbusch, 2002, Sims *et al.*, 2003] s'intéressent à cette approche. Ils utilisent une organisation *hiérarchique* des agents où les feuilles représentent les *agents* et les nœuds des *gestionnaires*. Ces derniers ne transmettent les tâches à réaliser qu'à leurs fils (leur congrégation). Ceci permet de réduire pour chaque processus de formation de coalitions le nombre d'agents impliqués. La différence entre [Abdallah et Lesser, 2004] et [Ortiz et Rauenbusch, 2002, Sims *et al.*, 2003] repose sur le protocole de négociation : enchères combinatoires et Contract Net respectivement.

[Soh et Tsatsoulis, 2003, Soh et Tsatsoulis, 2002, Soh et Tsatsoulis, 2001] proposent une approche adaptative pour former des coalitions au sein d'un réseau de capteurs où un agent ne peut suivre seul une cible. Pour former une coalition, l'agent fait appel à son voisinage et identifie les agents candidats en évaluant leur utilité par une somme pondérée d'une mesure de confiance passée, d'une mesure de confiance courante et d'une mesure de capacité. Plusieurs négociations entre des sous-ensembles de candidats ont lieu

3. C'est-à-dire la plus mauvaise valeur de la structure de coalitions.

en parallèle et les coalitions sont construites incrémentalement par un jeu de propositions et contre-propositions.

L'utilisation d'algorithmes génétiques, soit une approche aléatoire et adaptative, ont été envisagées par [Scully *et al.*, 2004, Sen et Dutta, 2000]. [Scully *et al.*, 2004] s'intéressent notamment à des agents ne raisonnant que selon leurs croyances quant aux capacités d'autrui. L'algorithme génétique construit une population dont la fonction d'adaptation est fondée sur une stratification selon une Pareto-dominance qui est modulée par la distance euclidienne au reste de la population et la distance aux coalitions précédemment formées. Cette méthode a pour but de travailler sur de multiples critères et les distances servent à couvrir le maximum de l'espace de recherche. L'algorithme retourne alors un ensemble de coalitions Pareto-dominantes. L'agent doit ensuite sélectionner dans cet ensemble sa coalition préférée et se fonde sur une minimisation de la distance aux coalitions précédentes qui lui ont été bénéfiques. En cas d'égalité, la préférence est tirée au hasard.

2.3 Limites des approches actuelles

De façon générale, les méthodes de formation de coalitions ont deux limites.

D'un côté, bien qu'elles soient souvent distribuées⁴ dans le sens où les agents ne calculent qu'une partie de la structure de coalition finale, elles sont centralisées car ou bien elles supposent une connaissance commune des tâches et un mécanisme de formation de coalitions hors-ligne [Rahwan *et al.*, 2007, Dang *et al.*, 2006, Goradia et Vidal, 2007, Shehory et Kraus, 1998, Shehory et Kraus, 1999]; ou bien elles utilisent un commissaire-priseur (ou un autre type de hiérarchie) [Abdallah et Lesser, 2004, Lau et Zhang, 2003, Ortiz et Rauenbusch, 2002, Sims *et al.*, 2003, Vauvert et El Fallah-Seghrouchni, 2000] qui centralise l'information et organise les négociations.

Afin de pallier cette limite, la formation décentralisée de coalitions a été envisagée par [Vauvert et El Fallah-Seghrouchni, 2001] et [Krainin *et al.*, 2007]. L'approche de [Vauvert et El Fallah-Seghrouchni, 2001] consiste à constituer dans un premier temps des congrégations (appelées alliances) qui regroupent les agents ayant des préférences proches. Ce sont ensuite les alliances qui vont négocier les coalitions à l'aide d'une approche adaptative. Bien qu'il s'agisse d'un algorithme distribué et décentralisé, il n'y a pas de contraintes de communication : chaque agent est en mesure de communiquer avec tous les autres. [Krainin *et al.*, 2007] abandonnent cette hypothèse et proposent une approche incrémentale et décentralisée en trois étapes :

1. Chaque agent calcule son meilleur plan en fonction de sa connaissance des plans de ses voisins. Il calcule alors l'utilité marginale de chaque révision potentielle et

4. Nous faisons une distinction entre distribution et décentralisation. Nous parlons de distribution pour un processus dont les entrées et les sorties sont centralisées mais dont le calcul ou l'exécution sont répartis. Nous parlons de décentralisation lorsque le calcul, l'exécution mais aussi les entrées et les sorties sont répartis entre différents agents.

sélectionne la meilleure avant d'envoyer celle-ci sous forme de proposition à ses voisins ;

2. Ayant reçu les propositions des voisins, un agent ne modifie son plan que s'il dégage la plus grande utilité marginale positive (en cas d'égalité c'est l'agent ayant le numéro d'identification le plus élevé qui l'emporte) ;
3. Chaque agent envoie son plan courant à ses voisins et le processus est réitéré.

En ce qui concerne les communications, les méthodes fondées sur la structure organisationnelle du système s'intéressent à quelques contraintes : les agents ne peuvent communiquer qu'au travers d'une hiérarchie [Abdallah et Lesser, 2004, Ortiz et Rauenbusch, 2002, Sims *et al.*, 2003] ou d'un voisinage [Krainin *et al.*, 2007, Soh et Tsatsoulis, 2001, Soh et Tsatsoulis, 2002, Soh et Tsatsoulis, 2003]. Ces contraintes sont associées à des coûts de communication [Tohmé et Sandholm, 1999] dont la complexité a été calculée [Procaccia et Rosenschein, 2006]. Cependant ces approches ne considèrent que des organisations statiques et les modèles d'acointances n'évoluent pas avec le temps. De plus, dans un environnement réel dynamique, les agents ne peuvent pas toujours échanger de l'information et doivent décider seuls. De plus, certaines tâches ne peuvent pas attendre le calcul complet de la structure de coalitions et doivent être réalisées très vite.

3 Des satellites et des coalitions

Les mécanismes de formation de coalitions nous intéressent particulièrement pour trois raisons :

1. les agents s'associent pour réaliser une tâche collective ou se partager des tâches ;
2. ils s'organisent en structures au cycle de vie court, ce qui permet une adaptation à un environnement dynamique ;
3. les agents tentent de trouver les solutions les plus efficaces en présence d'informations incertaines et (ou) incomplètes.

Dans notre application, les tâches complexes nécessitent que des agents réalisent conjointement un ensemble de tâches atomiques. Cependant, ces réalisations conjointes ne peuvent pas être planifiées par les agents car un agent ne planifie pas pour les autres.

Afin d'organiser dynamiquement les agents, nous proposons un mécanisme de formation de coalitions décentralisé qui prend en compte les caractéristiques de notre problème, c'est-à-dire : agents coopératifs et communications restreintes. Ainsi, nous nous intéressons aux deux premières étapes du processus de formation de coalitions, en écartant celle de la distribution de gain car nos agents ne tentent pas de maximiser un gain individuel. Nous proposons dans ce paragraphe un mécanisme décentralisé fondé sur deux points :

1. les agents construisent des coalitions de taille maximale en fonction de leurs connaissances ;

2. ces coalitions sont ensuite raffinées au fil des rencontres des agents en retirant les agents inutiles.

3.1 Définitions

Les agents doivent générer dynamiquement des coalitions, sélectionner les plus intéressantes et maintenir la stabilité de ces coalitions. En effet les agents pouvant se désister lors d'une phase de planification, il est nécessaire de définir un mécanisme pour éviter le retrait d'une coalition alors qu'elle a été formée.

Nous définissons les coalitions comme suit :

Définition 4.1 (Coalition)

Une coalition C est un triplet $\langle A, O, \mathfrak{P} \rangle$ tel que $A \subseteq \mathcal{A}$ est un sous-ensemble d'agents membres de la coalition, O est un ensemble de tâches atomiques objectifs de la coalition et \mathfrak{P} un ensemble de tâches atomiques représentant le pouvoir de la coalition.

La *taille* de la coalition est $|A|$. Le pouvoir d'une coalition est fonction des intentions des agents qui en sont membres : une coalition a le pouvoir de réaliser une tâche t si au moins un agent de la coalition se propose (\diamond) ou s'engage (\square) à réaliser t .

Définition 4.2 (Pouvoir)

Une coalition C a le pouvoir de réaliser son objectif O si et seulement si $O \subseteq \mathfrak{P}$. L'ensemble \mathfrak{P} est défini comme suit : $\mathfrak{P} = \{t_j \in O \mid \exists a_i \in A : \text{mod}(I_{t_j}^{a_i}) \in \{\square, \diamond\}\}$

Dans notre approche, qui est une approche par référence, la notion de pouvoir correspond à la capacité de la coalition à réaliser ses objectifs.

Définition 4.3 (Coalition complète)

Une coalition C est dite *complète* si et seulement si C a le pouvoir de réaliser son objectif O . Dans le cas contraire, on dira que la coalition est *incomplète*.

Afin d'éviter les redondances, nous considérons les coalitions minimales comme les plus petites coalitions pouvant réaliser leur objectif :

Définition 4.4 (Coalition minimale)

Une coalition C est dite *minimale* si et seulement si l'ensemble A est minimal pour l'inclusion parmi les coalitions qui ont le pouvoir de réaliser O .

Les coalitions sont construites et manipulées *localement* par chaque agent, en fonction des connaissances qu'il a des autres agents *via* la communication. Ainsi, chaque agent utilise la notion de coalition pour raisonner et adapter ses propres intentions aux intentions des autres. Les coalitions sont donc formées implicitement au travers des intentions mais ne sont jamais construites explicitement par le système multiagent. Chaque agent :

1. calcule la structure de coalitions courante en fonction de ses connaissances ;
2. vérifie s'il peut rejoindre une coalition pour accroître son pouvoir ;
3. vérifie s'il peut quitter une coalition pour la minimiser ;
4. modifie ses intentions en fonction.

3.2 Calculer la structure de coalitions

L'étape de génération des coalitions est effectuée individuellement par chaque agent. Il doit :

1. considérer l'ensemble des coalitions potentielles du système.
2. calculer le pouvoir et l'état de chacune d'entre elles.

Chaque agent a_i génère la structure de coalitions courante de la manière suivante :

1. a_i partitionne l'ensemble des tâches atomiques $\mathcal{T}_{a_i}^\tau$ en un ensemble $\{\mathcal{T}_1 \dots \mathcal{T}_h\}$ tel que $\mathcal{T}_i \subseteq \mathcal{T}_{a_i}^\tau$ est une tâche complexe ;

Exemple 4.5

Supposons $\mathcal{T}_{a_i}^\tau = \{t_1, t_2, t_3, t_4, t_5\}$. Supposons que les tâches atomiques t_1 et t_2 soient les sous-tâches d'une tâche complexe T_1 et qu'il en est de même pour les tâches atomiques t_4 , t_5 et une tâche complexe T_3 . Alors $\mathcal{T}_{a_i}^\tau$ est partitionné en $\{\{t_1, t_2\}, \{t_3\}, \{t_4, t_5\}\}$.

2. chaque \mathcal{T}_i est l'objectif d'une coalition potentielle. Comme ces ensembles sont disjoints, le nombre de coalitions générées par l'agent a_i dépend du nombre de tâches complexes qu'il connaît ;
3. du point de vue de a_i , les membres d'une coalition pour un sous-ensemble \mathcal{T}_i sont définis par : $\{a_k \in \mathcal{A} : \exists t \in \mathcal{T}_i / \text{mod}(I_t^{a_k}) \in \{\square, \diamond\} \text{ et } I_t^{a_k} \in \mathcal{K}_{a_i}^\tau\}$

Exemple 4.6

Reprenons l'Exemple 4.5. Considérons la tâche t_3 et supposons que $\text{mod}(I_{t_3}^{a_i}) = \diamond$, $\text{mod}(I_{t_3}^{a_k}) = \square$ et $I_{t_3}^{a_k} \in \mathcal{K}_{a_i}^\tau$. a_i peut construire la coalition $C = \langle \{a_i, a_k\}, \{t_3\}, \{t_3\} \rangle$. Cette coalition est complète mais pas minimale car $\{a_i, a_k\}$ n'est pas minimal pour l'inclusion. Remarquons que a_i planifie t_3 même s'il sait que a_k a fait de même. En effet, les intentions d'autrui ne sont pas prises en compte dans la planification : elles seront considérées lors des phases suivantes de la collaboration.

On peut remarquer que ce cadre capture à la fois des coalitions *recouvrantes* et des coalitions *disjointes* en fonction des capacités des agents qui sont exprimées au travers des intentions. Si un agent formule plusieurs intentions, il peut rejoindre plusieurs coalitions ; s'il formule une seule intention (par manque de ressources), l'agent ne rejoindra qu'une seule coalition.

Chaque coalition ainsi formée peut être minimale (et donc complète), non minimale mais complète, ou incomplète (donc non minimale).

3.3 Rejoindre une coalition

Une coalition incomplète signifie qu'au moins une tâche objectif n'est pas dans le pouvoir de la coalition. Si une coalition reste incomplète, tous ses membres vont gaspiller leurs

ressources en réalisant des tâches atomiques mais en ne réalisant pas la tâche complexe associée. Ainsi, plus il y a une partie importante de l'objectif est dans le pouvoir d'une coalition, plus il devient impératif de compléter cette coalition afin d'éviter un important gaspillage de ressources. Nous proposons un mécanisme d'incitation pour accroître le pouvoir des coalitions incomplètes.

Lorsqu'un agent a_i calcule la structure de coalitions courante, il détecte les coalitions incomplètes. Comme a_i est coopératif, il devrait être incité à modifier ses intentions pour compléter ces coalitions lorsqu'il planifie. Pour cela, nous proposons d'accroître les priorités des tâches objectifs des coalitions incomplètes. Dans toute la suite, nous noterons $prio(t)'$ la priorité de la tâche atomique t que a_i utilise lors de sa prochaine phase de planification. Remarquons que $prio(t)'$ est une priorité locale, utilisée uniquement par a_i . La priorité commerciale ou physique $prio(t)$ de la tâche t (définie par l'utilisateur de la constellation) reste inchangée.

Protocole 4.7 (Rejoindre une coalition)

Pour chaque coalition incomplète C d'objectif O et de pouvoir \mathfrak{P} , l'agent a_i exécute : $\forall t \in O$, $prio(t)' \leftarrow \frac{prio(t)}{1+|\mathfrak{P}|}$.

Les agents sont encouragés à rejoindre une coalition si et seulement si l'objectif de la coalition est de réaliser une tâche composée qui a été partiellement planifiée.

En ce qui concerne une tâche complexe singleton $\{t\}$:

- si t n'est pas planifiée, c'est qu'elle ne satisfait aucun critère d'optimisation (Chapitre 2 Section 3.3) : il n'y a aucune coalition formée et la priorité de t reste la même ;
- si t est planifiée, la coalition ayant pour objectif t est complète et la priorité de t reste inchangée.

Ce mécanisme est *stable*. Deux phases d'incitation successives ne sont pas incohérentes. Par exemple, un agent ne doit pas être incité à abandonner une tâche atomique pour en réaliser une autre puis, *ceteris paribus*, être incité à abandonner cette dernière pour réaliser la première.

Proposition 4.8 (Stabilité de l'incitation)

Le Protocole 4.7 est stable.

Démonstration 4.8

Soit $C = \langle A, O, \mathfrak{P} \rangle$ une coalition incomplète où O n'est pas un singleton. Après application du Protocole 4.7, $\forall t \in O$, $prio(t)' = \frac{prio(t)}{1+|\mathfrak{P}|}$. $\forall t_1, t_2 \in O$ telles que $prio(t_1) < prio(t_2)$, $prio(t_1)' < prio(t_2)'$. Ainsi pour chaque agent, le ratio des priorités reste inchangé au sein de la coalition. Le protocole est stable.

3.4 Minimiser une coalition

S'il existe une coalition complète non minimale, il s'agit d'une coalition qui peut atteindre son objectif mais en y affectant des agents inutiles, c'est-à-dire des agents qui effectuent le même travail qu'un autre. Un agent membre d'une coalition doit donc considérer tous les agents qui ont les mêmes compétences que lui (réaliser une ou plusieurs tâches identiques au travers d'une ou plusieurs intentions) et prendre la décision de modifier ou non ses propres intentions.

Conflit

Afin de prendre une décision, c'est-à-dire s'engager ou se désister fortement, un agent doit considérer les autres membres de la coalition, leurs intentions et leurs capacités de communication. De plus, cette décision doit être prise avant la date de réalisation de la tâche. Nous définissons le conflit comme étant l'ensemble des agents d'une coalition devant prendre une décision concernant une tâche donnée t .

Définition 4.9 (Conflit)

Soit deux agents a_i et a_j et une coalition $C = \langle A, O, \mathfrak{P} \rangle$ telle que $\{a_i, a_j\} \subseteq A$. $\forall t \in \mathfrak{P}$, les agents a_i et a_j sont en conflit si et seulement si $\text{mod}(I_t^{a_i}) \in \{\square, \diamond\}$ et $\text{mod}(I_t^{a_j}) \in \{\square, \diamond\}$.

Exemple 4.10

Reprenons l'Exemple 4.6. Comme la coalition n'est pas minimale et qu'il y a un conflit sur la tâche t_3 , a_i doit prendre la décision de se désister fortement ($\square \neg$), s'engager (\square) ou de maintenir son intention (\diamond).

Chaque conflit pour un agent a_i peut être de deux types différents en fonction de ses capacités de communication.

Définition 4.11 (Type de conflit)

Soit un agent a_i et un ensemble A^* d'agents avec lesquels il est en conflit sur une tâche t . A^* peut être partitionné en deux sous-ensembles A^+ et A^- :

1. **conflit souple** : (Figure 4.2) A^+ est un sous-ensemble d'agents a_k tels que : soit a_i peut communiquer (directement ou non) avec a_k à la date $\tau \in \mathbb{T}$ où $\tau_{I_t^{a_i}} < \tau < \min(\text{rea}(I_t^{a_i}), \text{rea}(I_t^{a_k}))$; ou a_k connaît l'intention de a_i à propos de t .
2. **conflit dur** : (Figure 4.3) $A^- = A^* \setminus A^+$.

Proposition 4.12 (Symétrie)

Soit a_i un agent et A^* l'ensemble des agents avec lesquels il est en conflit sur une tâche t . $\forall a_k \in A^+$, le conflit est symétrique. $\forall a_k \in A^-$, le conflit est asymétrique.

Démonstration 4.12

Soit a_i un agent et A^* l'ensemble des agents avec lesquels il est en conflit sur une tâche t .

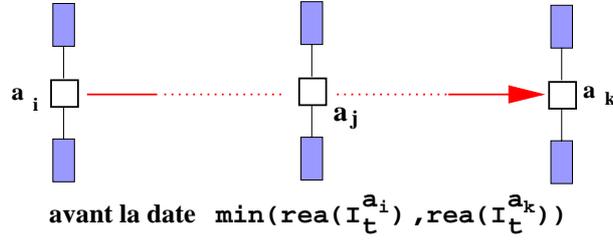


FIGURE 4.2 – Conflit souple

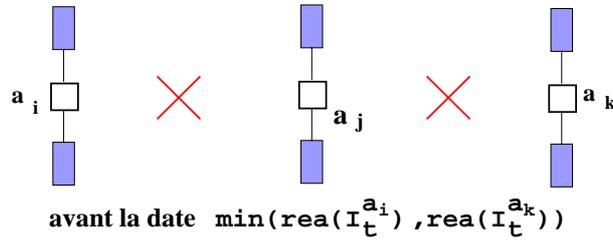


FIGURE 4.3 – Conflit dur

- $\forall a_k \in A^+$, a_i connaît $I_t^{a_k}$. À l'inverse, soit a_k connaît $I_t^{a_i}$, ou $\exists \tau_i, \tau_k \in \mathbb{T}$ telles que $\tau_{I_t^{a_i}} < \tau_i$ et $\tau_k < \min(\text{rea}(I_t^{a_i}), \text{rea}(I_t^{a_k}))$ où a_k connaîtra $I_t^{a_i}$ car il recevra l'information. Dans les deux cas le conflit souple est symétrique ;
- $\forall a_k \in A^-$, a_k ne connaît pas $I_t^{a_i}$ et ne le connaîtra pas avant la date $\min(\text{rea}(I_t^{a_i}), \text{rea}(I_t^{a_k}))$. Ainsi, a_k ne sera pas au courant du conflit ; le conflit dur est asymétrique.

Conflits souples et conflits durs sont résolus au travers de protocoles fondés sur l'expertise des agents pour réaliser la tâche.

Critère d'optimisation

Comme nous cherchons à optimiser la réactivité du système, les agents doivent s'engager ou se désister fortement en fonction de leur capacité à réaliser rapidement la tâche et à consommer le moins de ressources possibles⁵. Agrégeons ces deux critères au sein d'un unique critère d'expertise. Formellement,

Définition 4.13 (Expertise)

Soit un agent a_i et un ensemble d'agents $A^* \subseteq \mathcal{A}$ avec lequel a_i est en conflit sur une tâche t . Notons $\text{rea}^* = \min_{a_k \in A^*} \text{rea}(I_t^{a_k})$. L'agent a^* est l'expert pour t si et seulement si :

$$a^* = \arg \min_{a_k \in A^* \cup \{a_i\}} \|(\text{rea}(I_t^{a_k}) - \text{rea}^*, \text{tel}(I_t^{a_k}) - \text{rea}^*)\|$$

5. Dans notre application, il s'agit de conserver les images en mémoire le moins longtemps possible.

L'expertise de l'agent a_i est notée $R(a_i)$ et donnée par :

$$R(a_i) = \|(rea(I_t^{a_i}) - rea^*, tel(I_t^{a_i}) - rea^*)\|$$

La Figure 4.4 est une représentation du critère d'expertise à propos d'une tâche t dans le plan $(rea(I_t^{a_k}), tel(I_t^{a_k}))$, $a_k \in A^*$. L'origine rea^* est la date de réalisation la plus proche possible pour t ; l'intention (rea^*, rea^*) est l'intention idéale correspondant à l'agent pouvant réaliser la tâche le plus rapidement et pouvant télécharger l'image associée immédiatement après. tel^* est la date de téléchargement de l'agent pouvant réaliser t à la date rea^* . Comme il existe toujours une date future de téléchargement pour un agent, tel^* ne peut pas tendre vers l'infini. De plus, il est trivial que $tel(I_t^{a_k}) > rea(I_t^{a_k})$; par conséquent seule la partie hachurée est significative.

Tout point de la partie hachurée est une intention potentielle $I_t^{a_k}$ concernant t . La consommation de ressources, c'est-à-dire le temps pendant lequel l'image associée à t reste dans la mémoire du satellite, est défini comme une durée. La distance entre une intention potentielle et l'intention idéale représente la projection des critères temporels sur le plan $(rea(I_t^{a_i}), tel(I_t^{a_i}))$. Cette distance représente un compromis entre date de réalisation et consommation en ressources mémoire. L'agent expert est celui qui minimise cette distance.

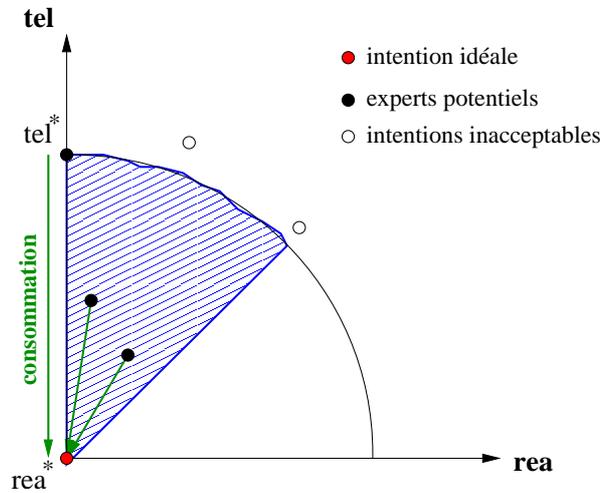


FIGURE 4.4 – Critère d'expertise pour la tâche t

Stratégies de coordination

Soit un agent a_i et l'ensemble A^* des agents avec lesquels il est en conflit sur une tâche t . L'agent a_i doit prendre une décision pour la question suivante : doit-il se désister ($\square\neg$), s'engager (\square) ou maintenir sa proposition (\diamond)? Nous définissons trois stratégies qui associent à une situation donnée une des trois décisions possibles :

1. la *stratégie sécuritaire* consiste pour a_i à maintenir sa proposition (\diamond) si a_i n'a confiance dans aucun autre agent de A^* quant à leurs intentions. Comme ces agents peuvent changer leurs intentions, cette stratégie permet une redondance afin d'assurer que la tâche sera réalisée ;
2. la *stratégie collaborative* consiste pour a_i à s'engager (\square) si a_i est expert. a_i contraint ainsi une partie de son plan et renforce le pouvoir de la coalition dont la tâche pour laquelle il s'engage est un objectif ;
3. la *stratégie opportuniste* consiste pour a_i à se désister ($\square \neg$) si a_i ne peut appliquer aucune des deux autres stratégies. En effet, a_i sait que l'expert s'est engagé ou qu'il va le faire en raison de la confiance qu'il lui porte. Il est donc plus intéressant pour a_i d'économiser des ressources en minimisant la taille de la coalition dont la tâche objet du conflit est un objectif.

À partir de ces trois stratégies, de l'expertise et de la mesure de confiance, construisons les deux protocoles de résolution de conflits. Nous rappelons ici que $M_{\tau^*}^{a_i}(I_t^{a_k})$ est le nombre de rencontres que fait l'agent a_k entre la date de dernière confirmation τ^* à l'agent a_i et la date de réalisation de la tâche associée à l'intention $I_t^{a_k}$.

Protocole 4.14 (Résolution d'un conflit dur)

Soit A^* l'ensemble des membres de la coalition avec lesquels l'agent a_i est en conflit sur une tâche t tel que $A^- \neq \emptyset$. a_i est au courant du conflit et applique :

1. si $\min_{a_k \in A^-} M_{\tau^*}^{a_i}(I_t^{a_k}) > 0$ alors $\text{mod}(I_t^{a_i}) \leftarrow \diamond$
2. sinon $\text{mod}(I_t^{a_i}) \leftarrow \square \neg$

En cas de conflit dur, l'agent au courant du conflit applique (1) la stratégie sécuritaire s'il n'a confiance dans aucun autre agent ; sinon (2) s'il a confiance dans au moins un agent, il applique la stratégie opportuniste.

Protocole 4.15 (Résolution d'un conflit souple)

Soit A^* l'ensemble des membres de la coalition avec lesquels l'agent a_i est en conflit sur une tâche t tel que $A^+ \neq \emptyset$. Soit $\text{rea}^* = \min_{a_k \in A^+} \text{rea}(I_t^{a_k})$. L'agent a_i applique :

1. si $a_i = \arg \min_{a_k \in A^+} \|\text{rea}(I_t^{a_k}) - \text{rea}^*, \text{tel}(I_t^{a_k}) - \text{rea}^*\|$ alors $\text{mod}(I_t^{a_i}) \leftarrow \square$
2. sinon soit a^* l'agent expert :
 - (a) si $M_{\tau^*}^{a_i}(I_t^{a^*}) > 0$ alors $\text{mod}(I_t^{a_i}) \leftarrow \diamond$
 - (b) sinon $\text{mod}(I_t^{a_i}) \leftarrow \square \neg$

En cas de conflit souple, l'agent détermine qui est l'agent expert. (1) Si c'est lui, il applique la stratégie collaborative et s'engage. (2.a) Sinon, s'il n'a pas confiance dans l'agent expert, il applique la stratégie sécuritaire (2.b) Sinon, s'il a confiance dans l'agent expert, il applique la stratégie opportuniste.

4 Propriétés de la formation de coalitions

4.1 Convergence et optimalité

Dans ce paragraphe, nous noterons \mathcal{C} la structure de coalitions courante et $v(\mathcal{C})$ sa valeur qui correspond à la valeur du plan global.

Définition 4.16 (Valeur de la structure de coalition)

Soit une constellation \mathcal{S} et $\mathcal{C} = \{C\}$ la structure de coalitions courante. La valeur de \mathcal{C} , notée $v(\mathcal{C})$ est donnée par :

$$v(\mathcal{C}) = \sum_{C \in \mathcal{C}} v(C) \text{ où } v(C) = \begin{cases} \sum_{t_k \in O} \frac{1}{prio(t_k)} & \text{si } C \text{ est complète} \\ 0 & \text{sinon} \end{cases}$$

La valeur d'une structure de coalitions est donc différente de la valeur des plans individuels des agents qui est, à partir de la Définition 2.14, calculée ainsi :

Définition 4.17 (Valeur du plan individuel)

Soit un agent a_i et $\mathcal{I}_{a_i}^T$ l'ensemble de ses intentions. La valeur du plan individuel correspondant, notée $v(a_i)$, est donnée par :

$$v(a_i) = \sum_{t_k: mod(I_{t_k}^{a_i}) \in \{\diamond, \square\}} \frac{1}{prio(t_k)}$$

[Wieder, 2008] a dénombré le nombre de coalitions et de structures de coalitions possibles sous diverses hypothèses. Cependant, notre domaine qui n'est ni super-additif et ni sous-additif possède trois particularités. Tout d'abord, le nombre de coalitions est fixe et égal au nombre de tâches complexes dans \mathcal{T} . De plus, un agent a_i peut n'appartenir à aucune coalition. Enfin, les coalitions sont recouvrantes.

Nous nous intéressons alors aux combinaisons de $|\mathcal{T}|$ objets (les tâches complexes du système) parmi $2^{|\mathcal{A}|}$ objets discernables (les sous-ensembles d'agents possibles) avec remise (due aux coalitions recouvrantes) et avec ordre (car une coalition correspond à un objectif). Ainsi,

Proposition 4.18 (Cardinalité du treillis des structures de coalitions)

Soit un treillis \mathbb{M} de structures de coalitions. La cardinalité du treillis est $|\mathbb{M}| = 2^{|\mathcal{A}| \times |\mathcal{T}|}$.

Le mécanisme proposé au paragraphe précédent explore le treillis \mathbb{M} ainsi :

1. soit $|\mathcal{A}|$ processus d'exploration notés a_i ;
2. chaque processus a_i est initialisé à $\mathcal{C}_{a_i} \in \mathbb{M}$;

3. a_i explore $2^{|\mathcal{T}|}$ structures⁶ jusqu'à trouver un optimum local noté $\mathcal{C}_{a_i}^*$;
4. lorsque a_i et a_j communiquent,
 - (a) ils fusionnent leurs structures en $\mathcal{C}_{a_i}^* \cup \mathcal{C}_{a_j}^*$;
 - (b) une sous-partie de la structure est fixée.
5. les processus a_i et a_j relancent l'exploration à partir de cette structure.

Nous devons donc nous assurer que le mécanisme proposé (étape 4) ne dégrade pas la valeur de la nouvelle structure de coalition.

Proposition 4.19 (Effet de la formation de coalitions)

Soit une constellation \mathcal{S} et un agent a_i . L'application du mécanisme de formation de coalitions proposé dans la Section 3 suite à une communication venant d'un agent a_j ne peut pas dégrader la valeur de la structure de coalitions après une phase de replanification.

Démonstration 4.19

Soit \mathcal{C} la structure de coalitions courante, $v(\mathcal{C})$ sa valeur et un agent a_i . Supposons que a_i modifie ses intentions suite à une phase de collaboration : la structure de coalitions courante devient \mathcal{C}' et le gain en valeur globale est $v(\mathcal{C}') - v(\mathcal{C})$. Ce gain dépend des modifications d'intentions de a_i qui sont de trois types :

1. *un engagement (\square) suite à un conflit. Dans ce cas, la valeur du plan individuel de a_i reste inchangée ainsi que la valeur de la structure de coalitions et cela peut conduire les autres agents a_k en conflit à se désister fortement ($\square \neg$) ;*
2. *un désistement fort ($\square \neg$) suite à un conflit. Dans ce cas, la valeur du plan individuel de a_i diminue mais, comme il y avait une redondance, la valeur de la structure de coalitions reste inchangée. Lors de la phase de replanification, l'agent a_i pourra profiter de la relaxation de certaines contraintes d'exclusion mutuelle et de ressource mémoire pour réaliser de nouvelles tâches ;*
3. *une modification de priorité suite à une incitation ne modifie pas directement la valeur du plan individuel de a_i ni celle de la structure de coalitions. Après une phase de replanification, a_i peut accroître la valeur de la structure de coalitions (en complétant une coalition).*

Parce que le mécanisme est décentralisé, il n'est pas possible d'assurer qu'il converge vers un optimum global mais, parce qu'il ne dégrade pas les plans localement optimaux des agents, il permet d'accroître la valeur de la structure de coalitions courante en fonction des connaissances partielles des agents.

6. Parce que l'agent a_i planifie individuellement, il explore l'ensemble des structures de coalitions atteignables en modifiant ses propres intentions et sélectionne la structure qui maximise la valeur.

Ainsi,

$$\begin{aligned}
R(a_i) &= \|(rea(I_t^{a_j}) - rea^*, rea(I_t^{a_j}) - rea^*)\| \\
R(a_i) &= \sqrt{(rea(I_t^{a_j}) - rea^*)^2 + (rea(I_t^{a_j}) - rea^*)^2} \\
R(a_i) &= \sqrt{2(rea(I_t^{a_j}) - rea^*)^2} \\
R(a_i) &= \sqrt{2}(rea(I_t^{a_j}) - rea^*) \\
rea(I_t^{a_j}) &= \frac{R(a_i)}{\sqrt{2}} + rea^*
\end{aligned}$$

Donc toute date de réalisation proposée par un futur agent expert sera inférieure ou égale à $\frac{R(a_i)}{\sqrt{2}} + rea^*$ où rea^* est la première date de réalisation idéale calculée par le premier agent expert. Comme rea^* est constante lorsque le temps s'écoule, la tâche t sera nécessairement réalisée au plus tard à $\frac{R(a_i)}{\sqrt{2}} + rea^*$.

Remarquons que si la délégation nous assure formellement qu'une tâche sera réalisée avant une date donnée, il nous faut tenir compte de la notion de confiance. En effet, il n'y a délégation que si l'agent qui s'est engagé a confiance dans le nouvel agent expert. Comme toute confiance, celle-ci peut être mise en porte-à-faux par la dynamique du système : le nouvel expert peut se désister avant d'être au courant du conflit⁷.

4.3 Utilité de la formation de coalitions

Si le mécanisme de formation de coalitions ne dégrade pas la valeur de la structure de coalitions courante et permet d'en accroître la valeur, il permet aussi d'accroître l'utilité des fenêtres de communication. En effet, une information sur une intention n'est utile que si elle permet à l'agent qui la reçoit de prendre une décision qu'il n'aurait pu prendre auparavant.

Proposition 4.21 (Effet des communications sur la formation de coalitions)

Soit une constellation \mathcal{S} . Plus les communications sont fréquentes, moins le mécanisme de formation de coalitions proposé est utile. A contrario, plus les communications sont rares, plus le mécanisme proposé est utile.

Démonstration 4.21

Soit deux agents a_i et $a_j \in \mathcal{A}$ ayant l'intention de réaliser une même tâche t . Ces intentions sont respectivement $I_t^{a_i}$ et $I_t^{a_j}$. Considérons alors une communication de a_i vers a_j à (τ_i, τ_j)

7. Supposons un agent a_i en conflit sur une tâche t avec un agent a_j . S'il s'agit d'un conflit souple, a_j peut ne pas être immédiatement au courant du conflit mais il le sera à partir d'une date τ_j donnée. Cependant, si a_j reçoit de nouvelles tâches entre la date courante τ et τ_j , il peut modifier ses intentions et sortir du conflit.

et supposons que a_i peut réaliser t avant a_j . Dans le contexte de a_j , l'utilité de $I_t^{a_i}$ est fonction des dates de communication. Nous pouvons distinguer trois cas comme l'illustre la Figure 4.6 :

1. $\tau_j \leq \text{rea}(I_t^{a_i})$ et $\tau_j \leq \text{rea}(I_t^{a_j})$: l'information est utile à a_j pour former une coalition ;
2. $\tau_j > \text{rea}(I_t^{a_i})$ et $\tau_j \leq \text{rea}(I_t^{a_j})$: l'information est utile à a_j pour se désister ;
3. $\tau_j > \text{rea}(I_t^{a_i})$ et $\tau_j > \text{rea}(I_t^{a_j})$: l'information est inutile.

Former des coalitions permet d'augmenter l'intervalle de temps durant lequel une communication est utile. Si les communications sont fréquentes, le deuxième cas ($\tau_j > \text{rea}(I_t^{a_i})$ et $\tau_j \leq \text{rea}(I_t^{a_j})$) est plus fréquent lui-aussi et former des coalitions est moins utile. Le cas extrême est celui où les agents peuvent communiquer en permanence. Dans ce cas, il est inutile de coopérer puisque dès qu'un agent réalise une tâche, les autres peuvent la retirer de leur plan et réaffecter leurs ressources.

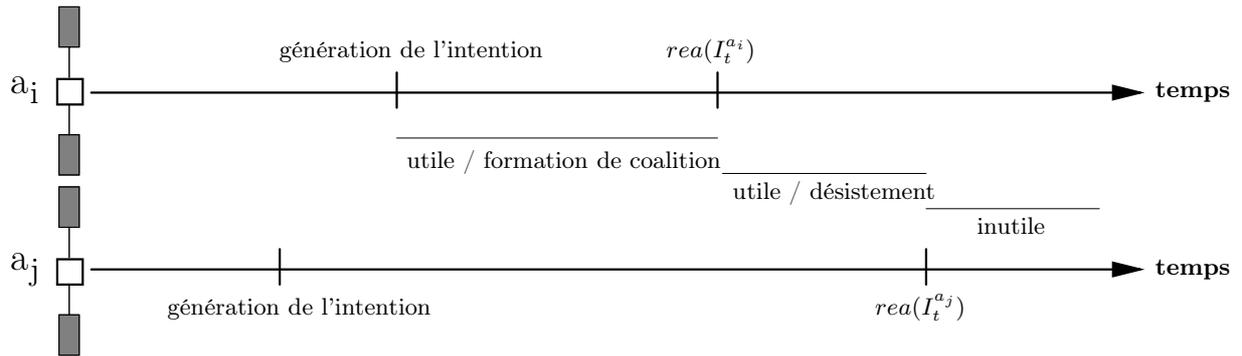


FIGURE 4.6 – Communications possibles entre deux agents en conflit

Dans ce chapitre, nous nous sommes intéressés à la notion de coopération entre les agents. Nous avons proposé d'utiliser la notion de coalitions et avons défini un mécanisme décentralisé de formation de coalitions fondé sur un modèle d'accointances dynamique. Il s'agit d'une approche *anytime* car les agents n'attendent pas qu'une structure de coalitions unique et complète soit calculée. En effet, les agents construisent localement une structure potentielle de coalitions puis raffinent cette structure au fil des communications par un mécanisme d'incitation et de minimisation. Le chapitre suivant sera consacré à l'évaluation de nos propositions.

Chapitre 5

Évaluation : simulations et mesures

Dans ce chapitre, après une première évaluation des protocoles proposés aux chapitres précédents, nous discutons des difficultés induites par le besoin de mesures pertinentes pour évaluer le passage à l'échelle puis nous définissons des critères adaptés aux systèmes multiagents ainsi que des mesures sur ces critères. Des résultats en termes de rapidité d'exécution, de qualité d'exécution et de capacité d'extension sont ensuite donnés.

1 Évaluation des protocoles

1.1 Simulation par événements discrets

En communiquant, les satellites construisent une connaissance commune des tâches à réaliser et des intentions d'autrui concernant ces tâches afin de construire des coalitions. Nous nous intéressons aux performances de cette approche et pour cela nous considérons trois types d'agents en fonction de leurs capacités de coordination :

1. une coordination *a posteriori* où les satellites évitent de réaliser les tâches qui ont déjà été réalisées par d'autres satellites ; il s'agit d'une approche réactive, peu coûteuse en communication et raisonnement ;
2. une coordination *a priori* où les satellites négocient les tâches à réaliser et forment implicitement des coalitions ; il s'agit de l'approche délibérative proposée au chapitre précédent, plus coûteuse en communication et en raisonnement.
3. une approche *sans coordination* qui sert de témoin : les satellites ne communiquent pas entre eux et planifient individuellement les tâches au mieux en fonction de leurs connaissances.

Nous avons implanté en JAVA un simulateur de constellations de satellites qui simule de manière centralisée la constellation distribuée par l'intermédiaire d'un système à

événements discrets. En cas de concurrence entre les actions de plusieurs agents¹, l'agent possédant le numéro d'identification le plus petit est prioritaire. Cette méthode, semblable à celle de [Chow, 1996], permet de réitérer une expérience de manière déterministe.

La machine sur laquelle les simulations ont été réalisées est une SunBlade 1500 équipée d'un processeur Sparc v.9 cadencé à 1062 MHz et d'une mémoire RAM de 1 Go.

Les trois approches (pas de coordination, coordination *a posteriori* et coordination *a priori*) ont été évaluées d'abord sur une constellation comportant 3 satellites. Nous nous intéressons au protocole de communication ainsi qu'aux protocoles de coopération. Les métriques que nous avons retenues sont les sorties de la simulation : nombre de messages échangés et nombres de tâches réalisées.

1.2 Volume des communications

Cette expérience porte sur 3 agents qui communiquent entre eux pendant une exécution simulée de 6 heures, période \mathbb{T}_P de la relation Voisins. La Figure 5.1 indique le nombre de messages échangés en fonction du protocole de coopération (coordination *a posteriori* et coordination *a priori*) et en fonction du nombre de tâches atomiques injectées à l'initialisation du système.

Si le protocole de communication est en théorie polynomial en nombre de tâches atomiques et d'agents (voir Proposition 3.8), en pratique il est linéaire. Ceci est dû au fait qu'un agent ne communique pas une tâche ou une intention qu'il sait déjà connue de son interlocuteur.

La coordination *a posteriori* nécessite un nombre de messages plus faible que la coordination *a priori*. En effet, les intentions ne sont pas transmises. Ainsi, la complexité du protocole de communication devient linéaire en nombre d'agents, au lieu d'être polynomiale en nombre d'agents. Cependant, nous avons vu au Chapitre 4 Section 4.3 qu'il y a un compromis à faire en matière de communication car :

- la coordination *a priori* permet de rendre efficace des communications peu fréquentes mais nécessite plus de messages par communication ;
- la coordination *a posteriori* permet de ne communiquer que peu de messages à la fois mais nécessite des communications fréquentes pour être efficace.

1.3 Tâches réalisées

Deux types de simulations sont exploitées. Les *simulations statiques* dotent les agents d'un nombre fixé de tâches tandis que les *simulations dynamiques* génèrent de nouvelles tâches par l'intermédiaire des stations sol. Aucun agent satellite ne génère de tâches par

1. Par exemple, il y a concurrence si à la date τ un agent a_i doit réaliser une observation tandis qu'à la même date un agent a_j doit télécharger ses images à une station sol.

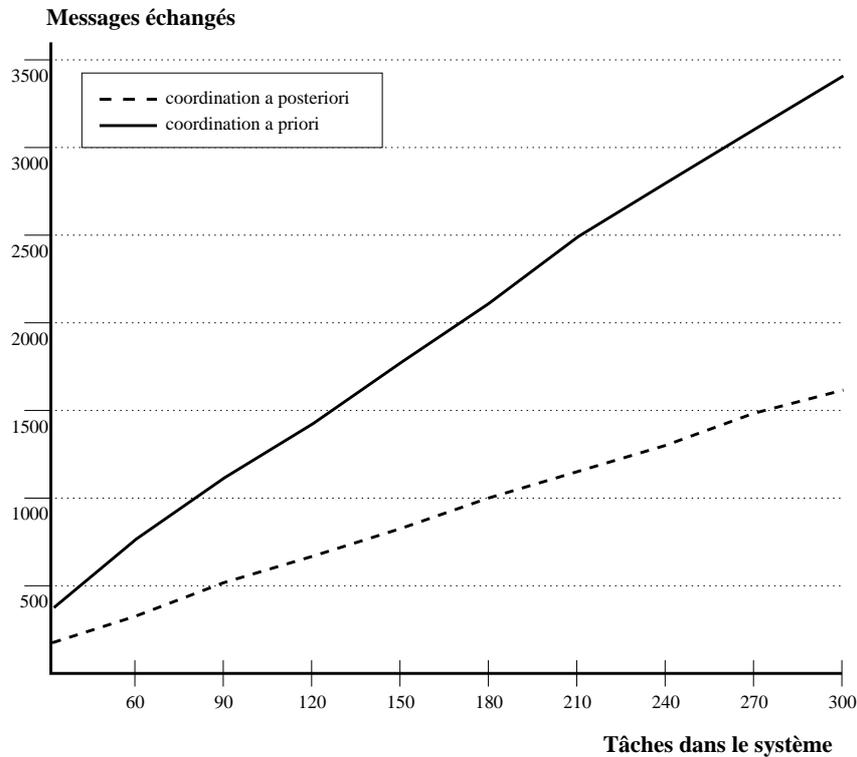


FIGURE 5.1 – Nombre de messages échangés au cours d'une période de 6 heures

lui-même. Deux métriques nous ont permis d'évaluer les résultats: le nombre de tâches réalisées et le nombre de tâches réalisées sans redondances.

Notons que, dans ces expériences, les agents ne sont pas limités en ressources mémoire, c'est-à-dire qu'ils peuvent réaliser autant d'observations que possible entre deux téléchargements étant donné les contraintes d'exclusion mutuelle. Ainsi, la proportion de tâches réalisées sans redondances est significative lorsque, lors d'une mission d'observation réelle, les agents sont contraints en ressources mémoire.

Simulations statiques

Partant d'un scénario comportant 100 tâches atomiques et 3 agents, nous avons progressivement augmenté le nombre de tâches en conflit, c'est-à-dire pouvant être réalisées par au moins deux agents.

Définition 5.1 (Densité)

La densité représente la proportion de tâches atomiques en exclusion mutuelle au sein du système.

Les résultats sont présentés en Figure 5.2.

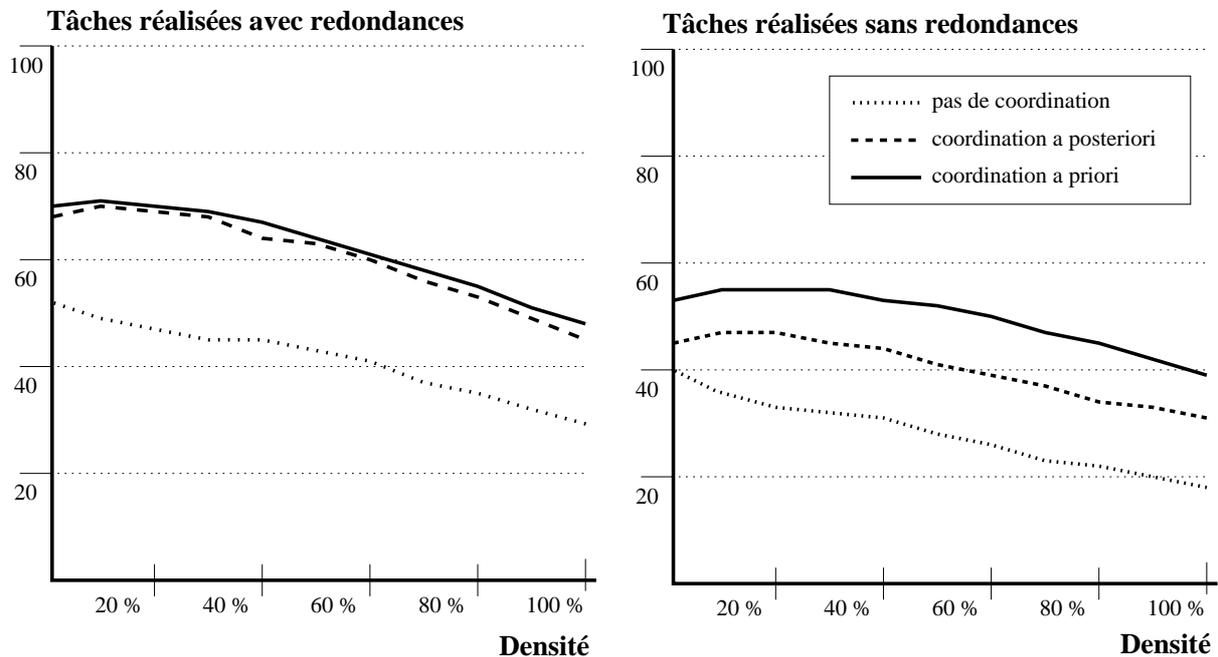


FIGURE 5.2 – Effet de la densité sur une simulation statique

Pour les deux métriques, la coordination *a posteriori* et la coordination *a priori* obtiennent de meilleurs résultats que ceux des simulations sans coordination. Si les deux approches de la coopération réalisent pratiquement le même nombre de tâches (avec un avantage pour la coordination *a priori*), la coordination *a priori* permet d'accroître largement le nombre de tâches réalisées sans redondances.

Il peut sembler contre-intuitif qu'en économisant des ressources *via* la coordination *a priori*, les agents ne puissent pas réaliser plus de tâches. Cela est dû au fait que le mécanisme décentralisé de formation de coalitions est une méthode d'optimisation locale. Ceci peut être mis en lumière en simulant deux agents et un nombre restreint de tâches en conflit. Dans ce cas, l'ensemble des tâches est réalisé sans redondances. Toutefois, en prenant en compte l'arrivée de nouvelles tâches, un horizon temporel non borné et des connaissances initiales dissymétriques entre les agents, les ressources économisées permettent de s'adapter aux aléas.

Simulations dynamiques

Afin d'illustrer ceci, l'expérience suivante est fondée sur un scénario dynamique : toutes les 6 heures, les stations au sol envoient 40 nouvelles tâches complexes (elles-mêmes composées d'au moins 2 sous-tâches atomiques) à l'un des 3 satellites.

Les approches par coordination obtiennent évidemment de meilleurs résultats que des

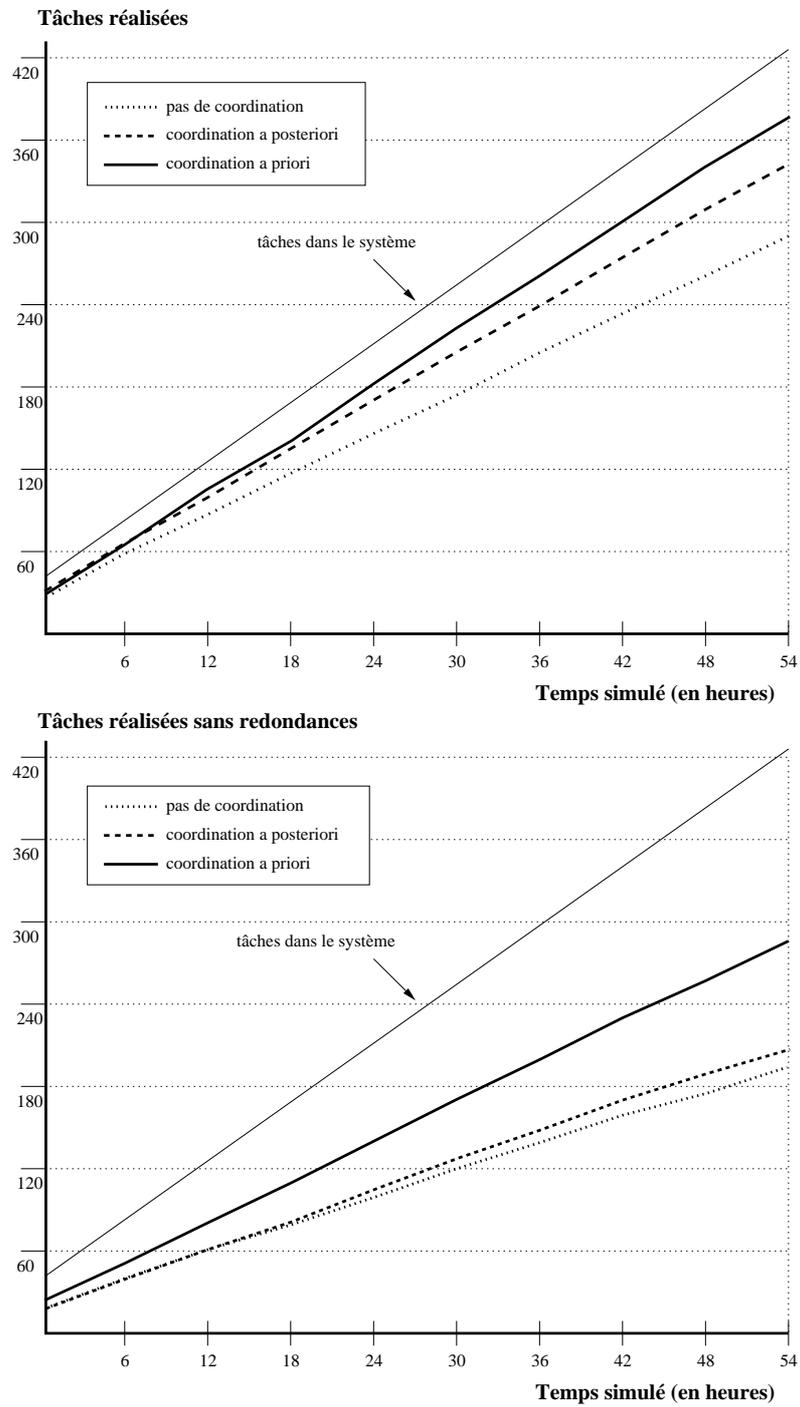


FIGURE 5.3 – Nombre de tâches réalisées (avec et sans redondances)

satellites isolés (sans coordination). Cependant, nous pouvons remarquer que la coordination *a priori* permet à la constellation de réaliser plus de tâches, avec moins de redondances. Au fur et à mesure que le temps passe et que l'environnement se sature en tâches, l'écart entre l'approche *a posteriori* et *a priori* se creuse : la coordination *a priori* permet d'économiser des ressources et de les réaffecter plus tard pour réaliser plus de tâches.

Ainsi, l'économie de ressources dans un contexte dynamique suite à une réaffectation de ces dernières permet de réaliser incrémentalement de plus en plus de tâches. La Figure 5.4 illustre le gain de la coordination *a priori* et de la coordination *a posteriori* relativement aux résultats d'une constellation non coordonnée.

Durant une première phase où les agents construisent une connaissance commune de l'environnement initial, le gain décroît puis suit une progression logarithmique. La coordination *a posteriori* permet d'atteindre un gain d'environ 20% en tâches réalisées et 7,5% en tâches réalisées sans redondances tandis que la coordination *a priori* atteint un gain de 30% en tâches réalisées et 50% en tâches réalisées sans redondance.

Suite à ces premières évaluations, nous devons nous interroger à propos de l'influence du nombre d'agents sur les résultats.

2 Comment évaluer le passage à l'échelle ?

2.1 Problématique

Résoudre un problème *via* une modélisation multiagent permet, dans un premier temps, de n'affecter à chaque agent qu'une partie du problème à résoudre et, dans un second temps, de transformer des contraintes globales en contraintes locales qui sont résolues par coordination entre les agents. Dans les chapitres précédents, nous avons présenté un problème d'affectation décentralisée de tâches au sein d'une constellation de satellites et nous avons proposé une méthode de résolution par formation de coalitions. De par la structure du problème, un unique agent peut le résoudre, mais en un laps de temps important ; l'ajout d'autres agents en accélère la résolution.

Dans un contexte d'agents informatiques, l'ajout d'un nouvel agent est généralement peu coûteux. Ce n'est pas le cas en robotique autonome et moins encore dans le domaine spatial. Se pose alors la question suivante : est-il intéressant d'ajouter un nouvel agent au système ? Cela nous conduit à évaluer la performance d'un système multirobot et sa capacité d'extension², c'est-à-dire sa capacité à « passer à l'échelle », et donc de définir des mesures pertinentes permettant de dégager un compromis entre performance et nombre d'agents robots. Cependant pour un problème donné, augmenter le nombre de satellites - même s'ils ne communiquent pas entre eux - permet de résoudre plus facilement le problème mais trois questions se posent alors :

2. *scalability* en anglais.

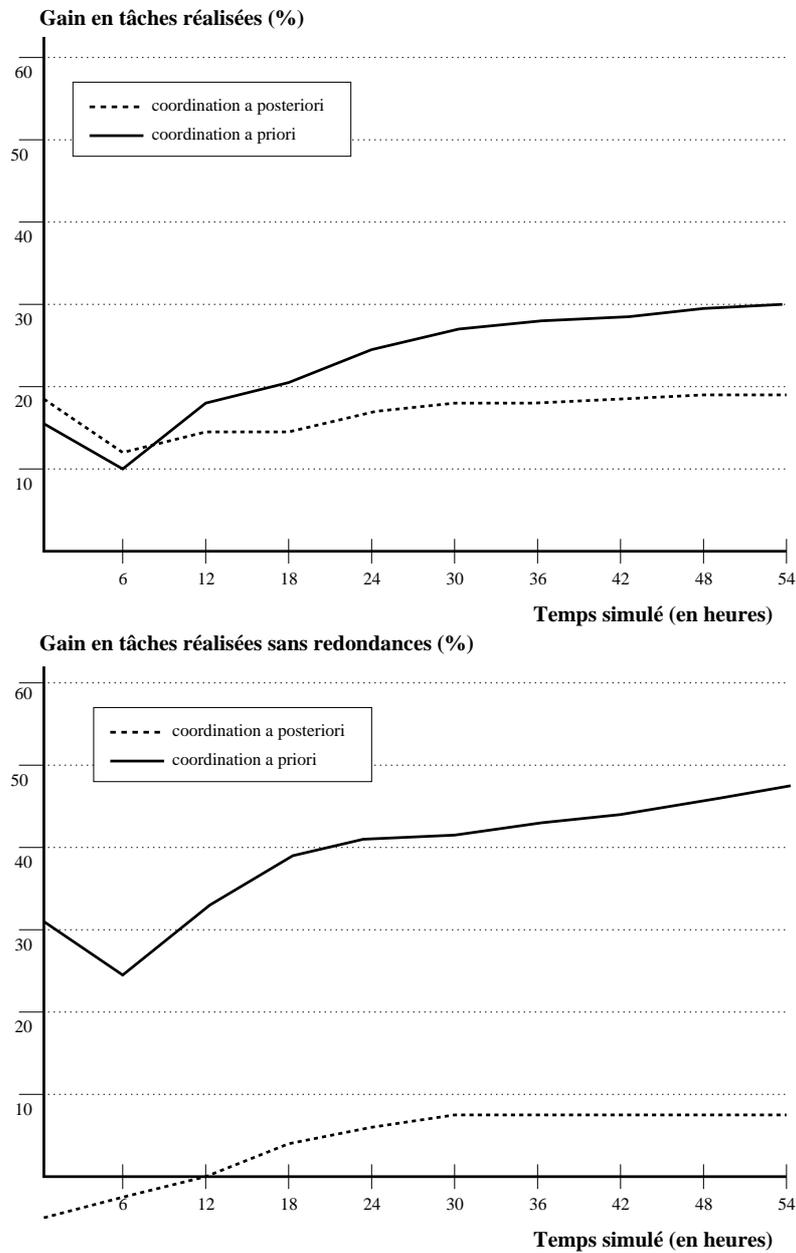


FIGURE 5.4 – Gain en termes de tâches réalisées (avec et sans redondance)

1. quelles sont les mesures pertinentes pour évaluer la performance d'un tel système ?
2. comment prendre en compte le biais dû à la diminution de la complexité d'un problème au fur et à mesure que le nombre d'agents affectés à sa résolution augmente ?
3. à partir de ces deux notions, peut-on définir une relation entre le nombre de satellites mis en œuvre et leur mode de coordination ?

Remarquons qu'il ne s'agit pas d'évaluer la méthodologie qui mène à la construction d'un système multiagent [Bernon *et al.*, 2003, Sturm et Shehory, 2003, Zambonelli *et al.*, 2003] mais d'évaluer le système multiagent en fonctionnement comme le font par exemple [Contet *et al.*, 2008] pour les agents réactifs. [Lee *et al.*, 1998] proposent de le faire selon les trois axes classiques pour évaluer les systèmes distribués : performance, stabilité et capacité d'extension.

2.2 Trois critères

La *performance* du système se mesure par le biais d'indicateurs statistiques comme la consommation en ressources, le temps de réponse, le nombre de tâches réalisées, le temps de calcul et la charge de communication. [Davidsson et Johansson, 2003] définissent par exemple les indicateurs suivants : réactivité du système, répartition des tâches entre les agents, utilisation de toutes les ressources, charge de communication. De leur côté, [Dutta et Sen, 2002] n'en proposent que deux : rapidité d'exécution et qualité d'exécution.

La *stabilité* est liée au concept d'équilibre. Lorsqu'un événement vient influencer sur le système, ce dernier est stable s'il parvient à revenir à un état d'équilibre. C'est un concept à rapprocher de la robustesse fonctionnelle du système selon [Davidsson et Johansson, 2003]. Intuitivement, un système multiagent est à l'équilibre si ses performances restent stationnaires suite à un ou plusieurs événements qui peuvent venir le perturber [Lee *et al.*, 1998]. La notion d'événement doit bien évidemment être définie : il peut s'agir de l'ajout ou du retrait d'agents, de l'ajout de nouvelles tâches, de bruit sur les observations ou de toute autre variation de paramètres au cours de l'exécution.

Si la définition des deux critères précédents est communément admise, ce n'est pas le cas pour la *capacité d'extension* dont la définition est liée à la complexité du système, terme difficile à définir pour les systèmes multiagents comme le font remarquer [Kubera *et al.*, 2007]. Par exemple, [Lee *et al.*, 1998] s'intéressent à la complexité algorithmique et définissent la capacité d'extension comme le fait que la performance au pire cas du système est bornée par une fonction polynomiale de la charge de travail. De leur côté, [Turner et Jennings, 2000] considèrent un système comme extensible si la consommation en ressources par agent est bornée par une fonction polynomiale de la taille du système. Mais pour [Davidsson et Johansson, 2003], la capacité d'extension correspond à la dégradation (ou non) de la performance à mesure que le système croît en nombre d'agents.

La première définition revient en fait à une mesure de performance et ne s'intéresse pas à la dimension multiagent du système. La seconde définition prend en compte cette dimension mais définit un système comme extensible si les agents utilisent individuellement de moins en moins de ressources pour résoudre leur problème. Or dans le domaine qui nous intéresse, il y a une contradiction : si des ressources sont disponibles, elles doivent être utilisées. La troisième définition nous donne une piste de réflexion : nous pouvons dire que la capacité d'extension réfère à la capacité du système à accroître ses performances

alors que sa taille augmente elle aussi. Par exemple, quatre agents sont-ils deux fois plus efficaces que deux agents ?

Nous proposons de définir des mesures sur les notions de performance, stabilité et capacité d'extension pour un système multirobot mis en œuvre ou simulé. Dans les deux cas, nous parlerons d'expérience, que nous définissons formellement.

Définition 5.2 (Expérience)

Une expérience \mathcal{E} est un tuple $\langle E, M, S, \mathbb{T} \rangle$ où $E = \langle e_i \rangle$ est un ensemble de n entrées et $S = \langle s_j \rangle$ est un ensemble de m sorties où les domaines de valeurs de chacun des éléments de E et S sont quelconques ; $\mathbb{T} \subseteq \mathbb{N}$ est une horloge et $M : E \times \mathbb{T} \mapsto S$ est un modèle de la dynamique qui lie E à S .

E est l'ensemble des paramètres initiaux de l'expérience (contraintes sur les objectifs des agents, nombre de tâches connues, nombre d'agents, ressources, conditions d'arrêt de la simulation, etc.) et S l'ensemble des observations en sortie. La fonction M est considérée comme une *boîte noire* qui modélise le système multiagent ainsi que l'environnement. Notons que nous considérons un *temps discrétisé*.

2.3 Performance

Une expérience produit une grande quantité de données dont le résultat dépend de nombreux paramètres (comme le nombre d'agents, les différentes échelles de complexité du problème, les paramètres intrinsèques des agents ou des interactions). Ainsi, il n'est pas trivial de définir de *bonnes mesures* pour évaluer les performances du système. Nous proposons de définir formellement ce qu'est une performance.

Définition 5.3 (Performance)

Soit une expérience \mathcal{E} , un ensemble $V = \{v_i \in E \cup S \cup \mathbb{T} : \exists v_s \in S\}$. Une performance $G(\mathcal{E})$ sur cette expérience est définie par :

$$G(\mathcal{E}) : V \mapsto \mathbb{R}$$

Cette définition nous permet de considérer $G(\mathcal{E})$ comme une fonction quelconque des variables du système. Cependant nous nous limitons aux mesures de performance monovaluées, c'est-à-dire monocritères. La raison tient au fait que nous définissons dans le paragraphe suivant la stabilité et la capacité d'extension en fonction des performances et que, dans un cadre multicritère, un système peut être stable (ou extensible) pour un critère et pas pour un autre. Ceci nous amène à analyser chaque critère, c'est-à-dire chaque performance, séparément.

2.4 Stabilité et capacité d'extension

Trouver des mesures pertinentes lors d'un changement d'échelle est un problème que les sciences physiques traitent depuis longtemps à travers le concept de *grandeur sans dimension*. Ces grandeurs servent à étudier les propriétés intrinsèques des modèles. Le nombre de Reynolds par exemple représente le rapport entre les forces d'inertie et les forces visqueuses en dynamique des fluides. Un autre exemple peut être le nombre de Mach qui exprime le rapport de la vitesse locale d'un fluide sur la vitesse du son dans ce même fluide.

De manière générale, une grandeur sans dimension est une quantité permettant de décrire une caractéristique physique sans unité explicite d'expression. Elle est constituée, comme le rappelle [Giacomo, 1996], du produit ou rapport de grandeurs ayant une dimension, de telle façon que le rapport des unités soit 1.

Nous pouvons définir la stabilité et la capacité d'extension comme des grandeurs sans dimension sur les performances.

Stabilité

Suite à un événement, nous supposons que les performances du système vont se dégrader. Nous mesurons alors la stabilité du système par la vitesse à laquelle il recouvre ses performances initiales après l'événement. Un événement est défini de la manière suivante :

Définition 5.4 (Événement)

Soit une expérience $\mathcal{E} = \langle E, M, S, \mathbb{T} \rangle$. Le système modélisé par M est soumis à un événement δ s'il existe une date $\tau \in \mathbb{T}$ telle que $E(\tau + 1) \neq E(\tau)$. Notons \mathcal{E}^δ l'expérience fondée sur M et subissant l'événement δ .

Dans ce cas, la stabilité est définie comme suit :

Définition 5.5 (Stabilité)

Soit une expérience $\mathcal{E} = \langle E, M, S, \mathbb{T} \rangle$ et soit \mathcal{E}^δ l'expérience fondée sur M subissant l'événement δ à la date $\tau \in \mathbb{T}$. Soit $G(\mathcal{E}(\tau'))$ une performance réalisée à la date $\tau' > \tau$. La stabilité du système est donnée par :

$$\mathcal{S}(\mathcal{E}, \mathcal{E}^\delta, \tau') = \frac{G(\mathcal{E}^\delta(\tau'))}{G(\mathcal{E}(\tau'))}$$

Capacité d'extension

En considérant la capacité d'extension comme la variation des performances du système à mesure que le nombre d'agents augmente, la capacité d'extension est définie par la variation du rapport de la performance du système comportant un nombre d'agents plus grand et de la performance du système comportant un nombre d'agents plus petit, *ceteris paribus*.

Définition 5.6 (Capacité d'extension)

Soit une expérience \mathcal{E}_n portant sur n agents et une performance $G(\mathcal{E}_n)$. La capacité d'extension du système à $(n + p)$ agents ($p \neq 0$) est donnée par :

$$X(n, p) = \frac{\binom{G(\mathcal{E}_{n+p})}{n+p}}{\binom{G(\mathcal{E}_n)}{n}}$$

Une capacité d'extension valant 1 indique que la performance par agent reste inchangée si le nombre d'agents augmente. Si la capacité d'extension est inférieure à 1 cela signifie que les agents ne sont plus aussi performants individuellement (même si le système dans sa globalité peut être plus performant). Une capacité d'extension supérieure à 1 indique qu'augmenter le nombre d'agents permet d'accroître la performance par agent et, par conséquent, d'accroître la performance du système en plus du gain dû au simple nombre d'agents.

Ainsi si une performance $G(\mathcal{E}_n)$ est sous-linéaire³ en fonction du nombre d'agents, le système n'est pas extensible. Si la performance est super-linéaire en fonction du nombre d'agents, le système est extensible.

3 Application aux constellations de satellites

Considérant les spécificités des constellations de satellites, nous présentons dans un premier temps la manière dont les expériences sont mises en œuvre puis nous adaptons les définitions générales des performances, stabilité et capacité d'extension. Nous définissons aussi une mesure de coopération qui permet de dégager le compromis entre nombre de satellites et mode de coordination.

3.1 Mesures de performance

Nous définissons trois mesures de performance: la *réactivité*, c'est-à-dire la capacité de n agents à accomplir plus rapidement qu'un unique agent un ensemble de tâches, l'*efficacité*, c'est-à-dire une mesure des ressources mises en œuvre pour accomplir ces tâches, et l'*activité*, la proportion de tâches réalisées à une date donnée. Formellement,

Définition 5.7 (Réactivité)

Soit une expérience \mathcal{E} et $\mathbb{T}_n \in S$ le temps nécessaire à n satellites pour la mener à bien. La réactivité de la constellation est donnée par :

$$\frac{\mathbb{T}_1}{\mathbb{T}_n}$$

3. Une fonction $f(n)$ est sous-linéaire si $f(n) \in o(n)$, c'est-à-dire $\lim_{n \rightarrow \infty} \frac{f(n)}{n} = 0$.

La réactivité est toujours supérieure à 1 car n satellites ($n > 1$) ne peuvent pas avoir besoin de plus de temps qu'un unique satellite pour réaliser un ensemble de tâches. Ainsi plus la réactivité est élevée, meilleure est la performance de la constellation.

Définition 5.8 (Efficacité)

Soit une expérience \mathcal{E} , $O \subset S$ l'ensemble des observations réalisées et $T \subset S$ l'ensemble des tâches réalisées. L'efficacité de la constellation est donnée par :

$$\frac{|T|}{|O|}$$

Dans notre application, le nombre d'observations $|O|$ correspond au nombre d'images prises par l'ensemble des satellites, comptabilisant ainsi les tâches et leurs éventuelles redondances. Le nombre de tâches $|T|$ correspond au nombre de tâches réalisées par la constellation, à l'exclusion des redondances.

Exemple 5.9

Soit a_i et a_j deux agents. Supposons que a_i a réalisé les tâches t_1, t_2, t_3, t_4 et que a_j a réalisé les tâches t_2, t_3, t_5, t_6 . Ainsi, il y a eu 8 observations et 6 tâches réalisées.

Ainsi le nombre d'observations est toujours supérieur ou égal au nombre de tâches réalisées. L'efficacité est donc toujours inférieure ou égale à 1. Plus elle est élevée, meilleure est la performance de la constellation.

Définition 5.10 (Activité)

Soit une expérience \mathcal{E} , $\mathcal{T} \subset E$ l'ensemble des tâches dans le système et $T_\tau \subset S$ l'ensemble des tâches réalisées à la date $\tau \in \mathbb{T}$. L'activité du système multiagent est :

$$A(\mathcal{E}, \tau) = \frac{|T_\tau|}{|\mathcal{T}|}$$

Le nombre de tâches $|T_\tau|$ correspond au nombre de tâches réalisées par la constellation à la date τ , à l'exclusion des redondances. Cela correspond donc à la quantité de travail qui a été effectuée par la constellation à la date τ . L'activité est donc toujours inférieure ou égale à 1. Plus elle est élevée, plus la constellation est performante à la date courante.

3.2 Mesure de stabilité

Les événements que nous considérons sont les arrivées de nouvelles tâches dans le système.

Définition 5.11 (Événement)

Soit une expérience $\mathcal{E} = \langle E, M, S, \mathbb{T} \rangle$ et $\mathcal{T}(\tau) \subset E$ l'ensemble des tâches dans le système à la date τ . Il y a un événement δ à la date $\tau \in \mathbb{T}$ si $|\mathcal{T}(\tau + 1)| > |\mathcal{T}(\tau)|$. Notons \mathcal{E}^δ cette expérience subissant l'événement δ .

Nous définissons alors la stabilité de l'activité.

Définition 5.12 (Stabilité)

Soit une expérience $\mathcal{E} = \langle E, M, S, \mathbb{T} \rangle$ et soit \mathcal{E}^δ l'expérience fondée sur M subissant l'événement δ à la date $\tau \in \mathbb{T}$. Soit $A(\mathcal{E}, \tau)$ l'activité à la date τ . La stabilité du système est donnée par :

$$\mathcal{S}(\mathcal{E}, \mathcal{E}^\delta) = \frac{A(\mathcal{E}^\delta, \tau + 1)}{A(\mathcal{E}, \tau + 1)}$$

La stabilité représente la dégradation de l'activité de la constellation lorsque la charge de travail (le nombre de tâches à réaliser) augmente. Cette mesure est toujours inférieure à 1. Plus elle est élevée, meilleure est la résistance de la constellation à une augmentation de la charge de travail. Ainsi, une valeur de 1 signifie que l'augmentation de cette charge de travail n'a aucune influence sur l'activité de la constellation.

3.3 Mesure de la capacité d'extension

Dans notre application, nous nous intéressons à des constellations de taille 1, 4, 6, 8, 9, 12 et 16 respectivement. La capacité d'extension pour la réactivité et l'efficacité exprime alors l'intérêt à passer d'une taille de constellation à la suivante, comme par exemple de 1 à 4 satellites ou de 9 à 12 satellites. Dans notre cas, cette mesure reste inchangée par rapport à la Définition 5.6.

3.4 Une nouvelle mesure : mesure de coopération

Pour rappel, nous nous intéressons dans notre application aux deux types de coordination suivants :

1. réactive *via* une coordination *a posteriori* où les satellites évitent de réaliser les tâches qui ont déjà été réalisées par d'autres satellites ;
2. délibérative *via* une coordination *a priori* où les satellites négocient les tâches à réaliser et forment implicitement des coalitions.

Dans notre application, nous devons définir une mesure pour mettre en évidence le gain à coordonner *a priori* des satellites qui se coordonnaient *a posteriori*, malgré un coût de communication et de raisonnement supplémentaire. Formellement,

Définition 5.13 (Mesure de coopération)

Soit une expérience \mathcal{E}_k impliquant k satellites qui se coordonnent *a posteriori* et une mesure de performance G . Soit une expérience \mathcal{E}_n identique si ce n'est qu'elle implique n satellites qui se coordonnent *a priori*. La capacité de coopération de la constellation est donnée par :

$$C(\mathcal{E}_k) = \min\{n : G(\mathcal{E}_n) \geq G(\mathcal{E}_k)\}$$

La capacité de coopération indique combien de satellites coordonnés *a priori* suffisent pour être aussi performants qu'une constellation composée de satellites coordonnés *a posteriori*. Plus cette mesure est faible, plus la taille de la constellation peut être réduite pour une performance identique.

4 Évaluation pour le passage à l'échelle

Nous considérons des expériences sur un scénario comportant 500 tâches initiales et des constellations de Walker [Walker, 1984] de tailles différentes (1, 4, 6, 8, 9, 12 ou 16 satellites régulièrement répartis sur un nombre fini de plans orbitaux). Une expérience se termine lorsque toutes les tâches ont été réalisées. Les performances sont ensuite mesurées.

4.1 Réactivité

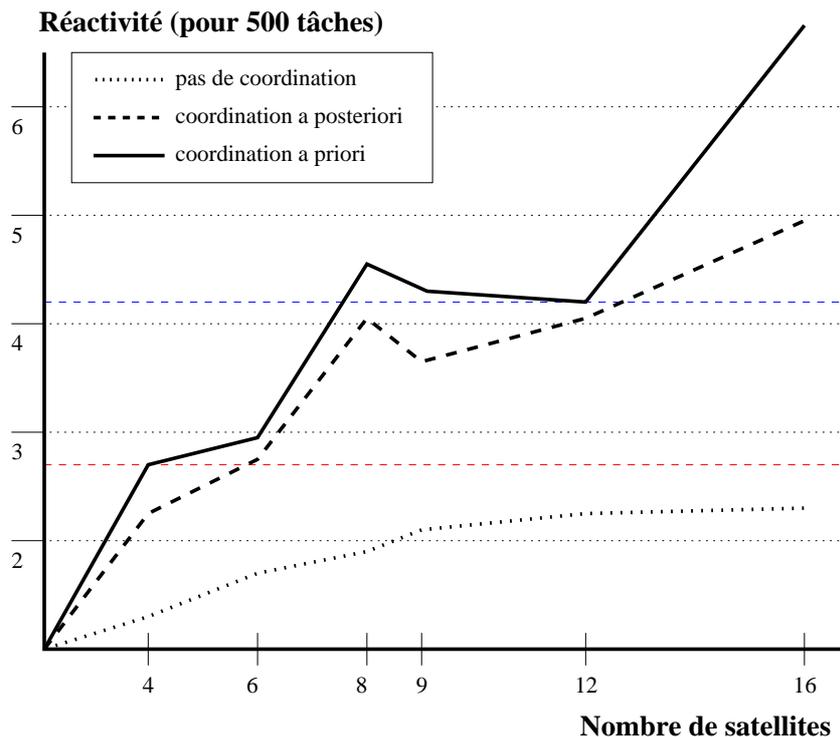


FIGURE 5.5 – Réactivité de la constellation

Sur la Figure 5.5, nous pouvons remarquer que la réactivité sans coordination suit une courbe logarithmique tandis que la coordination (*a posteriori* ou *a priori*) ne comporte pas une telle régularité. Cela est dû à la structure non homogène des interactions entre les satellites. En effet, des satellites qui ne sont pas coordonnés sont toujours homogènes

dans le sens où il n'y a aucune interaction entre eux. En revanche, lorsqu'il y a interaction, celle-ci ne peut exister qu'entre des satellites placés sur des plans orbitaux différents. La Figure 5.6 représente quelques-uns de ces plans.

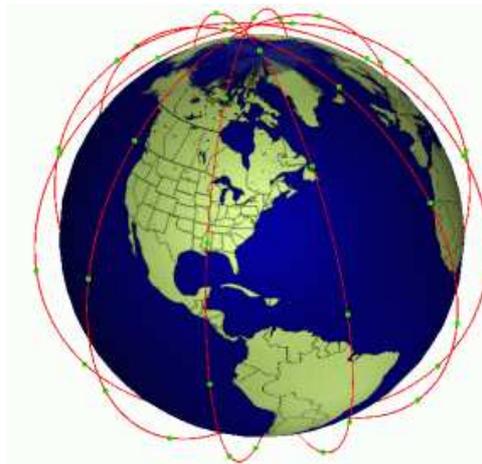


FIGURE 5.6 – Différents plans orbitaux

Ainsi, 4 plans de 2 satellites peuvent permettre plus d'interactions que 3 plans de 4 satellites. Il y a donc une notion de topologie des interactions que l'on ne trouve pas dans d'autres applications multiagents. En fait le nombre de satellites n'est pas le facteur déterminant dans le nombre d'interactions, il s'agit plutôt de leurs orbites : peu de satellites peuvent communiquer souvent tandis que de nombreux satellites peuvent ne communiquer qu'épisodiquement. En conséquence, cette topologie des interactions influe sur la performance de la constellation. Ce phénomène peut être observé dans la plage de 8 à 12 satellites pour les constellations coordonnées *a priori*. Sur la Figure 5.5, la ligne bleue illustre la perte de réactivité lorsque l'on passe de 8 à 12 satellites.

Le tableau de la Figure 5.7 donne la capacité d'extension du système coordonné *a priori* pour le scénario précédent selon le critère de réactivité. Nous pouvons remarquer que passer de 6 à 8 satellites, tout comme passer de 12 à 16 satellites, permet d'obtenir une réactivité supplémentaire qui provient des interactions entre les satellites. Ainsi mettre en œuvre 8 satellites comparativement à 6 permet non seulement d'obtenir le gain de performance attendu par rapport au nombre d'agents, mais d'obtenir une réactivité accrue de 16 %. De même, passer de 12 à 16 satellites permet d'accroître la réactivité de 26 % en plus du gain dû au nombre d'agents.

Cependant nous avons des cas sous-linéaires, c'est-à-dire des cas où le système n'est pas extensible pour la coordination *a priori*. Ceci est dû, une nouvelle fois, à la topologie des interactions entre les agents. En effet si ajouter un nouvel agent n'augmente pas proportionnellement le nombre d'interactions positives au sein du système, ce dernier ne va pas accroître suffisamment sa réactivité pour être considéré comme extensible.

Nb. de satellites (coord. a priori)		$X(n, p)$
n	$n + p$	
1	4	0,67
4	6	0,73
6	8	1,16
8	9	0,85
9	12	0,74
12	16	1,26

FIGURE 5.7 – Capacité d'extension (réactivité)

Le tableau de la Figure 5.8 indique la mesure de coopération du système selon le critère de réactivité. Les valeurs ont été arrondies à l'entier le plus proche afin de prendre en compte la granularité du nombre d'agents. Nous pouvons remarquer que 4 satellites coordonnés *a priori* suffisent pour être aussi réactifs que 6 satellites coordonnés *a posteriori*, comme l'illustre la ligne rouge sur la Figure 5.5. De même, 8 satellites coordonnés *a priori* sont au moins aussi réactifs que 12 satellites coordonnés *a posteriori*. L'économie au niveau de la mise en œuvre de la constellation est donc importante.

Nb. de satellites k	$C(\mathcal{E}_k)$
4	4
6	4
8	8
9	8
12	8
16	16

FIGURE 5.8 – Mesure de coopération (réactivité)

Trois cas d'équivalence peuvent cependant être remarqués, signifiant qu'une réduction du nombre de satellites coordonnés *a priori* entraîne une dégradation de la réactivité par rapport à l'utilisation d'une constellation de plus grande taille composée de satellites coordonnés *a posteriori*. Dans ces cas d'équivalence, il est nécessaire, pour choisir le type de satellites à utiliser, de se poser la question suivante : la réactivité supplémentaire dégagée par la coordination *a priori* compense-t-elle le coût de développement de tels satellites ?

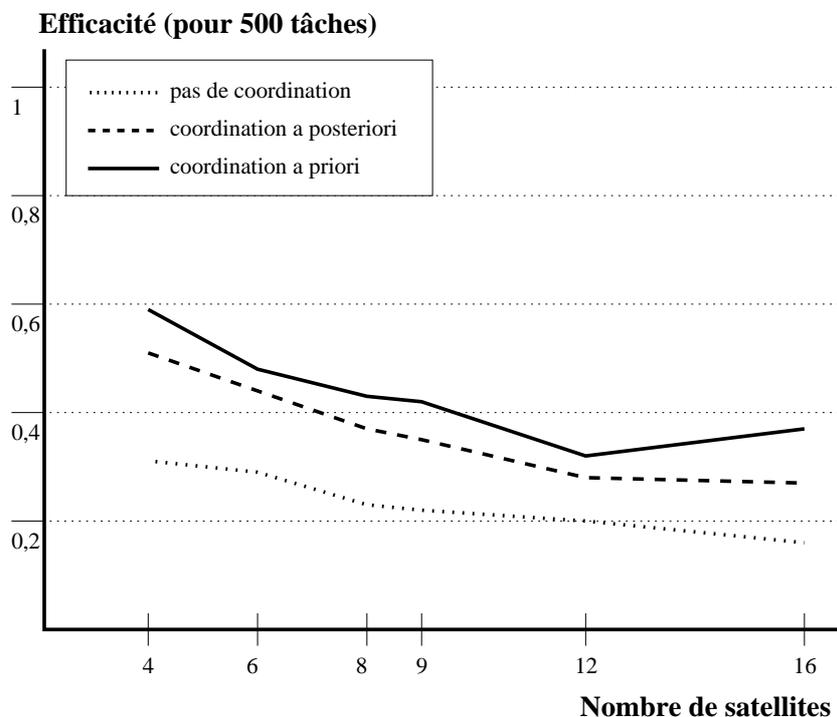


FIGURE 5.9 – Efficacité de la constellation

4.2 Efficacité

La Figure 5.9 nous permet de remarquer que la coordination *a priori* est en moyenne plus efficace de 5% que la coordination *a posteriori* et de 19% par rapport à l'absence de coordination. Ceci correspond à une économie en termes de mémoire de masse utilisée pour stocker les images. Si l'efficacité décroît au fur et à mesure que le nombre de satellites augmente, cela signifie que le système coordonné *a priori* n'est pas extensible selon notre définition (se reporter au tableau de la figure 5.10). En effet, le gain en termes de ressources n'est pas proportionnel au nombre d'agents. Nous pouvons cependant remarquer que le système est extensible au sens de Turner [Turner et Jennings, 2000]. En effet, la consommation en ressources (et plus précisément la surconsommation en ressources au niveau de la constellation) peut être mesurée par le nombre d'observations divisé par le nombre de tâches réalisées $\frac{|O|}{|T|}$. Or, c'est l'inverse de l'efficacité. Ainsi, la consommation en ressources peut être bornée par une fonction polynomiale en le nombre d'agents.

Le tableau de la Figure 5.11 indique la mesure de coopération du système selon le critère d'efficacité. Étant donné que l'efficacité est décroissante à mesure que le nombre d'agents s'accroît, un nombre minimal d'agents coordonnés *a priori* sera toujours plus efficace qu'un nombre quelconque d'agents coordonnés *a posteriori*. Ainsi la coordination *a priori* est toujours intéressante pour l'efficacité de la constellation. Cependant, il convient

Nb. de satellites (coord. a priori)		$\mathbf{X}(\mathbf{n}, \mathbf{p})$
\mathbf{n}	$\mathbf{n} + \mathbf{p}$	
1	4	0, 15
4	6	0, 19
6	8	0, 15
8	9	0, 16
9	12	0, 1
12	16	0, 08

FIGURE 5.10 – Capacité d’extension (efficacité)

de se poser la question suivante : le gain d’efficacité de 5% procuré par la coordination *a priori* compense-t-il le coût de développement de cette approche ?

Nb. de satellites k	$\mathbf{C}(\mathcal{E}_k)$
4	4
6	4
8	4
9	4
12	4
16	4

FIGURE 5.11 – Mesure de coopération (efficacité)

4.3 Stabilité

Le scénario utilisé pour mesurer la stabilité d’une constellation comporte 250 tâches initiales. Après 6 heures de temps simulé, un événement ajoute 250 tâches au système puis la stabilité est mesurée 6 heures après. L’accroissement de la charge de travail est représenté en Figure 5.12.

Nous pouvons remarquer sur la Figure 5.13 deux phénomènes : la stabilité croît dans un premier temps puis devient constante ; et les coordinations *a priori* et *a posteriori* sont aussi stables l’une que l’autre en termes d’activité.

Le premier résultat vient du fait que la stabilité dépend en partie de la capacité des agents à construire une connaissance commune en termes de tâches à réaliser et de tâches déjà réalisées. Ajouter des agents permet de transmettre plus rapidement de l’information et de bâtir plus vite cette connaissance commune. Cependant au-delà d’un certain nombre

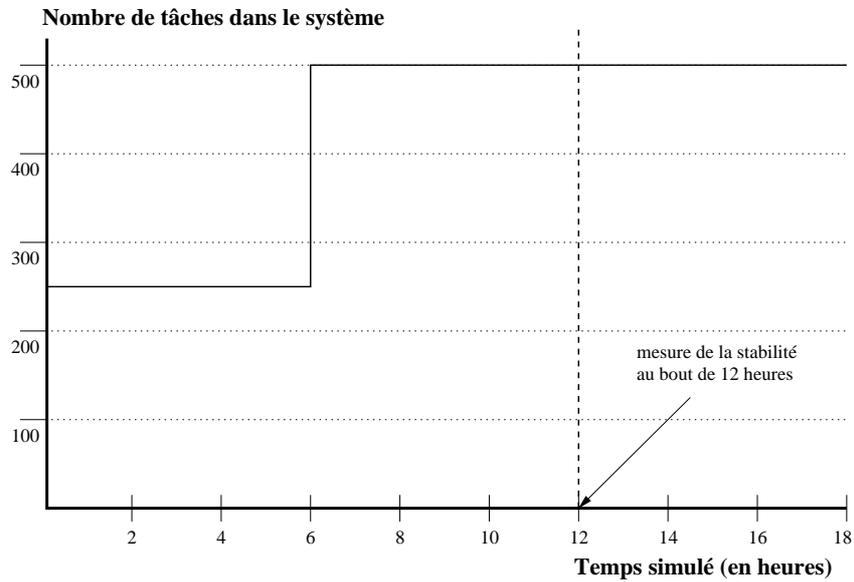


FIGURE 5.12 – Charge de travail de la constellation

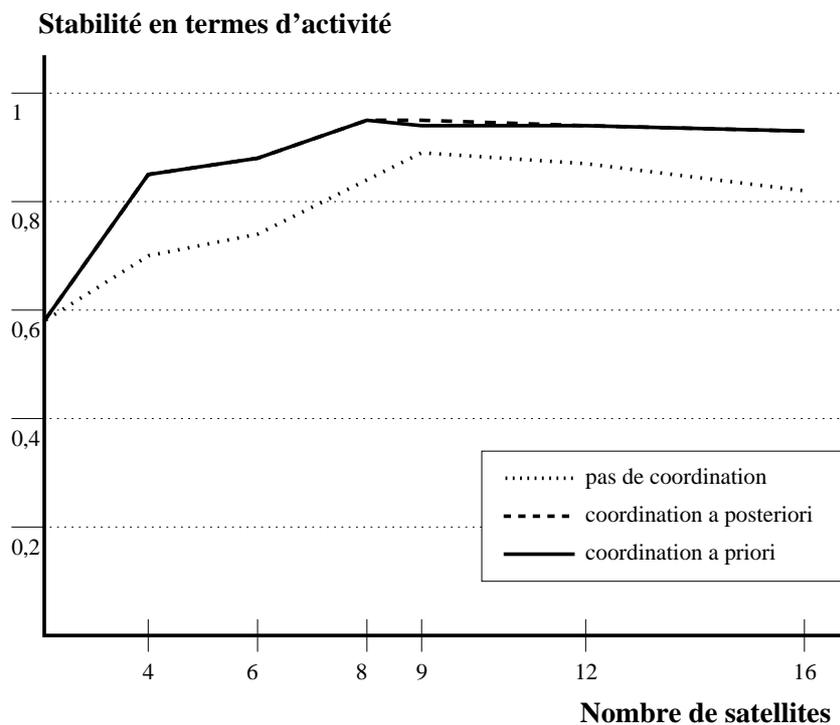


FIGURE 5.13 – Stabilité de la constellation

d'agents, ajouter un nouvel agent n'a pas d'influence sur la vitesse de propagation des informations.

Le deuxième résultat provient de deux phénomènes qui interagissent. La coordination *a priori* est plus efficace (se reporter à la section 4.2) et cette économie de ressources permet d'absorber les perturbations de la charge de travail. Cependant ce type de coordination pousse les agents à prendre des engagements qui ne peuvent être remis en cause et qui sont donc sensible aux événements. Ces deux phénomènes conjoints peuvent expliquer l'équivalence entre la coordination *a priori* et *a posteriori* en termes de stabilité.

Nb. de satellites (coord. a priori)		$X(n, p)$
n	$n + p$	
1	4	0,37
4	6	1,02
6	8	0,81
8	9	0,95
9	12	0,75
12	16	0,74

FIGURE 5.14 – Capacité d'extension (stabilité)

Le tableau de la Figure 5.14 indique la capacité d'extension du système en termes de stabilité. Passer de 4 à 6 satellites permet de légèrement accroître la stabilité au-delà de ce qui est attendu. La capacité d'extension décroît par la suite, atteignant une valeur maximale comme présenté sur la Figure 5.13.

Nb. de satellites k	$C(\mathcal{E}_k)$
4	4
6	6
8	8
9	9
12	12
16	16

FIGURE 5.15 – Mesure de coopération (stabilité)

Le tableau de la Figure 5.15 indique la mesure de coopération du système en termes de stabilité. Comme la mesure de stabilité est identique pour les constellations coordonnées *a*

priori et *a posteriori*, la mesure de coopération est égale au nombre de satellites considérés. Il n'y a pas de gain à coordonner *a priori* en ce qui concerne la stabilité du système.

Il est cependant intéressant de noter que la coordination *a posteriori*, en tant qu'approche réactive, est naturellement stable car le coût de réorganisation au sein de la constellation est nul. Or comme il y a équivalence en termes de stabilité entre la coordination *a priori* et *a posteriori*, nous pouvons dire que l'approche par formation décentralisée de coalitions voit son coût de réorganisation compensé par son gain.

Nous avons proposé dans ce chapitre une manière de formaliser des mesures sur des notions de performance, de stabilité et capacité d'extension. Afin mettre en évidence le compromis nécessaire entre nombre d'agents d'une constellation et extension des performances, nous avons proposé des mesures de *réactivité*, d'*efficacité*, de *stabilité* et de *capacité de coordination*. Nous pouvons remarquer d'une part qu'un système multirobot n'est donc pas extensible en soi mais selon un critère donné. D'autre part, plusieurs critères peuvent être contradictoires.

Par exemple dans notre application, si on désire un système stable en termes d'activité, on préférera une constellation ayant un faible nombre de satellites coordonnés *a posteriori* tandis que si on privilégie la réactivité, on préférera une constellation de grande taille coordonnée *a priori*. De même si la limitation en termes de ressources est une contrainte forte, il est préférable d'utiliser une coordination *a priori*.

Cependant ces résultats se fondent sur le nombre de satellites mis en œuvre et non pas sur le nombre d'interactions au sein de la constellation. Une perspective d'étude serait de considérer la capacité d'extension d'un système non pas en fonction du nombre d'agents qui le composent mais en fonction de la quantité d'interactions entre les agents.

Chapitre 6

Conclusion et perspectives

Dans ce dernier chapitre, nous récapitulons l'ensemble de nos contributions ainsi que nos résultats d'évaluation. Nous proposons ensuite deux perspectives de travail concernant l'analyse du modèle d'accointances et la gestion des défaillances au sein de la constellation.

1 Contrôler une constellation de satellites

Dans ce manuscrit, nous nous sommes intéressé au contrôle autonome d'une constellation de satellites qui communiquent entre eux par liaisons intersatellites. Chaque satellite de la constellation est considéré comme *autonome* dans le sens où :

1. il dispose de **connaissances privées** en termes de tâches à réaliser ;
2. il est capable de planifier pour lui-même ;
3. il représente son plan sous formes d'intentions concernant les tâches ;
4. ces intentions sont, au même titre que les tâches, une connaissance privée.

Dans ce contexte, notre problématique était de concevoir des mécanismes de coopération et de coordination pour la constellation, en tenant compte des contraintes de communication, et permettant à la constellation de s'affranchir du segment sol et de réaliser conjointement des **tâches complexes**. Eu égard à ces contraintes, une approche centralisée n'était pas envisageable, ni désirée et nous avons opté pour une approche décentralisée où :

1. les agents construisent une connaissance commune des tâches et des intentions ;
2. un mécanisme de coopération et de coordination décentralisé permet aux agents de tirer parti du reste de la constellation.

1.1 Du modèle d'acointances à la congrégation

Nous avons proposé une modélisation d'une constellation de satellites à l'aide d'un système multiagent dont le modèle d'acointances, fondé sur les orbites des satellites, possède deux propriétés principales : il est d'une part, **périodique** et d'autre part, **public**.

Ceci nous a permis de définir une constellation de satellites comme une *congrégation* : un ensemble d'agents partageant un même modèle d'acointances. Cette structure organisationnelle abstraite fonde la définition d'une notion de connaissance commune au sein de la constellation et de confiance en cette connaissance. En effet, le fait que le modèle d'acointances soit périodique et public permet à chaque agent de :

1. déterminer la manière dont les connaissances privées se propagent au sein de la constellation et concevoir une connaissance commune sur ces connaissances privées en se fondant sur le fait que les autres agents raisonnent de la même manière ;
2. déterminer la quantité de communications qu'un agent peut avoir entre deux dates et, en considérant que les communications procurent de nouvelles connaissances susceptibles de remettre en cause les intentions d'un agent, de définir ainsi une confiance dans les intentions courantes de ce dernier.

1.2 De la congrégation aux coalitions

Afin de construire un mécanisme de coopération et de coordination décentralisé, nous nous sommes intéressé à la **formation de coalitions**. Ces structures organisationnelles constituent une notion pertinente pour définir des groupes d'agents devant collaborer pour un objectif donné sur un intervalle de temps restreint. En tenant compte des contraintes de notre application, nous avons proposé un mécanisme décentralisé de formation de coalitions.

À partir de leurs connaissances sur les intentions d'autrui, chaque agent construit une structure potentielle de coalitions. Chaque coalition a pour objectif une tâche complexe (et par conséquent l'ensemble des sous-tâches atomiques qui la composent). Ces coalitions sont ensuite raffinées de deux manières :

1. un mécanisme d'**incitation** permet aux agents de rejoindre une coalition en cours de formation. Pour cela, les priorités des tâches objectifs de la coalition sont renforcées au fur et à mesure que de nouveaux agents la rejoignent. Ainsi, plus une coalition est proche d'être complétée (et viable), plus les agents sont incités à la compléter ;
2. un mécanisme d'**optimisation** permet aux agents de quitter une coalition s'il ne sont pas utiles, c'est-à-dire s'ils n'accroissent pas le pouvoir de la coalition. En fonction des capacités de communication des agents et d'une notion de conflits locaux, la taille de la coalition est minimisée pour économiser des ressources et permettre aux agents de rejoindre d'autres coalitions.

Le processus de formation de coalitions est un processus décentralisé incrémental. Il s'inscrit dans une boucle **planification - communication - formation de coalitions** :

1. chaque agent planifie localement ses activités en fonction de ses connaissances ;
2. à chaque communication, de nouvelles connaissances viennent remettre en cause le plan courant et déclenchent une phase de formation de coalitions ;
3. la formation de coalitions vient fixer une partie du plan des agents *via* des **engagements** et relaxer des contraintes de ressources mémoire et d'exclusions mutuelles *via* des **désistements forts** ;
4. le processus de planification individuelle est relancé pour réaffecter les ressources économisées.

1.3 Apport de notre approche

Nous avons évalué notre approche en nous comparant avec deux autres approches :

1. une méthode de coordination *a posteriori* (ou réactive) où les agents ne communiquent que des connaissances concernant les tâches à réaliser et abandonnent toute tâche qui déjà été réalisée par un autre agent ;
2. une méthode témoin où les agents ne communiquent ni coopèrent : ils se contentent de construire des plans individuels optimaux en fonction de leurs connaissances des tâches du système.

La coordination permet d'économiser des ressources et de les réaffecter lors du processus de planification. Nous avons constaté que la coordination *a posteriori* permet d'accroître de 20% le nombre de tâches réalisées dont un gain de 7,5% en tâches réalisées sans redondances. **Notre approche permet d'accroître de 30% le nombre de tâches réalisées avec un gain de 50% en tâches réalisées sans redondances.**

Nous avons aussi pu vérifier expérimentalement que **l'utilisation de liaisons inter-satellites permettait aux satellites de recevoir des requêtes d'observation avant même d'être en visibilité de la station sol**. En effet, un satellite peut recevoir une requête du sol et la propager à toute la constellation avant qu'un certain nombre d'agents soient à leur tour en visibilité de la station.

Afin d'effectuer un « passage à l'échelle », nous avons formalisé des notions d'expérience, performances, stabilité et capacité d'extension. Nous avons aussi proposé une mesure de coopération permettant de mettre en lumière un équilibre entre approches délibératives et réactives (dont le coût de développement des agents individuels est plus faible). Nous avons pu constater expérimentalement que :

1. en termes de stabilité, notre approche et la coordination *a posteriori* sont aussi performantes l'une que l'autre ;

2. en termes d'efficacité, notre approche permet de diminuer de 5% les ressources mémoire consommées par rapport à une coordination *a posteriori* et de 19% par rapport à l'absence de coordination ;
3. en termes de réactivité, c'est-à-dire de temps nécessaire à réaliser un ensemble donné de tâches, notre approche est non seulement plus performante mais permet de **réduire le nombre de satellites nécessaires** comparativement à une coordination *a posteriori*.

2 Perspectives de travail

Cependant, notre travail comporte une limite intrinsèque. La connaissance commune, la confiance en cette connaissance et la résolution des conflits lors de la formation des coalitions sont fondées sur la relation Voisins. Ce sont les capacités de communication des agents et leur structure particulière qui permet la coopération. Qu'arrive-t-il si une liaison intersatellite se rompt ?

C'est pour répondre à cette question que trois perspectives de travail s'offrent à nous :

1. l'étude de la structure de la relation Voisins et de méthodes pour l'analyser. Nous pensons notamment à quantifier les interactions au sein du système. En effet, la quantité d'interactions est corrélée à la mise en commun des ressources au sein d'un système multiagent. Il serait donc intéressant de travailler sur la capacité d'extension d'un système non pas uniquement selon le nombre d'agents mais aussi selon la quantité d'interactions ;
2. la prise en compte des défaillances en termes d'aléas de mission ou de ruptures de communications comme les définissent [Laborde *et al.*, 2006]¹. Cela nous conduit à envisager un mécanisme décentralisé de *suivi de situation* :
 - (a) si un agent ne peut pas tenir ses propositions ou ses engagements et qu'il se désiste au dernier moment (en raison d'une défaillance du planificateur individuel ou du module de gestion de la mémoire), il s'agit d'un *aléa de mission*. L'agent va alors perturber la coopération au sein de la constellation en générant des intentions intenables. Afin de tenir compte de ce type de défaillance, il serait intéressant de travailler sur une notion de confiance qui ne qualifierait plus seulement une proposition en fonction du temps mais une confiance qui dépendrait aussi de l'agent qui formule la proposition ;

1. [Laborde *et al.*, 2006] ont défini quatre classes d'anomalies : les anomalies *absorbantes* qui font échouer la mission et qui ne peuvent pas être corrigées ; les anomalies de *trajectoire* qui provoquent un risque de collision et qui nécessitent une manœuvre d'évitement ; les *aléas de mission* qui modifient les objectifs de la mission ou leur réalisation ; les *ruptures de communications* qui affectent un sous-ensemble du système. Nous ne nous intéresserons qu'aux aléas de mission et aux ruptures de communication. En effet, les problématiques de maintien à poste se trouvent hors de notre champ de recherche. Quant aux anomalies absorbantes, elles ne peuvent pas être traitées par nature.

- (b) en cas de défaillance d'une liaison intersatellite, le satellite concerné ne peut plus transmettre ou recevoir de connaissances. Il ne peut donc plus faire partie de la relation Voisins (qui doit être par définition symétrique) et ne peut plus être utilisé pour calculer la connaissance commune, la confiance en cette connaissance et les conflits au sein des coalitions. Afin de rendre notre approche robuste à ces défaillances, il serait intéressant de définir la relation Voisins non plus comme une connaissance publique mais comme une connaissance privée que les agents pourraient mettre à jour.

Une approche décentralisée du contrôle d'une constellation de satellites permet d'accroître les performances du système tout en garantissant une autonomie individuelle à chacun des satellites. Cette approche se fonde sur les propriétés du modèle d'acointances, spécifiques au domaine spatial. Il est intéressant de prendre en compte de possibles défaillances au sein du modèle, qui constituent une connaissance privée au même titre que les tâches et les intentions des agents.

Annexe A

Propositions pour les perspectives

Dans cette annexe, nous reviendrons plus longuement sur les perspectives présentées en conclusion et nous proposerons pour chacune des pistes de réflexions. Nous rappelons que ces perspectives sont : (1) l'analyse du modèle d'accointances ; (2) la gestion des ruptures de communication ; (3) la gestion des aléas de mission.

1 Analyser le modèle d'accointances

La question de la complexité des interactions au sein des systèmes multiagents n'est pas triviale. Traditionnellement, la complexité représente la difficulté à décrire le problème et constitue une mesure de l'information algorithmique du système. Cependant, nous pourrions nous inspirer de la complexité structurale définie en chimie. En effet, la chimie définit la complexité structurale comme des mesures sur des graphes représentant les interactions entre molécules. Chaque molécule est représentée par un sommet et chaque interaction par un arc. Formellement,

Définition A.1 (Complexité structurale)

Soit un graphe de connectivité $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. La complexité structurale de \mathcal{G} est :

$$C(\mathcal{G}) = \sum_{v_i \in \mathcal{V}} (\deg(v_i) + \sum_{v_j \in \mathcal{V}, v_i \neq v_j} \frac{\deg(v_j)}{d(v_i, v_j)})$$

où $d : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}^+$ est une fonction de distance définie entre les sommets et $\deg(v_i)$ est le degré du sommet v_i .

[Abubakr et Egerstedt, 2005a, Abubakr et Egerstedt, 2005b, Abubakr et Egerstedt, 2004] ont défini une notion de complexité structurale croissante avec les interactions locales des agents sur des *graphes de connectivité*. Ces graphes sont des graphes dynamiques

qui représentent les interactions locales au sein d'un ensemble d'agents communicants spatialement localisés. Par interactions locales, les auteurs entendent communications et perceptions qui portent la même information et sont donc deux facettes de la même pièce¹.

En nous inspirant de ces travaux, nous pouvons définir formellement une notion de complexité au sein du graphe de voisinage.

Définition A.2 (Complexité des interactions)

Soit une constellation \mathcal{S} et son graphe de voisinage \mathcal{V} . La complexité des interactions de \mathcal{V} est :

$$C(\mathcal{V}) = \sum_{a_j \in \mathcal{A}} \sum_{a_i \neq a_j} |v_{ij}| \cdot F_{i,j}$$

où $F_{i,j}$ est la complexité des protocoles entre a_i et a_j .

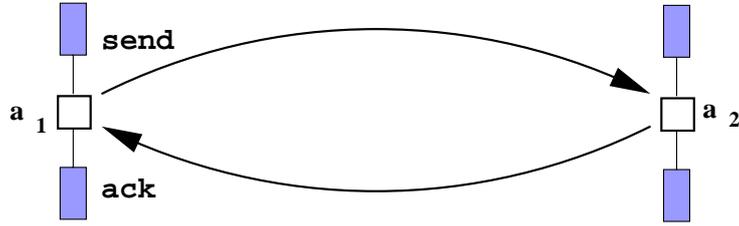
Cette notion de complexité nous conduit à deux réflexions :

1. dans un système multiagent la quantité d'interactions qui est aussi essentielle que le nombre d'agents (bien que ces deux quantités soient généralement corrélées). En effet, l'ajout d'un nouvel agent sans augmenter les interactions est possible mais il ne s'agit pas d'une stratégie qui tire parti des interactions entre les agents. Ainsi, il serait intéressant de travailler sur la capacité d'extension d'un système non pas en termes du nombre d'agents mais en termes de quantité d'interactions ;
2. nous pouvons raffiner cette notion de complexité en évitant les redondances dans les communications. Par exemple, deux agents qui communiquent deux fois de suite sans avoir entre-temps communiqué avec un agent tiers ne font que confirmer leur précédente interaction sans pour autant apporter de nouvelles connaissances. Cette définition nous permet d'envisager par exemple la question suivante : quelle serait l'orbite optimale d'un nouveau satellite à ajouter à la constellation afin d'accroître la vitesse de propagation des informations ?

2 Ruptures de communication

Les satellites des constellations de télécommunication pouvant dériver sur leurs orbites et les coûts de communication étant fonction de la distance entre deux voisins, des algorithmes de routage distribués comme ceux de [Zhao *et al.*, 2005, Bai *et al.*, 2004, Ekici *et al.*, 2000] ont été proposés. L'idée principale est que chaque satellite calcule le coût de communication avec ses voisins puis transmet ces informations au reste de la constellation à l'aide d'un protocole épidémique. Chaque satellite est alors en mesure de contruire une table de routage. En nous inspirant de cette approche, nous nous proposons de définir un

1. Dans cet article, l'auteur prouve que pour deux agents en interaction échangeant leurs connaissances au sujet de l'état du système, le contenu informatif est le même qu'ils utilisent un canal de communication parfait ou une perception directe de l'état de l'autre agent.

FIGURE A.1 – L'agent a_1 envoie un message à l'agent a_2

mécanisme décentralisé pour suivre l'état de la relation Voisins en détectant les ruptures de liaisons intersatellites.

En cas de défaillance d'une liaison intersatellite, le satellite concerné ne peut plus transmettre ou recevoir de connaissances. Il ne peut donc plus faire partie de la relation Voisins (qui doit être par définition symétrique) et ne peut plus être utilisé pour calculer la connaissance commune, la confiance en cette connaissance et les conflits aux sein des coalitions.

De plus, si l'agent en question n'est pas capable de détecter la défaillance par lui-même, il se trouve localement dans l'impossibilité de savoir qui, de lui ou de son voisin, est défaillant. En effet, un agent qui ne reçoit plus peut croire que c'est autrui qui n'émet plus. Il s'agit d'un problème proche de celui des Généraux Byzantins [Lamport *et al.*, 1982], représenté en Figure A.1. Si l'agent a_1 ne peut plus émettre, il ne recevra jamais d'accusé de réception et peut supposer une anomalie chez a_2 (en termes d'émission ou de réception). Si l'agent a_1 ne peut plus recevoir, il peut faire de même.

L'état de chaque agent est encapsulé dans une connaissance (voir Définition 2.17) et est propagé par le protocole de communication (voir Protocole 3.5). Formellement,

Définition A.3 (État de la constellation)

Soit une constellation \mathcal{S} et un vecteur $\langle s_k \rangle$ où s_k est l'état de l'agent $a_k \in \mathcal{A}$. L'état d'un agent a_k est un couple $s_k = (S_k, \tau_k)$ où $S_k \in \{\top, \perp\}$ et $\tau_k \in \mathbb{T}$.

- S_k indique si l'agent a_k fait partie ou non de la relation Voisins;
- τ_k indique la date de la dernière communication directe avec a_k .

Comme un agent ne peut déterminer localement qui, de lui ou de son voisin, subit une rupture de communication, un agent va toujours supposer que autrui est défaillant. Si cette supposition est fautive et que c'est l'agent lui-même qui est défaillant, deux cas de figures se présentent :

1. l'agent ne peut plus émettre et ne pourra pas propager son état ;
2. l'agent ne peut plus recevoir et sera considéré comme défaillant par ses voisins. Ces derniers ne tiendront pas compte de ses connaissances au sujet de l'état de la constellation.

Le protocole proposé est le suivant :

Protocole A.4 (Suivi de l'état d'un agent)

Soit deux agents a_i et a_j engageant une communication directe à la date τ . Le suivi de l'état de a_j pour a_i s'opère ainsi :

- si a_i détecte une anomalie alors $s_j \leftarrow (\perp, \tau)$;
- si a_i ne détecte pas d'anomalie alors :

1. $s_j \leftarrow (\top, \tau)$;
2. a_i fusionne son suivi d'état avec celui de a_j .

Définition A.5 (Fusion du suivi d'état)

Soit $s_k^i = (S_k^i, \tau_k^i)$ l'état de a_k selon a_i . Deux agents a_i et a_j fusionnent leurs suivis d'états comme suit :

$$\forall k (k \neq i \neq j) \text{ si } S_k^i = S_k^j \text{ alors } s_k = \arg \min_{s_k^i, s_k^j} (\tau_k^i, \tau_k^j) \text{ sinon } s_k = \arg \max_{s_k^i, s_k^j} (\tau_k^i, \tau_k^j)$$

Ce processus de fusion consiste à dire que lorsque deux agents se rencontrent et peuvent communiquer sans anomalie, ils savent qu'ils ne sont pas défailants et confirment ainsi que ce sont les autres qui peuvent l'être. De cette manière les états des autres agents peuvent être mis à jour :

1. si les suivis des agents concordent alors l'état décrit par ce suivi était vrai à partir de l'observation la plus ancienne ;
2. si les suivis des agents ne concordent pas, un changement a eu lors de l'observation la plus récente.

À une date donnée, seuls les agents a_k tels que $S_k = \top$ sont considérés comme faisant partie de la relation Voisins.

Exemple A.6

Soit trois agents a_1 , a_2 et a_3 qui communiquent directement dans l'ordre suivant : (a_1, a_2) à la date τ_1 , (a_2, a_3) à la date τ_2 puis (a_1, a_3) à la date τ_3 (telles que $\tau_1 < \tau_2 < \tau_3$). Supposons une rupture de communication chez a_2 . Le tableau ci-dessous illustre le suivi d'état à chaque date pour chaque agent.

	τ_1	τ_2	τ_3
a_1	(s_2, \perp, τ_1)	(s_2, \perp, τ_1)	$(s_2, \perp, \tau_1), (s_3, \top, \tau_3)$
a_2	(s_1, \perp, τ_1)	$(s_1, \perp, \tau_1), (s_3, \perp, \tau_2)$	$(s_1, \perp, \tau_1), (s_3, \perp, \tau_2)$
a_3		(s_2, \perp, τ_2)	$(s_1, \top, \tau_3), (s_2, \perp, \tau_1)$

À la date τ_3 , les agents a_1 et a_3 savent et partagent l'information spécifiant que a_2 est défailant depuis la date τ_1 .

À partir de sa connaissance du suivi de situation, chaque agent peut construire le graphe de voisinage que nous redéfinissons formellement.

Définition A.7 (Graphe de voisinage local)

Soit $\mathcal{S} = \langle \mathcal{A}, \mathbb{T}, \text{Voisins} \rangle$ une constellation, un agent $a_i \in \mathcal{A}$ et une date $\tau \in \mathbb{T}$. Le graphe de voisinage local $\mathcal{V}_{a_i}^\tau$ de l'agent a_i à la date τ est défini par $\mathcal{V}_{a_i}^\tau = (V, \{(a_i, a_j)\}, \{\{v_{ij}\}\})$ où :

- l'ensemble des sommets de \mathcal{V} est $V = \{a_k \in \mathcal{A} : s_k = (\top, \tau_k), \tau_k < \tau\}$;
- l'arête (a_i, a_j) existe si et seulement si $\exists \tau_j \in \mathbb{T}_P, a_j \in \text{Voisins}(a_i, \tau_j)$;
- chaque arête (a_i, a_j) est évaluée par l'ensemble v_{ij} de ces dates τ_j .

Ce graphe peut encore être raffiné en l'inscrivant dans un intervalle temporel. Ainsi, il est intéressant de définir la relation Voisin telle qu'elle était entre deux dates afin de savoir si une connaissance a été propagée avant qu'un agent donné ne soit défaillant.

3 Aléas de mission

Les aléas de mission correspondent à l'ensemble des défaillances qui viennent perturber les objectifs de mission. Il s'agit donc de défaillances touchant aux intentions des agents. Nous pouvons considérer trois cas :

1. un **aléa de génération** est une défaillance qui empêche l'agent de générer de nouvelles intentions. Typiquement, il peut s'agir d'une défaillance du planificateur individuel ou du module de mémoire où sont conservées ces informations. Quoi qu'il en soit, l'agent ne pourra plus faire de nouvelles observations et il va petit à petit jouer un simple rôle de relais de communication. Que l'agent détecte lui-même ou non la défaillance, ce retrait de la constellation se fera automatiquement par le fait qu'il n'aura plus d'interactions avec les autres agents : l'anomalie se résout d'elle-même ;
2. un **aléa de conservation** signifie que l'agent ne peut pas tenir ses propositions ou ses engagements et qu'il se désiste au dernier moment. Ceci peut être dû à une défaillance du planificateur individuel ou du module de gestion de la mémoire. Quoi qu'il en soit, l'agent va perturber la coordination de la constellation en générant des intentions intenables. Si l'agent peut détecter lui-même la défaillance, il lui suffit d'arrêter de générer des intentions (et considérer un aléa de génération). S'il ne peut la détecter lui-même, les autres agents doivent le faire par l'intermédiaire d'un mécanisme de *suivi de situation* ;
3. un **aléa de réalisation** correspond à une défaillance de l'instrument d'observation : l'agent ne peut plus réaliser de tâches. Si l'agent détecte lui-même cette défaillance, il peut arrêter de générer des intentions (et considérer de nouveau un aléa de génération). Dans le cas contraire, il s'agit d'une *anomalie absorbante* : seule une station sol peut se rendre compte de l'anomalie et ensuite prévenir le reste de la constellation.

Dans le Chapitre 3 Section 4, nous avons considéré une notion de confiance relative à une proposition afin de capturer la dynamique du système. Ici, nous nous proposons de définir une notion de confiance relative à un agent afin de détecter une défaillance de type **aléas de mission / conservation**.

Nous nous inspirons du modèle de réputation beta proposé par [Josang et Ismail, 2002]. Il s'agit d'un modèle qui se fonde sur la prédiction du comportement binaire d'un agent² en regard de son comportement passé. La probabilité *a posteriori* d'un événement binaire peut être représentée par des distributions beta. Il s'agit de fonctions de densité de probabilités paramétrées par un couple (α, β) telles que :

$$f(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1} (1-p)^{\beta-1} \quad \text{où } 0 \leq p \leq 1, \alpha \geq 0, \beta \geq 0$$

L'espérance mathématique de la distribution beta est donnée par :

$$E(p) = \frac{\alpha}{(\alpha + \beta)}$$

Si on considère r le nombre d'engagements réalisés et s le nombre d'engagements non tenus, la distribution beta permettant de prédire le comportement futur est donnée par $\alpha = (r + 1)$ et $\beta = (s + 1)$. On peut donc définir une *valeur de réputation*³ et un *rang de réputation*⁴. Respectivement,

$$E(\phi(p|r, s)) = \frac{r + 1}{r + s + 2} \quad \text{et} \quad R(r, s) = \frac{r - s}{r + s + 2}$$

La mise à jour de la réputation d'un agent se fait par simple addition sur r et s . Les informations transmises par les agents doivent alors être prises en compte en fonction de leur réputation. Notons (r', s') la réputation de l'agent qui transmet des informations et (r'', s'') la réputation qu'il transmet alors la réputation (r, s) prise en compte est :

$$r = \frac{2r'r''}{(s' + 2)(r'' + s'' + 2) + 2r'} \quad \text{et} \quad s = \frac{2s's''}{(s' + 2)(r'' + s'' + 2) + 2r'}$$

Il peut être intéressant d'oublier les informations successivement transmises par un même agent. En effet, un agent peut modifier son comportement et des informations par trop anciennes peuvent ne plus avoir cours. Considérons $(r(\tau), s(\tau))$ l'information courante, (r, s) la nouvelle information reçue et λ un facteur d'oubli⁵ alors :

$$r(\tau) = r(\tau - 1)\lambda + r \quad \text{et} \quad s(\tau) = s(\tau - 1)\lambda + s$$

2. Soit un engagement de l'agent a_i , ce dernier le réalisera-t-il ?
3. Probabilité que l'agent respecte son engagement.
4. Mesure de la confiance représentée entre $[-1; +1]$.
5. Si $\lambda = 1$ l'agent n'oublie jamais tandis qu'avec $\lambda = 0$ l'agent ne prend en compte que les dernières informations.

Un système de réputation beta peut être mis en place à partir des intentions échangées entre les agents. Il nous faut définir deux agrégats, l'ensemble des tâches théoriquement réalisées par l'agent et l'ensemble des tâches effectivement réalisées.

Définition A.8 (Ensemble des tâches théoriquement réalisées)

Soit deux agents a_i, a_j et τ la date courante. L'ensemble des tâches théoriquement réalisées par a_j selon a_i est :

$$S_{a_j}^\tau = \{t \in \mathcal{T}_{a_i}^\tau : \exists I_t^{a_j} \in \mathcal{K}_{a_i}^\tau, I_t^{a_j} \in \{\square, \diamond\}, \text{rea}(I_t^{a_j}) < \tau\}$$

Définition A.9 (Ensemble des tâches effectivement réalisées)

Soit deux agents a_i, a_j, τ la date courante et $S_{a_j}^\tau$ l'ensemble de ses tâches théoriquement réalisées par a_j selon a_i . L'ensemble des tâches effectivement réalisées selon a_i est :

$$R_{a_j}^\tau = \{t \in S_{a_j}^\tau : b_t\}$$

Remarquons que les tâches effectivement réalisées ne le sont pas nécessairement par l'agent a_j en raison du mécanisme de formation de coalitions. Nous pouvons alors formellement définir la réputation d'un agent.

Définition A.10 (Réputation)

Soit deux agents a_i, a_j et τ la date courante. La réputation de l'agent a_j à la date τ selon a_i est l'espérance qu'il réalise sa prochaine intention au vu de ses réalisations passées. Elle est définie par :

$$E_{a_j}^\tau = \frac{|R_{a_j}^\tau| + 1}{|S_{a_j}^\tau| + 2}$$

Comme dans le cas de la confiance dans une proposition, il nous faut définir un seuil en-dessous duquel la réputation est trop faible. Ce seuil doit être défini par l'utilisateur de la constellation. Nous pouvons considérer par exemple qu'une réputation de 0.5 ou moins entraîne l'exclusion de la constellation.



Bibliographie

- S. Abdallah et V. Lesser (2004). Organization-based cooperative coalition formation. In *Proceedings of the IEEE International Conference on Intelligent Agent Technology*, pages 162–168.
- A. Abdul-Rahman et S. Hailes (2000). Supporting trust in virtual communities. In *Proceedings of the 33th IEEE International Conference on Systems Sciences*, pages 1–9.
- M. Abubakr et M. Egerstedt (2004). On the structural complexity of multi-agent robot formations. In *Proceedings of the American Control Conference*, volume Vol. 6, pages 4957–4962.
- M. Abubakr et M. Egerstedt (2005a). Connectivity graphs as models of local interactions. *Journal of Applied Mathematics and Computation*, Vol. 168(1) :243–269.
- M. Abubakr et M. Egerstedt (2005b). Decentralized coordination with local interactions : Some new directions. *Springer Lecture Notes in Control and Information Sciences*, Vol. 309 :153–170.
- S. Aknine, S. Pinson et M. Shakun (2004). A multi-agent coalition formation method based on preference models. *International Journal Group Decision and Negotiation*, Vol. 13(6) :513–538.
- S. Aknine et O. Shehory (2005). Formation de coalitions : Concessions, relations entre tâches et réduction de complexité. *Systèmes Multi-agents*, pages 129–142.
- P. Aschwanden, V. Baskaran, S. Bernardini, C. Fry, M. Moreno, N. Muscettola, C. Plaunt, D. Rijsman et P. Tompkins (2006). Model-unified planning and execution for distributed autonomous system control. In *Proceedings of the AAAI 2006 Fall Symposium*, pages 1–10.
- F. Azzedin et M. Maheswaran (2003). Trust modeling for peer-to-peer based computing systems. In *Proceedings of the 17th International Parallel and Distributed Processing Symposium*, pages 99–109.

- C. Baeijs et Y. Demazeau (1996). Les organisations dans les systèmes multi-agents. In *Actes des 4es Journées National du PRC-IA sur les Systèmes Multi-Agents*, pages 35–46.
- J. Bai, X. Lu, Z. Lu et W. Peng (2004). A distributed hierarchical routing protocol for non-GEO satellite networks. In *Proceedings of the International Conference on Parallel Processing Workshops*, pages 148–154.
- P. Balbiani et A. El Fallah-Seghrouchni (1998). A modal logic for reasoning about beliefs, desires and intentions of complex agents in a distributed environment. In *Proceedings of the 3rd International Conference on Multi-Agent Systems*, pages 387–388.
- T. Balch et R. Arkin (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, Vol. 1:1–25.
- A. Barrett (1999). Autonomy architectures for a constellation of spacecraft. In *Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 291–296.
- G. Beaumet, G. Verfaillie et M.-C. Charmeau (2008). Autonomous planning for an agile earth-observing satellite. In *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
- C. Bernon, M.-P. Gleizes, S. Peyruqueou et G. Picard (2003). ADELFE, a methodology for adaptive multi-agent systems engineering. *Engineering Societies in the Agents World*, Vol. 3:70–81.
- G. Bonnet et C. Tessier (2007a). Collaboration among a satellite swarm. In *Proceedings of the 6th International Conference on Autonomous Agents and Multi-Agents Systems*, pages 287–294.
- G. Bonnet et C. Tessier (2007b). Coopération au sein d’une constellation de satellites. In *Actes des 15es Journées Francophones sur les Systèmes Multi-Agents*, pages 171–180.
- G. Bonnet et C. Tessier (2007c). On-board cooperation for satellite swarms. In *Proceedings of the IJCAI’07 Workshop on Artificial Intelligence in Space Application*.
- G. Bonnet et C. Tessier (2008a). Evaluer un système multiagent physique: retour sur expérience. In *Actes des 16es Journées Francophones sur les Systèmes Multi-Agents (prix de la meilleure contribution)*, pages 13–22.
- G. Bonnet et C. Tessier (2008b). Incremental adaptive organization for a satellite constellation. In *Proceedings of the OAMAS workshop at the 7th International Conference on Autonomous Agents and Multi-Agents Systems*, pages 63–78.
- G. Bonnet et C. Tessier (2008c). Incremental adaptive organization for a satellite constellation. *Lecture Notes in Artificial Intelligence: Special Issue on Organized Adaptation in Multi-Agent Systems*, Vol. 5358:108–125.
- G. Bonnet et C. Tessier (2008d). Multi-agent collaboration: A satellite constellation case. In *Proceedings of the 4th European Starting AI Researcher Symposium at the 18th European Conference on Artificial Intelligence*, pages 24–35.

- G. Bonnet et C. Tessier (2008e). A trust model based on communication capabilities for physical agents. In *Proceedings of the TRUST workshop at the 7th International Conference on Autonomous Agents and Multi-Agents Systems*, pages 2–6.
- G. Bonnet, C. Tessier, M.-C. Charmeau et P. Dago (2006). Planification pour un essaim de satellites: rapport préliminaire. In *Actes des 1res Journées Francophones Planification, Décision et Apprentissage pour la conduite de systèmes*, pages 47–48.
- E. Bornschlegl, C. Guettier, G. L. Lann et J.-C. Poncet (2001). Constraint-based layered planning and distributed control for an autonomous spacecraft formation flying. In *Proceedings of the 1st ESA Workshop on Space Autonomy*.
- E. Bornschlegl, C. Guettier et J.-C. Poncet (2000). Automatic planning for autonomous spacecraft constellation. In *Proceedings of the 2nd NASA International Workshop on Planning and Scheduling for Space*.
- S. Botelho (2000). *Une architecture décisionnelle pour la coopération multi-robots*. Thèse de doctorat, Institut National Polytechnique de Toulouse, Toulouse.
- S. Bouveret (2007). *Allocation et partage équitables de ressources indivisibles: modélisation, complexité et algorithmique*. Thèse de doctorat, Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse.
- A. Brambilla, M. Lavagna, A. DaCosta et A. Finzi (2005). Distributed planning and scheduling for space system flotillas. In *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
- C. Brooks et E. Durfee (2002). Congregating and market formation. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 96–103.
- C. Brooks et E. Durfee (2003). Congregation formation in multiagent systems. *Journal of Autonomous Agents and Multiagent Systems*, Vol. 7:145–170.
- C. Brooks, E. Durfee et A. Armstrong (2000). An introduction to congregating in multiagent systems. In *Proceedings of the 4th International Conference on Multi-Agent Systems*, pages 78–86.
- R. Brooks (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, Vol. 2(1):14–23.
- S. Buckreuss, W. Balzer, P. Muhlbauer, R. Werninghaus et W. Pitz (2003). The TerraSAR-X satellite project. In *Proceedings of the Geoscience and Remote Sensing Symposium*, volume Vol. 5.
- P. Buzing, A. Eiben et M. Schut (2005). Emerging communication and cooperation in evolving agent societies. *Journal of Artificial Societies and Social Simulation*, Vol. 8(1):27–52.
- P. Buzing, A. Eiben, M. Schut et T. Toma (2004). Cooperation and communication in evolving artificial societies. In *Proceedings of the 2004 Congress on Evolutionary Computation*, volume Vol. 2.

- P. Caillou, S. Aknine et S. Pinson (2002). A multi-agent method for forming and dynamic restructuring of Pareto optimal coalitions. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1074–1081.
- M. Campbell et T. Schetter (2002). Comparison of multiple agent-based organizations for satellite constellations. *Journal of Spacecraft and Rockets*, Vol. 39(2) :274–283.
- J. Carbo, J. Molina et J. Davila (2002). Comparing predictions of SPORAS vs. a fuzzy reputation agent system. In *Proceedings of the 3rd International Joint Conference on Fuzzy Sets and Fuzzy Systems*, pages 147–153.
- A. Carrel et P. Palmer (2005). An evolutionary algorithm for near-optimal autonomous resource management. In *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
- A. Carrel et P. Palmer (2006). Adaptive resource modelling for autonomous planning and scheduling. In *Proceedings of the 5th International Workshop on Planning and Scheduling for Space*.
- J. Carter, E. Bitting et A. Ghorbani (2002). Reputation formalization for an information-sharing multi-agent system. *Computational Intelligence*, Vol. 18(2) :515–534.
- C. Castelfranchi et R. Falcone (1998). Principles of trust for MAS: cognitive anatomy, social importance, and quantification. In *Proceedings of the International Conference on Multi-Agent Systems*, pages 72–79.
- M.-C. Charmeau et E. Bensana (2005). AGATA, a lab bench project for spacecraft autonomy. In *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
- A.-H. Chow (1996). Parallel DEVS: A parallel hierarchical modular modeling formalism and its distributed simulator. *Transactions of the International Society for Computer Simulation*, Vol. 13(2) :55–68.
- P. Cohen et H. Levesque (1990). Intention is choice with commitment. *Artificial Intelligence*, Vol. 42(3) :213–261.
- J.-M. Contet, F. Gechter, P. Gruer et A. Koukam (2008). Evaluation of global system state thanks to local phenomena. In *Proceedings of the 18th European Conference on Artificial Intelligence*, pages 865–866.
- D. Corkill et S. Lander (1998). Diversity in agent organization. *Object Magazine*, Vol. 8(4) :41–47.
- S. Curtis, M. Rilee, P. Clark et G. Marr (2003). Use of swarm intelligence in spacecraft constellations for the resource exploration of the asteroid belt. In *Proceedings of the 3rd International Workshop on Satellite Constellations and Formation Flying*, pages 24–26.
- S. Damiani (2005). *Gestion d'une constellation de satellites de surveillance de la Terre : autonomie et coordination*. Thèse de doctorat, École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse.

- S. Damiani, G. Verfaillie et M.-C. Charneau (2005). An Earth watching satellite constellation: How to manage a team of watching agents with limited communications. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 455–462.
- A. D. Dang, R. Dash et N. Jennings (2006). Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, pages 635–640.
- V. D. Dang et N. Jennings (2004). Generating coalition structures with finite bound from the optimal guarantees. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 564–571.
- G. Dantzig (1963). *Linear programming and Extensions*. Princeton University Press.
- G. Dantzig et P. Wolfe (1961). The decomposition algorithm for linear programs. *Econometric*, Vol. 29(4):767–778.
- S. Das, P. Gonzales, R. Krikorian et W. Truszkowski (1999). Multi-agent planning and scheduling environment for enhanced spacecraft autonomy. In *Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
- S. Das, D. Knights, C. Wu et W. Truszkowski (1995). Distributed intelligent planning and scheduling (DIPS). In *Proceedings of the 1st International Conference on Multi-Agent Systems*.
- P. Davidsson et S. Johansson (2003). Evaluating multi-agent system architectures: a case study concerning dynamic resource allocation. *Engineering Societies in the Agents World*, Vol. 3:25–77.
- R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith et R. Wahington (2002). Contingency planning for planetary rovers. In *Proceedings of the 3rd NASA International Workshop on Planning and Scheduling for Space*.
- R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith et R. Wahington (2003). Incremental contingency planning. In *Proceedings of ICAPS'03 Workshop on Planning under Uncertainty and Incomplete Information*, pages 1–10.
- F. Dietrich, P. Mentzen et P. Monte (1998). The Globalstar cellular satellite system. *IEEE Transaction on Antennas and Propagation*, Vol. 46(6):935–942.
- A. Drogoul (1995). When ants play chess (or can strategies emerge form tactical behaviors?). *Artificial Intelligence*, Vol. 957:13–27.
- J. Dungan, J. Frank, A. Jonsson, R. Morris et D. Smith (2002). Advances in planning and scheduling of remote sensing instruments for fleets of earth orbiting satellites. In *Proceedings of the Earth Science Technology Conference*.
- E. Durfee (2001). Scaling up agent coordination strategies. *IEEE Computer*, Vol. 34(7):39–46.

- P. Dutta et S. Sen (2002). Emergence of stable coalitions via task exchanges. In *Proceeding of the 1st International Joint Conference of Autonomous Agents and Multi-Agent Systems*, pages 312–313.
- A. Eiben, M. Schut et T. Toma (2005). Comparing multicast and newcast communication in evolving agent societies. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 75–82.
- E. Ekici, I. Akyildiz et M. Bender (2000). Datagram routing algorithm for LEO satellite network. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 500–508.
- J. Epstein et R. Axtell (1996). *Growing Artificial Societies: Social Science From The Bottom Up*. Brookings Institute Press.
- D. Escorial, I. F. Tourne et F. J. Reina (2003). Fuego : a dedicated constellation of small satellites to detect and monitor forest fires. *Acta Astronautica*, Vol.52(9-12) :765–775.
- B. Esfandiari et S. Chandrasekharan (2001). On how agents make friends : Mechanisms for trust acquisition. In *Proceedings of the 4th Workshop on Deception, Fraud, and Trust in Agent Societies*, pages 27–34.
- D. Evans, C. Elachi, B. Holt, M. Kobrick, H. Ottl, P. Pampaloni, M. Schier, E. Stofan, M. Vogt, S. Wall, J. van Zyl et J. Way (1993). The shuttle imaging radar-C and X-SAR mission. *EOS Transactions American Geophysical Union*, Vol. 74(13) :145–145.
- J. Ferber, O. Gutknecht et F. Michel (2004). From agents to organizations : an organizational view of multi-agent systems. *Agent-Oriented Software Engineering IV*, pages 214–230.
- H. Fiorino (1998). *Elaboration de conjectures par des agents coopérants*. Thèse de doctorat, École Nationale Supérieure de l’Aéronautique et de l’Espace, Toulouse.
- K. Fischer, M. Schillo et J. Siekmann (2003). Holonic multiagent systems : A foundation for the organisation of multiagent systems. In *Proceedings of the 1st International Conference on Applications of Holonic and Multi-Agent Systems*, pages 71–80.
- S. D. Florio, T. Zehetbauer et T. Neff (2005). Optimal operations planning for sar satellite constellations in low earth orbit. In *Proceedings of the 6th International Symposium on Reducing the Costs of Spacecraft Ground Systems and Operations*.
- S. D. Florio, T. Zehetbauer et T. Neff (2006). SCOOP : Satellite constellations optimal operations planner. In *Proceedings of the ICAPS’06 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems*.
- S. Franklin et A. Graesser (1996). Is it an agent or just a program? a taxonomy for autonomous agents. *Lecture Notes In Computer Science*, Vol. 1193 :21–35.
- V. Gabrel (1994). *Méthodologie pour la planification de production de systèmes d’observation de la Terre par satellite*. Thèse de doctorat, Université Paris - Dauphine, Paris.

- G. Gans, M. Jarke, S. Kethers, G. Lakemeyer, L. Ellrich, C. Funken et M. Meister (2001). Towards (dis)trust-based simulations of agent networks. In *Proceedings of the 4th Workshop on Deception, Fraud, and Trust in Agent Societies*, pages 49–60.
- J.-P. Georgé, M.-P. Gleizes et P. Glize (2003). Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie des Amas. *Revue Française d'Intelligence Artificielle*, Vol. 17(4) :591–626.
- P. Giacomo (1996). Sans dimension ou de dimension un ? *Metrologia*, Vol. 32 :311–313.
- M.-P. Gleizes, V. Camps, J.-P. Georgé et D. Capera (2008). Engineering systems which generate emergent functionalities. *Lecture Notes in Artificial Intelligence: Special Issue on Engineering Environment-Mediated Multiagent Systems*, Vol. 5049.
- A. Globus, J. Crawford, J. Lohn et A. Pryor (2003). Scheduling earth observing satellites with evolutionary algorithms. In *Proceedings of the 2nd International Conference on Space Mission Challenges for Information Technology*.
- C. Goldman et J. Rosenschein (2002). Evolutionary patterns of agent organizations. *IEEE Transactions on Systems, Man and Cybernetics Part A*, Vol. 30(1) :135–148.
- H. Goradia et J. Vidal (2007). An equal excess negotiation algorithm for coalition formation. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1052–1054.
- P.-P. Grassé (1959). La reconstruction du nid et les coordinations inter-individuelles chez *bellicositermes natalensis* et *cubitermes* sp. la théorie de la stigmergie : essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, Vol. 6 :41–81.
- N. Griffiths (2008). Tags and image scoring for robust cooperation. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 575–582.
- N. Griffiths et M. Luck (2003). Coalition formation through motivation and trust. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 17–24.
- C. Guettier et J.-C. Poncet (2001). Multi-level planning for spacecraft autonomy. In *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 18–21.
- I. Gupta, A.-M. Kermarrec et A. Ganesh (2002). Efficient epidemic-style protocols for reliable and scalable multicast. In *Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems*, pages 180–189.
- J. Holland (1993). The effect of labels (tags) on social interactions. *Santa Fe Institute Working Paper 93-10-064*.
- J. Holliday, D. Agrawal et A. El Abbadi (2000). Database replication using epidemic communication. In *Proceedings of the 6th Euro-Par Conference*, pages 427–434.

- B. Horling et V. Lesser (2004). A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, Vol. 19 :281–316.
- J. Hubner, J. Sichman et O. Boissier (2002a). Moise+ : Towards a structural, functional, and deontic model for the mas organization. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*.
- J. Hubner, J. Sichman et O. Boissier (2002b). Spécification structurelle, fonctionnelle et déontique d’organisations dans les systèmes multi-agents. In *Actes des 10es Journées Francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, pages 205–216.
- K. Jenkins, K. Hopkinson et K. Birman (2001). A gossip protocol for subgroup multicast. In *Proceedings of the 21st International Conference on Distributed Computing Systems Workshops*, pages 25–30.
- N. Jennings, S. Parsons, P. Norriega et C. Sierra (1998). On argumentation-based negotiation. In *Proceedings of the International Workshop on Multi-Agent Systems*, pages 1–7.
- M. Johnson, S. Tompkins et W. Truszkowski (1999). Information systems for nanosatellite constellations. *American Institute of Aeronautics and Astronautics*.
- A. Josang et R. Ismail (2002). The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*.
- N. Karmarkar (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, Vol. 4(4) :373–395.
- L. Khachiyan (1979). A polynomial algorithm in linear programming. *Soviet Math. Dokl.*, Vol. 20(1) :191–194.
- S. Knight, G. Rabideau, S. Chien, B. Engelhardt et R. Sherwood (2001). Casper : space exploration through continuous planning. *IEEE Intelligent Systems*, Vol. 16(5) :70–75.
- M. Krainin, B. An et V. Lesser (2007). An application of automated negotiation to distributed task allocation. In *Proceedings of the International Conference on Intelligent Agent Technology*, pages 138–145.
- S. Kraus, O. Shehory et G. Taase (2003). Coalition formation with uncertain heterogeneous information. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1–8.
- S. Kraus, O. Shehory et G. Taase (2004). The advantages of compromising in coalition formation with incomplete information. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 588–595.
- D. Kreps (1990). *Game Theory and Economic Modelling*. Oxford Press University.
- Y. Kubera, P. Mathieu et S. Picault (2007). La complexité dans la simulation multi-agent. In *Actes des 15es Journées Francophones sur les Systèmes Multi-Agents*, pages 139–148.

- B. Laborde, C. Castel, J.-F. Gabard, R. Soumagne et C. Tessier (2006). FDIR strategies for autonomous satellite formations - a preliminary report. In *Proceedings of the AAAI 2006 Fall Symposium on Spacecraft Autonomy: Using AI to Expand Human Space Exploration*.
- L. Lamport, R. Shostak et M. Pease (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, Vol. 4(3) :382–401.
- H. Lau et L. Zhang (2003). Task allocation via multi-agent coalition formation : Taxonomy, algorithms and complexity. In *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 346–350.
- L. Lee, H. Nwana, D. Ndumu et P. D. Wilde (1998). The stability, scalability and performance of multi-agent systems. *BT Technology Journal*, Vol. 16(3) :94–103.
- F. Legras (2003). *Organisation dynamique d'équipes d'engins autonomes par écoute flottante*. Thèse de doctorat, École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse.
- C. Lemaître et C. Excelente (1998). Multi-agent organization approach. In *Proceedings of the 2nd Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent Systems*.
- M. Lemaître et G. Verfaillie (2007). Interaction between reactive and deliberative tasks for on-line decision-making. In *Proceedings of ICAPS'07 Workshop on Planning and Plan Execution for Real-World Systems*.
- K. Maine, C. Devieux et P. Swan (1995). Overview of IRIDIUM satellite network. In *Proceedings of the Western Conference on Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies*, pages 483–490.
- S. Marsh (1994). *Formalising Trust as a Computational Concept*. PhD. thesis, University of Stirling, Stirling.
- R. Mayer, J. Davis et F. Schoorman (1995). An integrative model of organizational trust. *Academy of Management Review*, Vol. 20(3) :709–734.
- N. Meuleau et D. Smith (2003). Optimal limited contingency planning. In *Proceedings of the 19th AAAI Conference on Artificial Intelligence*, pages 417–426.
- P. Modi et M. Veloso (2005). Bumping strategies for the multiagent agreement problem. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 390–396.
- H. Moulin (2003). *Fair Division and Collective Welfare*. The MIT Press.
- J. B. Mueller, D. M. Surka et B. Udrea (2001). Agent-based control of multiple satellite formation flying. In *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.

- G. Muller (2006). *Utilisation de normes et de réputations pour détecter et sanctionner les contradictions : contribution au contrôle social des interactions dans les systèmes multi-agents ouverts et décentralisés*. Thèse de doctorat, École Nationale Supérieure des Mines de Saint-Etienne, Saint-Etienne.
- R. B. Murphy (2000). Marsupial and shape-shifting robots for urban search and rescue. *IEEE Intelligent Systems*, Vol. 15(2) :14–19.
- N. Muscettola, P. Nayak, B. Pell et B. Williams (1998). RemoteAgent : To boldly go where no AI system has gone before. *Artificial Intelligence*, Vol. 103(1-2) :5–48.
- H. S. Nwana, L. C. Lee et N. R. Jennings (1996). Coordination in software agent systems. *The British Telecom Technical Journal*, Vol. 14(4) :79–88.
- C. Ortiz et T. Rauenbusch (2002). Dynamic negotiation. In *Proceedings of the AAAI'02 Workshop on Planning with and for Multiagent Systems*.
- M. Ott, C. Huth et L. Nastansky (1999). Reengineering organizational structures from within. In *Proceedings of the 32nd Hawaii Conference on System Sciences*, pages 200–208.
- R. Parasuraman, T. Sheridan et C. Wickens (2000). A model for types and levels of human interaction with automation. *IEEE Transaction on Systems, Man and Cybernetics*, Vol. 30(3) :286–298.
- M. Pechoucek, V. Marik et J. Barta (2002). A knowledge-based approach to coalition formation. *IEEE Intelligent Systems*, Vol. 17(3) :17–25.
- B. Penne, C. Tobehn, M. Kassebom et H. Lubberstedt (2005). The Earth observation program at OHB-system. *Acta Astronomica*, Vol. 56(1-2) :17–23.
- B. Pittel (1987). On spreading a rumor. *SIAM Journal of Applied Mathematics*, Vol. 47 :213–223.
- B. Polle (2002). Autonomy requirement and technologies for future constellation. *Astrium Summary Report*.
- A. Procaccia et J. Rosenschein (2006). The communication complexity of coalition formation among autonomous agents. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 505–512.
- V. Rahwan, S. Ramchurn et N. Jennings (2007). Near-optimal anytime coalition structure generation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 2365–2371.
- N. Rey (2007). SIMBOL-X : Monitoring and control specification. *CNES Technical Report*, pages 1–59.
- M. Rodriguez-Moreno, G. Brat, N. Muscettola et D. Rijsman (2007). Validation of a multi-agent architecture for planning and execution. In *Proceedings of the 18th International Workshop on Principles of Diagnosis*, pages 368–371.

- J. Rosenschein et G. Zlotkin (1994). Designing conventions for automated negotiation. *AI Magazine*, Vol. 15(3) :29–46.
- M. Roth, D. Vail et M. Veloso (2003). A world model for multi-robot teams with communication. In *Proceedings of the International Conference on Robotics and Automation*.
- J. Sabater et C. Sierra (2001). REGRET : A reputation model for gregarious societies. In *Proceedings of the 4th Workshop on Deception, Fraud, and Trust in Agent Societies*, pages 61–69.
- J. Sabater et C. Sierra (2005). Review on computational trust and reputation models. *Artificial Intelligence*, Vol. 24(1) :33–60.
- T. Sandholm (1998). Contract types for satisficing task allocation. In *Proceedings of the AAAI Spring Symposium on Satisficing Models*, pages 23–25.
- T. Sandholm, K. Larson, M. Andersson, O. Shehory et F. Tohmé (1999). Coalition structure generation with worst case guarantees. *Artificial Intelligence*, Vol. 111(1-2) :209–238.
- T. Schetter, M. Campbell et D. Surka (2003). Multiple agent-based autonomy for satellite constellation. *Artificial Intelligence*, Vol. 145 :147–180.
- M. Schillo, B. Fley, M. Florian, F. Hillebrandt et D. Hinck (2002). Self-organization in multiagent systems : from agent interaction to agent organization. In *Proceedings of the 3rd International Workshop on Modelling Artificial Societies and Hybrid Organizations*, pages 37–46.
- M. Schillo, P. Funk et M. Rovatsos (2000). Using trust of detecting deceitful agents in artificial societies. *Applied Artificial Intelligence Special Issue on Deception, Fraud and Trust in Agent Societies*, pages 825–848.
- T. Scully, M. Madden et G. Lyons (2004). Coalition calculation in a dynamic agent environment. In *Proceedings of the 21st International Conference on Machine Learning*, pages 561–568.
- J. Searle (1969). *Speech Acts*. Cambridge University Press.
- I. Sen et S. Airiau (2006). Expertise and trust-based formation of effective coalitions : an evaluation on the ART testbed. In *Proceedings of the AAMAS'06 ALAMAS Workshop*.
- S. Sen et P. Dutta (2000). Searching for the optimal coalition structure. In *Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems*, pages 286–292.
- S. Sen et N. Sajja (2002). Robustness of reputation-based trust : Boolean case. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 288–293.
- G. Serugendo, M.-P. Gleizes et A. Karageorgos (2006). Self-organisation and emergence in MAS : An overview. *Informatica*, Vol. 30 :45–54.

- L. S. Shapley (1953). A value for n-person games. *Annals of Mathematical Studies*, Vol. 28 :307–317.
- O. Shehory et S. Kraus (1995a). Coalition formation among autonomous agents: Strategies and complexity. *Lecture Notes in Artificial Intelligence*, Vol. 957 :57–72.
- O. Shehory et S. Kraus (1995b). Task allocation via coalition formation among autonomous agents. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 655–661.
- O. Shehory et S. Kraus (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, Vol. 101(1-2) :165–200.
- O. Shehory et S. Kraus (1999). Feasible formation of coalitions among autonomous agents in non-super-additive environments. *Computational Intelligence*, Vol. 15(3) :218–251.
- R. Sherwood, S. Chien, M. Burl, R. Knight, G. Rabideau, B. Engelhardt, A. Davies, P. Zetocha, R. Wainright, P. Klupar, D. Surka, B. Williams, R. Greeley, V. Baker et J. Doan (2001). The Techsat-21 autonomous sciencecraft constellation demonstration. In *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
- J. Sichman, V. Dignum et C. Castelfranchi (2005). Agent organizations: a concise overview. *Special Issue in Agent Organizations in the Journal of the Brazilian Computer Society*, Vol. 11(1).
- M. Sims, C. Goldman et V. Lesser (2003). Self-organization through bottom-up coalition formation. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 867–874.
- E. Sklar, M. Schut, K. Diwold et S. Parsons (2006). Exploring coordination properties within populations of distributed agents. In *Proceedings of the AAAI Spring Symposium on Distributed Plan and Schedule Management*, pages 121–127.
- R. Smith (1981). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, Vol. 29(12) :1104–1113.
- L.-K. Soh et C. Tsatsoulis (2001). Reflective negotiating agents for real-time multisensor target tracking. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 1121–1127.
- L.-K. Soh et C. Tsatsoulis (2002). Allocation algorithms in dynamic negotiation-based coalition formation. In *Proceedings of the AAMAS'02 Workshop on Teamwork and Coalition Formation*.
- L.-K. Soh et C. Tsatsoulis (2003). Utility-based multiagent coalition formation with incomplete information and time constraints. *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 2 :1481–1486.

- H. Sorre et P. Sourisse (1997). Skybridge: Un système d'accès à large bande utilisant une constellation de satellites en orbite basse. *Revue des télécommunications*, Vol. 2:91–96.
- A. Sturm et O. Shehory (2003). A framework for evaluating agent-oriented methodologies. In *Proceedings of the Workshop on Agent-Oriented Information System*, pages 60–66.
- M. Sturza (1994). The Teledesic satellite system. In *Proceedings of the 4th National Conference on Telesystems*, pages 123–126.
- D. M. Surka (2001). ObjectAgent for robust autonomous control. In *Proceedings of the AAAI Spring Symposium*.
- D. Thang-Tung (2004). Improving plan quality through agent coalitions. In *Proceedings of the IEEE International Conference on Computational Cybernetics*, pages 171–176.
- D. Thanh-Tung, B. Frankovic, C. Sheahan et I. Bundiska (2005). Using agent coalitions for improving plan quality. *Intelligent Systems at the Service of Mankind*, Vol. 2:351–364.
- F. Tohmé et T. Sandholm (1999). Coalition formation processes with belief revision among bounded-rational self-interested agents. *Journal of Logic and Computation*, Vol. 9(6):793–815.
- W. Truszkowski, M. Hinchey, J. Rash et C. Rouff (2006). Autonomous and autonomic systems: A paradigm for future space exploration mission. *IEEE Transactions on Systems, Man and Cynernetics*, Vol. 36(3):279–291.
- W. Truszkowski et J. Karlin (2000). A cybernetic approach to the modeling of agent communities. In *Proceedings of the 4th International Workshop on Cooperative Information Agents and The Future of Information Agents in Cyberspace*, pages 166–178.
- W. Truszkowski, J. Rash, C. Rouff et M. Hinchey (2004). Some autonomic properties of two legacy multi-agent systems - LOGOS and ACT. In *Proceedings of the 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*, pages 1–9.
- P. Turner et N. Jennings (2000). Improving the scalability of multi-agent systems. In *Proceedings of the Workshop on Infrastructure for Scalable Multi-Agent Systems*.
- J. van Zyl (2001). The shuttle radar topography mission (SRTM): a breakthrough in remote sensing of topography. *Acta Astronautica*, Vol.48(5-12):559–565.
- G. Vauvert et A. El Fallah-Seghrouchni (2000). Coalition formation among egoistic agents. In *Proceedings of the International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce*.
- G. Vauvert et A. El Fallah-Seghrouchni (2001). A distributed algorithm for coalition formation among e-commerce agents. In *Proceedings of the 2nd Asia-Pacific Conference on Intelligent Agent Technology*.
- J. Walker (1984). Satellite constellations. *Journal of British Interplanetary Society*, Vol. 37:559–571.

- C.-J. Wang (1993). Structural properties of a low earth orbit satellite constellation – the walker delta network. In *Proceedings of the 12th Annual Military Communications Conference*, pages 968–972.
- B. Werger (1999). Cooperation without deliberation : A minimal behavior-based approach to multi-robot teams. *Artificial Intelligence*, Vol. 110 :293–320.
- M. Werner, A. Böttcher, A. Jahn et E. Lutz (1993). Analysis of connectivity requirements for communication networks based on low earth orbit satellites. In *Proceedings of the 2nd International Conference on Universal Personal Communications*, pages 299–303.
- T. Wieder (2008). The number of certain k -combinations of an n -set. *Applied Mathematics E-notes*, Vol. 8 :45–52.
- T. D. Wolf et T. Holvoet (2004). Emergence and self-organisation : a statement of similarities and differences. In *Proceedings of the International Workshop on Engineering Self-Organising Application*, pages 96–110.
- L. Wood (2003). *Internetworking and computing over satellite networks (p.13-34)*. Kluwer Academic Press.
- B. Yu et M. Singh (2002). Distributed reputation management for electronic commerce. *Computational Intelligence*, Vol. 18(4) :535–549.
- G. Zacharia et P. Maes (2000). Trust management through reputation mechanisms. *Applied Artificial Intelligence*, Vol. 14 :881–907.
- F. Zambonelli, N. Jennings et M. Wooldridge (2003). Developing multiagent systems : The GAIA methodology. *ACM Transactions on Software Engineering and Methodology*, Vol. 12(3) :317–370.
- P. Zetocha (2000). Satellite cluster command and control. *IEEE Aerospace Conference*, Vol. 7 :49–54.
- Z. Zhao, C. Gong et G. Wang (2005). Identification of faulty links in LEO/MEO satellite communication networks. In *Proceedings of the 10th International Conference on Engineering of Complex Computer Systems*, pages 394–400.