



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ «ABBÈS LAGHROUR» DE KHENCHELA
FACULTÉ DES SCIENCES ET DE LA TECHNOLOGIE



Département de Mathématique et Informatique

Mémoire

Présenté pour l'obtention du diplôme de Master en informatique

(L.M.D)

Par :

MERABET Nadji

Option : Sécurité et Technologies Web (STW)

Proposition d'un mécanisme de coordination de tâches pour les SMAs basé sur la théorie des jeux

Soutenu le : 08/09/2019 devant le jury :

- *Dr. SIAM Abd Errahim (président)*
- *Dr. HIOUAL Wassila (examineur)*
- *Dr. SOUIDI Mohammed El Habib (encadreur)*

Année : 2018/2019

Résumé

La théorie des jeux est une approche avancée pour acquérir des formations de coalition dans des systèmes multi-agents. Ce mémoire se concentre sur l'importance du calcul distribué, ainsi que sur la formation et la réformation dynamiques des groupes de poursuites dans les problèmes de poursuite et d'évasion. Afin de répondre à cette tâche, nous proposons un algorithme de formation de coalition décentralisée sur l'élimination itérative des stratégies dominées avec changement des priorités d'élimination (IEDSEPC). Ce processus de la théorie des jeux est commun pour résoudre des problèmes nécessitant le retrait de stratégies dominées de manière itérative.

Afin de refléter l'impact de ce mécanisme de coordination de tâche, nous l'avons appliqué au problème de Poursuite-Evasion dans lequel un ensemble d'agents endossant le rôle Poursuivant doivent former différentes coalitions pour stopper le mouvement des agents évadés détectés dans l'environnement.

De plus, nous avons utilisé la coordination du mouvement basée sur les principes du processus de Markov (MDP) pour contrôler la stratégie de déplacement des poursuivants dans l'environnement. Les résultats de la simulation démontrent la faisabilité et la validité de l'approche donnée par rapport à d'autres méthodes décentralisées.

Durant les simulations effectuées, nous avons étudié trois paramètres différents en comparaison avec une récente approche (IEDS) traitant le même problème. Ces trois paramètres sont le temps de capture des évadés, le développement de l'acquisition des récompenses, ainsi que le dynamisme des coalitions formées durant la poursuite.

Mots clés : systèmes multi-agents, La théorie des jeux, Poursuit-Evasion, Coordination des tâches, Coordination du mouvement.

Abstract

Game theory is an advanced approach allowing the coalition formations in multi-agents systems. This thesis focuses on the importance of distributed computing, as well as the dynamic formation of the pursuers' groups in pursuit-evasion problem. In order to answer to this task, we propose a decentralized coalition formation algorithm based on the iterative elimination of dominated strategies with change of elimination priorities (IEDSEPC). This process of game theory is common to solve problems requiring the iterative suppression of dominated strategies.

In order to reflect the impact of this task coordination mechanism, we applied it to the Pursuit-Evasion problem in which a set of agents equipped with the role pursuer must form different coalitions to stop the movement of evaders' agents detected in the environment.

In addition, we used A motion coordination method based on the Principles of the Markov Decision Process (MDP) to control the movement strategy of pursuers in the environment. The results of the simulation demonstrate the feasibility and validity of the given approach compared to other decentralized methods.

During the simulation part, we studied three different parameters in comparison with a recent approach (IEDS) dealing with the same problem. These three parameters are, the capturing time of the evaders, the development of the acquisition of the rewards, as well as the dynamism of the formed coalitions during the pursuit.

Key words: Multi-Agents Systems, Game Theory, Pursuit-Evasion, Task Coordination, Path planning.

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mes études et qui m'ont aidée lors de la rédaction de ce mémoire.

Je voudrais dans un premier temps remercier, mon directeur de mémoire Dr : Mohammed El Habib Souidi, pour sa patience, sa disponibilité et surtout ses judicieux conseils, Je le remercie de m'avoir encadré, orienté, aidé et conseillé.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de répondre à mes questions durant mes recherches.

Table des matières

Résumé	i
Abstract	ii
Remerciements	iii
Table des matières	iv
Table des figures.....	v
Liste des tableaux.....	ix
Introduction générale.....	1
I Introduction.....	2
II Problématique	3
III Objectifs.....	3
IV Méthodologie de recherche	3
V Organisation du mémoire	4
Chapitre 1 Agents et systèmes multi-agents	5
1.1 Introduction.....	6
1.2 Pourquoi s'intéresser aux systèmes multi-agents ?.....	7
1.3 Agent	8
1.3.1 Définitions	8
1.3.2 Typologie des agents	10
1.3.3 Propriétés clés des Agents.....	13
1.3.4 Propriétés additionnelles	14
1.4 Systèmes multi-agents	14
1.4.1 Définitions	14
1.4.2 Environnement	16
1.4.3 Propriétés des environnements	16
1.5 Propriétés attendues des SMA.....	17
1.5.1 Rationalité	17
1.1.1 Autonomie.....	18
1.6 Typologie des systèmes multi-agents	18
1.6.1 Les systèmes multi-agents cognitifs	18
1.1.2 Les systèmes multi-agents réactifs	19

1.7	Espace SMA	19
1.8	Domaines d'application des SMA	21
1.9	Communication dans les systèmes multi-agents	22
1.9.1	Définition	22
1.9.2	Types de communication	22
1.10	Interactions dans les SMA	23
1.10.1	Définition	23
1.10.2	Coopération	25
1.11	Organisations dans les SMA	26
1.11.1	Définition	26
1.11.2	Types d'organisation	27
1.12	Conclusion	28
Chapitre 2	Les mécanismes de coordination de tâches dans les SMAs	29
2.1	Introduction	30
2.2	L'importance de la coordination	30
2.3	Les techniques de coordination	32
2.3.1	Modèle organisationnel	32
2.3.2	Coordination par protocoles	33
2.3.3	Planification multi-agents	34
2.3.4	Négociation	36
2.3.5	Formation de la coalition	38
2.4	Critères pour choisir une technique de coordination	40
2.5	Organisations dans les systèmes multi-agents	42
2.6	Organisation et modèle organisationnel	43
2.7	Modèles organisationnels	43
2.7.1	AALAADIN (AGR) [GUT, 2001]	44
2.7.2	Le modèle AGRMF (Agent-Group-Role-Membership Function)	46
2.7.3	Le modèle YAMAM (YET ANOTHER MULTI-AGENT MODEL)	47
2.8	Théorie des jeux dans les systèmes multi-agents	49
2.9	La coordination avec l'application de la théorie des jeux	53
2.10	Conclusion	55
Chapitre 3	Une nouvelle approche de la poursuite en coopération multi-agents	56
3.1	Introduction	57
3.2	Description du problème de Poursuite-Evasion	57
3.2.1	Le poursuivant	58
3.2.2	L'évadé	58

3.3	Élimination itérative des stratégies dominées (IEDS).....	59
3.4	Notre contribution.....	60
3.4.1	Algorithme de formation de coalition basé sur IEDSEPC	60
3.4.2	Processus Décisionnel de Markov	65
3.5	Conclusion	68
Chapitre 4	Expériences de simulation	69
4.1	Introduction.....	70
4.2	La programmation orientée agents.....	71
4.2.1	La Plateforme JADE.....	72
4.2.2	La Plateforme MADKIT	73
4.2.3	La plateforme NetLogo	74
4.3	Expériences de simulation.....	78
4.3.1	Utilisation du model IEDS	81
4.3.2	Utilisation du model IEDSEPC avec 03 changement de priorités	81
4.4	Conclusion	82
	Conclusion et perspectives.....	83
I	Résumé conclusif.....	84
II	Perspectives et tendances futures	84
	Bibliographie.....	85

Liste des figures

Figure 1-1 : Agent réactif et environnement.....	8
Figure 1-2 : Structure classique d'un agent.....	9
Figure 1-3 : Environnement d'un système multi agents	14
Figure 1-4 : Positionnement d'un agent dans le monde	15
Figure 1-5 : Espace SMA [7].....	20
Figure 1-6 : Communication indirecte.....	22
Figure 1-7 : Communication directe.....	22
Figure 2-1 : Approches des mécanismes de coordination	32
Figure 2-2 : Agent – Group - Rôle méta modèle [66]	45
Figure 2-3 : YAMAM méta modèle	49
Figure 3-1 : Organigramme de l'algorithme de poursuite basé sur IEDSEPC.....	65
Figure 3-2 : fonction de récompense appliquée à l'environnement de grille	66
Figure 3-3 : Prévion du comportement des poursuivants	67
Figure 4-1 : Architecture logiciel de La plate-forme JADE.....	72
Figure 4-2 : Architecture Générale du MadKit	73
Figure 4-3 : Interface NetLogo.....	76
Figure 4-4 : Exemple : 02 évadés de type Re = IV après la capture.....	78
Figure 4-5 : Temps de capture (Itérations) après (20) épisodes des poursuites.....	80
Figure 4-6 : Le développement des récompenses des poursuivants	81
Figure 4-7 : Nombre de changement de rôle après (20) épisodes des poursuites	82

Liste des Tableaux

Tableau 3-1: Application de la technique IEDS.....	60
Tableau 3-2 : Algorithme de coalition de poursuite basé sur IEDSEPC	61
Tableau 4-1 : Répartition du calcul des coalitions possibles	78
Tableau 4-2 : résultats de la poursuite.....	79
Tableau 4-3 : moyen de résultat de la poursuite	82

Introduction générale

Contenu

INTRODUCTION GENERALE	1
I Introduction.....	2
II Problématique.....	3
III Objectifs	3
IV Méthodologie de recherche.....	3
V Organisation du mémoire	4

I Introduction

Un système Multi-Agents (SMA) comprend un ensemble d'agents qui interagissent les uns avec les autres. Ces agents peuvent soit partager un objectif commun, soit avoir des objectifs contradictoires [16]. Ce travail concerne des agents coopératifs cherchant à atteindre un objectif commun. Lorsqu'ils coordonnent plusieurs agents en équipe pour la même tâche, ils doivent être en mesure de gérer des situations inconnues et incertaines et de prendre en compte le succès de l'ensemble de l'équipe.

La formation de coalitions, considérée comme un objectif majeur dans les systèmes Multi-Agents, est une méthode de coopération importante. En général, les coalitions sont formées pour atteindre un objectif spécifique et se dissolvent quand il est accompli. La formation de coalitions a fait l'objet d'une attention considérable dans les recherches récentes [34],[35]. En outre, certaines activités de recherche sont basées sur la notification d'organisation qui permet la coalition des agents sous forme de groupes ainsi que la coopération entre leurs membres.

Les jeux Poursuit-Evasion (PE) sont considérés comme l'un des domaines distinctifs de l'intelligence artificielle distribuée traitant le problème de la décision coopérative dans les systèmes Multi-Agents (SMA). Il se concentre principalement sur les approches coopératives utilisées pour connecter de nombreux agents autonomes jouant le rôle de poursuivants pour capturer les évadés mobiles. En fait, le problème de la poursuite nécessite une coordination des actions des poursuivants. Il est largement utilisé en relation avec les différents types de mécanismes de coordination et d'algorithmes de formation de coalitions depuis sa création.

La théorie des jeux peut être considérée comme le moyen le plus simple de modéliser des situations de conflit et d'étudier les interactions entre les agents intéressés. La question classique entre la théorie des jeux et les systèmes Multi-Agents est "quelle est la meilleure action qu'un agent puisse effectuer ?" Ce principe a été largement utilisé dans les problèmes de poursuite Multi-Agents [36][37]. La négociation basée sur la théorie des jeux est centrée sur la valeur et récompenses de chaque agent, qui reflètent de manière appropriée l'objectif de la négociation de l'agent (satisfaire l'objectif de chaque agent). Le principal avantage des algorithmes de la théorie des jeux dans SMA réside dans la coordination apparaissant à travers l'hypothèse de rationalité mutuelle des agents. Par conséquent, les algorithmes de la théorie des jeux sont utilisés pour coordonner des agents rationnels autonomes sans utiliser de mécanisme de coordination explicitement intégré au modèle d'agents. En outre, ils fournissent

différentes méthodes qui définissent les coalitions d'agents optimaux dans plusieurs types de problèmes. Autrement, les inconvénients de ces algorithmes concernent les agents qui sont souvent considérés comme des raisonnements parfaits. De plus, les algorithmes de la théorie des jeux sont axés sur la valeur de la solution optimale et négligent la méthode la plus efficace pour y parvenir.

II Problématique

La poursuite coopérative Multi-Agents est un problème connu de Multi-Agents. Sur la base de conditions identiques entre les poursuivants et les évadés, le problème de poursuite-évasion peut être classé en poursuite à objet unique et poursuite à objet multiple. L'objectif principal des jeux de poursuite-évasion Multi-Agents est de permettre à un groupe de poursuivants de coopérer afin de capturer un autre groupe d'évadés. Dans ce genre de problème, les poursuivants et les évadés sont présentés sous différents types. Le type de poursuivant dénote sa capacité de poursuite. Autrement, le type d'évadé indique le nombre et le type de poursuivants requis pour capturer cet évadé. La valeur d'un évadé indique les récompenses attendues qui devraient être retournées aux poursuivants concernés après la capture.

III Objectifs

La coordination dans les systèmes Multi-Agents basé sur la théorie des jeux a été jugée très importante dans la conception et la mise en œuvre de tout système Multi-Agents. Pour notre problématique, nous proposons un mécanisme efficace et fiable qui nous permet de déterminer les coalitions optimales entre les agents (poursuivants) pour capturer les évadés dans un meilleurs temps et avec une certaines stabilités des groupes, ainsi que l'acquisition des récompenses durant la poursuite.

IV Méthodologie de recherche

Pour atteindre notre objectif, nous proposerons un algorithme de formation de coalition basé sur l'élimination itérative des stratégies dominées avec changement de priorités d'élimination (IEDSEPC), le processus de notre algorithme est répété en utilisant toutes les priorités d'élimination possibles pour obtenir le meilleur équilibre selon ces deux conditions :

- ✓ La maximisation de la récompense moyenne des poursuivants formant le meilleur équilibre.
- ✓ La minimisation des changements de rôle entre l'agent formant le nouvel équilibre optimal et ceux constituant le précédent.

V Organisation du mémoire

Ce mémoire est composé de 5 parties, s'organisant elles-mêmes dans une introduction générale et quatre chapitres.

L'introduction générale détaille la problématique et les objectifs de ce mémoire.

Le premier chapitre 1 fait le tour des différentes notions de définitions et terminologies. Nous abordons les Agents et systèmes Multi-Agents, leurs caractéristiques ainsi que l'importance d'utiliser les SMAs dans plusieurs domaines.

Dans le chapitre 2, nous sommes intéressés aux mécanismes de coordination de tâches dans les SMAs. Nous allons représenter l'importance, et les techniques de coordination, ainsi une description de certains modèles organisationnels, comme nous avons également abordé à la coordination avec l'application de la théorie des jeux dans les systèmes multi agents.

Le chapitre 3 expose notre contribution apportée pour la mise en œuvre d'un mécanisme de coordination de tâches basé sur les principes de la théorie des jeux. Nous présentons dans ce chapitre un algorithme de formation de coalition décentralisée basé sur l'élimination itérative des stratégies dominées avec changement de priorités d'élimination (IEDSEPC). De plus, nous prenons en considération l'application de mécanisme de tâches en relation avec le problème Poursuit-Evasion. Ainsi nous nous concentrons sur l'exploration d'une méthode de planification des tâches et des chemins appliquées aux systèmes Multi-Agents Processus Décisionnel de Markov (MDP)

Enfin, le chapitre 4 concerne la mise en pratique de notre contribution en proposant des études de cas permettant de mesurer l'efficacité, la fiabilité de notre algorithme proposé. Nous présentons dans ce chapitre une étude comparative entre notre mécanisme et un autre mécanisme récent pour montrer que notre mécanisme proposé est réalisable et efficace.

Chapitre 1

1 Agents et systèmes multi-agents

Contenu

1	Agents et systèmes multi-agents	5
1.1	Introduction.....	6
1.2	Pourquoi s'intéresser aux systèmes multi-agents ?.....	7
1.3	Agent	7
1.4	Systèmes multi-agents	13
1.5	Propriétés attendues des SMA.....	17
1.6	Typologie des systèmes multi-agents	18
1.7	Espace SMA	19
1.8	Domaines d'application des SMA.....	20
1.9	Communication dans les systèmes multi-agents	21
1.10	Interactions dans les SMA	23
1.11	Organisations dans les SMA	26
1.12	Conclusion	28

Agent vient de 'agir' : un agent est donc une entité susceptible d'action [3]

1.1 Introduction

La notion d'agent et de système multi-agents (SMA) est relativement récente en informatique, mais elle tend à prendre de plus en plus d'importance. Ce domaine est né au début des années 80, de l'idée de distribuer les connaissances et le contrôle dans les systèmes d'Intelligence Artificielle. Il offre aujourd'hui une alternative intéressante pour la conception, la mise en œuvre ou la simulation et la compréhension de systèmes coopératifs, distribués et ouverts [2], [4], [5]. En effet, l'approche par agents recouvre plusieurs domaines bien différents mais complémentaires à savoir : la résolution des problèmes en intelligence artificielle distribuée où l'on s'intéresse à une vision sociale de la pensée (pouvoir raisonner sur les connaissances et capacités de soi-même et des autres), les systèmes adaptatifs où les problèmes relèvent des domaines sociaux avec leur complexité organisationnelle, le génie logiciel pour l'évolution vers des composants logiciels de plus en plus autonomes et proactifs. Notons que d'autres domaines tels que la vie artificielle, la biologie, la sociologie, les sciences cognitives, etc., ont beaucoup apporté à l'approche agent [6].

La motivation du domaine des SMA est d'essayer de réaliser des entités artificielles (des programmes, des robots) qui interagissent entre elles et avec leur environnement. Une attention particulière est alors portée sur l'aspect multi-agents. Le passage de l'individu (l'agent) au collectif (le système multiagent) a nécessité l'extension et la généralisation de divers concepts issus de l'intelligence artificielle tels que l'apprentissage Multi-Agent, le raisonnement sur les connaissances de groupes ou encore la décision collective [7].

L'importance des SMA repose sur trois concepts fondamentaux :

- l'autonomie de l'activité d'un agent, c'est-à-dire sa capacité d'exécuter une action de sa propre initiative afin d'atteindre une plus grande satisfaction.
- la sociabilité des agents, c'est-à-dire le fait qu'ils agissent en relation avec d'autres agents dans une perspective sociale.
- l'interaction qui est une mise en relation de deux ou plusieurs agents et qui permet à des structures organisées d'apparaître.

Du fait de cette diversité d'influences, il n'est pas surprenant que, encore aujourd'hui, les notions d'agent et de système multi-agents n'aient pu faire l'objet d'un consensus complètement établi, comme le remarquent Jennings et al. [8].

1.2 Pourquoi s'intéresser aux systèmes multi-agents ?

Les systèmes multi-agents sont particulièrement appropriés lorsqu'on s'intéresse aux systèmes complexes. En effet, ils permettent de reproduire le fonctionnement global d'un système complexe à partir des entités qui le compose et de leurs interactions [9]. C'est un nouveau niveau d'abstraction qui permet d'exprimer une application en termes d'agents autonomes qui jouent des rôles et rendent des services dans une organisation [10].

Le développement de l'informatique en termes de puissance des machines, l'évolution des réseaux particulièrement le Web et l'augmentation de la masse d'informations à traiter et à stocker ont engendré des besoins en applications assez complexes. En général, ces dernières sont physiquement et fonctionnellement distribuées. Les systèmes multi-agents répondent à ce genre d'applications, ils permettent la conception modulaire du système. Les modules peuvent être répartis sur plusieurs machines. Un module est plus simple à concevoir qu'un programme monolithique. De plus, la maintenance du système est plus facile, l'amélioration d'un traitement est localisé en général au niveau d'un (ou d'un sous ensemble d') agent (s). L'approche par les systèmes multi-agents propose des solutions robustes et capables de s'adapter dans des environnements qui peuvent être aussi imprévisibles que l'Internet. Par ailleurs, le problème de ce genre de systèmes réside dans la coordination et la gestion des interactions entre les agents. C'est la maîtrise de ces interactions qui permettra aux agents qui sont autonomes, souvent hétérogènes, mais fonctionnant dans un environnement commun de réaliser une fonction cohérente dans un cadre collectif et de former ainsi un système.

1.3 Agent

Il n'y a pas une notion d'agent unique, car le terme agent est utilisé dans de très différentes applications par des communautés venant d'horizons divers. En informatique même, le terme agent est « pluri-défini » puisqu'on trouve dans la littérature une multitude de définitions. Elles se ressemblent toutes, mais elles n'ont pas abouti à une définition consensuelle. Nous présentons dans ce qui suit les plus utilisées :

1.3.1 Définitions

- **Définition 1** [1]

On appelle agent une entité physique ou virtuelle

a) qui est capable d'agir dans un environnement,

b) qui peut communiquer directement avec d'autres agents,

c) qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),

d) qui possède des ressources propres,

e) qui est capable de percevoir (mais de manière limitée) son environnement,

f) qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),

g) qui possède des compétences et offre des services,

h) qui peut éventuellement se reproduire,

i) dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

- **Definition 2:** [11]

“An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives.”

Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu.

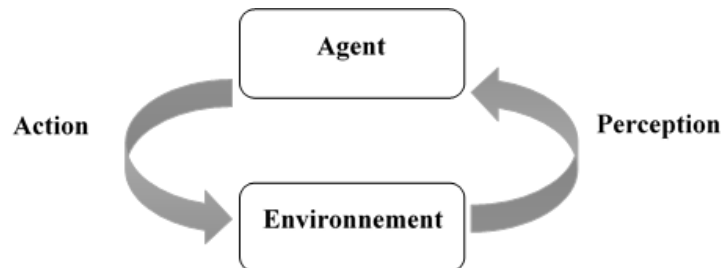


Figure 1-1 : Agent réactif et environnement

- **Définition 3** [12]

On appelle agent toute entité qui peut être considérée comme percevant son environnement grâce à des capteurs et qui agit sur cet environnement via des effecteurs.

L'intersection de ces définitions permet de dégager les principales caractéristiques d'un agent :

- **L'autonomie** : qui correspond à sa capacité à agir sans intervention directe d'un humain ou d'un autre agent, et sous-entend que l'agent a le contrôle de ses propres actions et de son état interne.
- **La réactivité** : un agent doit être capable de percevoir son environnement et de répondre dans les temps requis à un changement intervenant dans cet environnement.
- **La Pro-activité** : un agent tend par lui-même à atteindre des buts (explicites ou non, plus ou moins élaborés).
- **La sociabilité** : qui permet à l'agent d'interagir avec d'autres agents du système quand la situation l'exige afin de réaliser ses tâches ou aider les autres agents à réaliser les leur.

Cependant, d'autres caractéristiques selon les besoins peuvent être incluses dans l'agent telle que l'intelligence, l'adaptabilité, l'intentionnalité, l'engagement etc. La figure suivante présente la structure classique d'un agent où ce dernier est immergé dans un environnement et possède des capteurs et des effecteurs pour assurer sa fonction d'interaction via ou avec son environnement [13], naturellement il doit être doté de la capacité de décision qui lui permet de réagir :

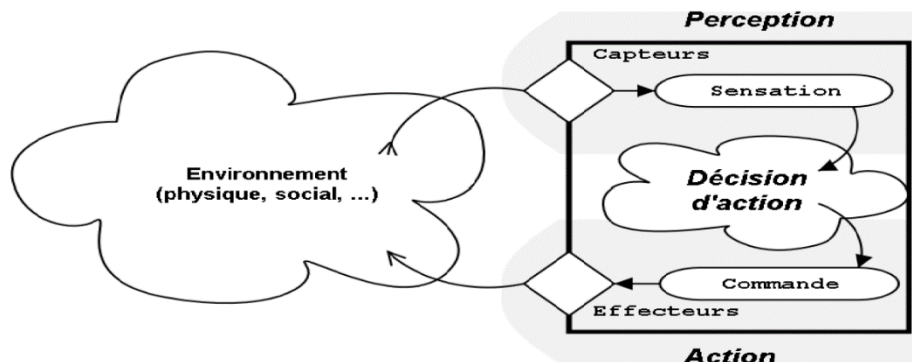


Figure 1-2 : Structure classique d'un agent

1.3.2 Typologie des agents

De manière générale, il est possible de classer un agent selon les trois principaux points suivants :

1.3.2.1 Selon la nature

Le terme d'architecture interne désigne l'ensemble des structures de données et des processus internes à un agent lui permettant de prendre une décision consistant à choisir une action en vue de modifier l'environnement. Bien que la frontière soit relativement floue, la communauté multi-agents distingue deux grandes catégories d'architectures d'agents : les architectures *cognitives* et les architectures *réactives* [1], [14].

La distinction qu'on peut faire entre cognitif et réactif tient essentiellement du processus décisionnel chez l'agent et de la représentation du monde dont il dispose. Si l'agent est doté d'une représentation symbolique du monde à partir de laquelle il est capable de formuler des raisonnements, on parle d'agent cognitif tandis que s'il ne dispose que d'une représentation symbolique, c'est-à-dire limitée à ses perceptions, on parle d'agent réactif [7]. Cette distinction cognitif/réactif correspond à deux écoles de pensée des systèmes multi-agents. La première soutient une approche de famille d'agents « intelligents » devant collaborer, avec une perspective plus sociologique. La deuxième étudie la possibilité de l'émergence d'un comportement « intelligent » d'un ensemble d'agents non-intelligents. Dans les faits, dans beaucoup d'approches multi-agents pour la résolution de problèmes la distinction entre agent réactif et cognitif n'est pas aussi nettement tranchée. Des comportements de type réactif et cognitif peuvent être combinés.

- **Les agents réactifs** : souvent qualifiés de peu intelligents, les agents réactifs possèdent une représentation très simplifiée de leur environnement. Leurs capacités consistent à réagir uniquement en mode stimulus/ action vu comme une forme de communication (ou perception) [15]. Un SMA constitué d'agents réactifs possède généralement un grand nombre d'agents (des milliers) et le succès, dans ce type de systèmes, porte sur l'émergence d'un comportement collectif intelligent. Bien que très simples dans leur conception, un ensemble d'agents réactifs peut avoir un comportement complexe, il s'agit de l'émergence d'un comportement. Ainsi, par exemple un système multi-agents reproduisant le comportement de fourmis pour trouver le chemin le plus rapide entre deux points d'un réseau dont le nombre de nœuds et de passerelles est inconnu fait émerger une intelligence collective dans son comportement.

- **Les agents cognitifs** : possèdent une représentation (partielle mais sophistiquée) de leur environnement, ont des buts explicites et sont capables de planifier leur comportement, de mémoriser leurs actions passées, de communiquer par envoi de messages ou via des langages d'interaction élaborés, de négocier, etc. Un SMA constitué d'agents cognitifs possède généralement un nombre restreint d'agents (des dizaines). Les agents cognitifs peuvent être intentionnels (dotés d'attitudes intentionnelles telles que les croyances, les désirs et les intentions), rationnels (agissant selon une rationalité donnée telle que la rationalité économique), normés (agents évoluant dans un système doté de normes sociales), etc.

L'une des architectures cognitives les plus connues est sans doute l'architecture BDI : Belief (Croyance), Desire (Désir), Intention (Intention) qui comme son nom l'indique, est basée sur les notions d'attitudes mentales que sont la croyance, le désir et l'intention :

- Les croyances correspondent aux informations dont dispose l'agent sur son environnement.
 - Les désirs correspondent aux états de l'environnement que l'agent souhaiterait voir réalisés.
 - Les intentions correspondent aux projets de l'agent pour satisfaire ses désirs.
- **Agents hybrides** : d'autres types d'agents qualifiés d'hybrides, utilisant donc ces deux types de comportement, sont ensuite apparus. Les agents hybrides intègrent l'aspect cognitif et réactif. L'idée est de combiner les deux types d'approches qui peuvent être vues comme complémentaires. Dans une telle architecture un agent est composé de modules qui gèrent indépendamment la partie réactive et la partie cognitive. Le problème central reste de trouver le mécanisme idéal assurant cette combinaison.

Partageant l'opinion d'un nombre croissant de chercheurs en IA, nous croyons que ni les architectures réactives, ni les architectures cognitives n'offrent la solution unique. En fonction du contexte, certaines architectures sont plus adaptées que d'autres.

Le choix des types d'agents dépend du système multi-agents le plus pertinent pour le problème à résoudre. Ainsi, certains SMA n'utilisent qu'un seul type d'agents regroupés par objectif, d'autres plusieurs types correspondant à des rôles précis nécessaires à la résolution.

1.3.2.2 Selon l'utilisation

- **Agents collaboratifs** : Ces agents ont des capacités de coopération. Un regroupement de ces agents permet, entre autres, de réduire un problème complexe en sous problèmes moins complexes.
- **Agents d'interface** : Ces agents collaborent avec l'utilisateur pour effectuer certaines tâches telles que les activités de bureautique, les systèmes tuteurs intelligents qui doivent faciliter l'apprentissage et diminuer la surcharge cognitive chez l'apprenant.
- **Agents pour la recherche d'informations** : Ces agents effectuent, en premier lieu, une recherche d'informations parmi une collection de données et, en second lieu, procèdent à une analyse des informations utiles trouvées afin de découvrir de nouvelles connaissances.
- **Agents pour le commerce électronique** : La montée de l'Internet a bien entendu créé de nouvelles nécessités. Les agents issus de cette tendance permettent la promotion, la vente ainsi que l'achat de produits et de services par l'entremise des réseaux informatiques, etc.
- **Agents conversationnels animés** : ce sont des interfaces de dialogue entre des utilisateurs et des systèmes d'information. Ils se déploient sur des sites Internet, notamment des sites marchands. Ils sont pourvus de bases de dialogues correspondant aux contextes d'interaction dans lesquels ils agissent.
- **Agents guide ou assistants** : ce type d'agents essaient de suggérer des sites susceptibles d'intéresser l'utilisateur, en observant son comportement sur Internet. Le principe est assez simple. Généralement les sites que l'on visite sur Internet reflètent les goûts ou les besoins de l'utilisateur. Ainsi, en analysant ses habitudes de navigation, un « agent assistant » tente d'apprendre des habitudes de son utilisateur et de lui suggérer des sites en relation avec ce qu'il désire.

1.3.2.3 Selon la technologie employée

- **Agents stationnaires** : Il s'agit du cas où l'agent s'exécute toujours sur la même machine.
- **Agents mobiles** : Ces agents s'exécutent sur différentes machines en se promenant d'un hôte à l'autre. Typiquement, ils suivent ce que l'on appelle un « itinéraire ».

1.3.3 Propriétés clés des Agents

- **Autonomie** : un agent peut agir sans l'intervention directe d'un tiers (agent ou humain) et contrôler ses actions ainsi que son état interne.
- **Réactivité** : un agent perçoit son environnement et répond aux changements qui s'y produisent en un temps raisonnable. L'environnement peut être le monde physique, un utilisateur via une interface graphique, d'autres agents, un système d'information (internet), ...
- **Communication** : un agent peut communiquer avec d'autres agents ainsi qu'avec des utilisateurs humains.
- **Aptitude sociale** : un agent peut interagir avec d'autres agents de façon coopérative ou compétitive pour atteindre ses objectifs.
- **Pro-activité** : l'agent est capable, sur sa propre initiative, de se fixer des buts pour atteindre ses objectifs (opportuniste).
- **Agent Intelligent** : un agent est « intelligent » s'il est capable de réaliser des actions *flexibles* et *autonomes* pour atteindre les objectifs qui lui ont été fixés. La flexibilité correspond aux propriétés de Réactivité, Pro-activité, Aptitudes sociales.

1.3.4 Propriétés additionnelles

- **Apprentissage** : un agent peut mémoriser ses expériences et adapter son comportement en conséquence.
- **Mobilité** : un agent peut se déplacer d'une machine à une autre et éventuellement se dupliquer.

1.4 Systèmes multi-agents

Au premier abord, un système multi-agents peut d'une manière simpliste être considéré comme un ensemble d'agents partageant un environnement commun. Certes, la notion d'environnement est primordiale dans un système multi-agents mais elle reste, tout de même insuffisante pour qualifier un système multi-agents. En outre, de la même manière qu'il n'est pas évident de définir d'une manière unique ce qu'est un agent, il est difficile de caractériser un système multi-agents. Ainsi, il est question d'identifier les aspects fondamentaux qui caractérisent un système multi-agents d'un système mono-agent.

1.4.1 Définitions

1.4.1.1 Définition 1

Ferber [1] propose une définition très utilisée dans la littérature jusqu'à nos jours [16].

On appelle système multi-agents un système composé des éléments suivants :

- Un environnement E, disposant en général d'une métrique,
- Un ensemble d'objets O, auxquels on peut associer une position dans E à un moment donné. Ces objets (hormis les agents) sont passifs : les agents peuvent les percevoir, les créer, les détruire et les modifier.
- Un ensemble d'agents A, lesquels représentent les entités actives du système,
- Un ensemble de relations R, qui unissent les objets (et agents) entre eux,
- Un ensemble d'opérateurs Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O,
- Et, des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

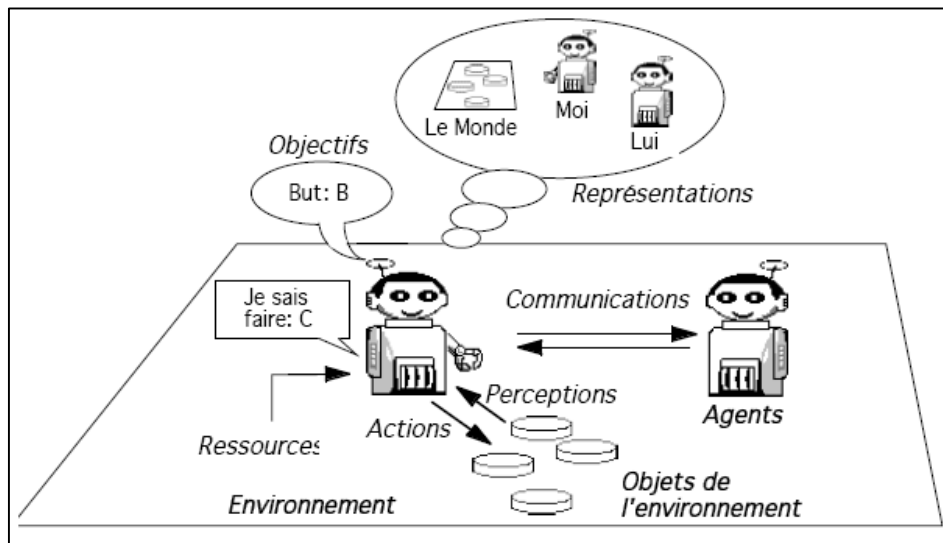


Figure 1-3 : Environnement d'un système multi agents

1.4.1.2 Définition 2

Selon Sycara, Jennings et al. [4], le terme système Multi-Agents correspond à *un réseau faiblement couplé de solveurs de problèmes qui travaillent ensemble pour résoudre des problèmes qui sont au-delà des capacités individuelles ou des connaissances de chacun d'eux.*

Toutefois, le terme système Multi-Agents dans le sens le plus général désigne les types de systèmes constitués de composants autonomes, dont la principale caractéristique réside dans l'exécution décentralisée des tâches : chaque agent agit de manière autonome mais aucun d'eux ne contrôle totalement la dynamique du système. Le comportement collectif d'un tel système provient du couplage à l'exécution des différents comportements des entités qui y sont présentes. Ce sont ces couplages entre les comportements des agents qui permettent de construire des réponses plus complexes que les comportements des agents isolés.

Un système multi-agents est ainsi caractérisé par plusieurs niveaux d'observations :

- un niveau local ou individuel dans lequel les agents sont décrits et prennent leurs décisions. C'est à ce niveau que le comportement d'un agent par rapport au reste du système peut être observé.
- un niveau global ou collectif dans lequel il est possible d'observer la dynamique globale du système et l'avancement de la résolution du problème.

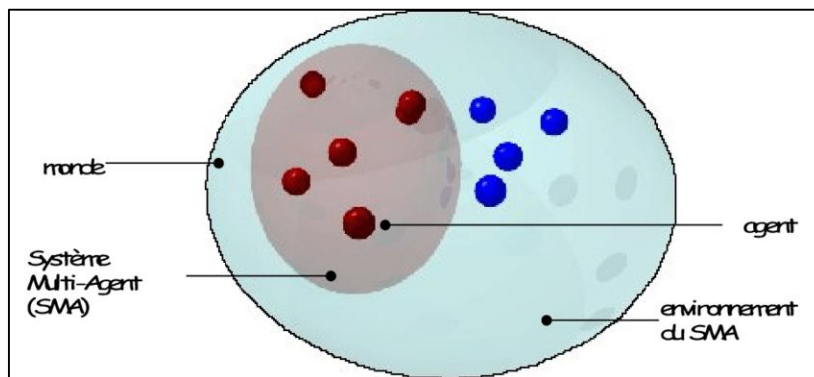


Figure 1-4 : Positionnement d'un agent dans le monde

1.4.2 Environnement

1.4.2.1 Définition

L'environnement est communément défini par tout ce qui entoure l'agent. C'est l'endroit d'immersion dans lequel évoluent les agents. L'environnement peut être décomposé en deux parties. L'environnement social qui est composé des autres agents du système et l'environnement physique au sein duquel les agents évoluent.

1.4.3 Propriétés des environnements

La nature de l'environnement ainsi que ses propriétés influent de manière considérable les actions de l'agent, nous présentons dans ce qui suit les propriétés [12] les plus importantes :

- **Entièrement observable par opposition à partiellement observable** : si l'agent a accès à la totalité de l'environnement à tout moment, ce dernier est dit entièrement observable, l'agent peut obtenir dans ce cas une information complète, exacte et à jour sur l'état de son environnement. Dans un environnement partiellement observable ou inaccessible, seule une information partielle est disponible.
- **Déterministe par opposition à non déterministe ou stochastique** : si l'état suivant de l'environnement est complètement déterminé par l'état courant et par l'action qu'exécute l'agent, on dit que l'environnement est déterministe, sinon, il est stochastique. Dans un environnement non déterministe, une action n'a pas un effet unique garanti. Toutefois, si l'environnement est partiellement observable, il pourra paraître stochastique.
- **Dynamique par opposition à statique** : si l'environnement peut changer alors qu'un agent est entrain de prendre une décision (délibérer), on dit qu'il est dynamique sinon il est statique. L'état de l'environnement dynamique dépend des actions du système qui se trouve dans cet environnement, mais aussi des actions d'autres processus. Ainsi, les changements ne peuvent pas être prédits par le système. Un environnement statique ne change pas sans que le système agisse.
- **Continu par opposition à discret** : Le nombre d'actions et de perceptions possibles dans un environnement continu est infini et indénombrable. Dans le cas discret, l'environnement est défini par un nombre d'états.

- **Mono-agent ou par opposition multi-agents** : La distinction entre environnements mono-agents et multi-agents peut sembler relativement simple. Un agent qui résout un problème seul ne doit pas considérer les évènements et actions exercés par les autres agents.

1.5 Propriétés attendues des SMA

Le fait d'avoir plusieurs entités dans le système les propriétés en termes de rationalité et d'autonomie doivent être précisées.

1.5.1 Rationalité

Tout d'abord, la rationalité peut s'exprimer de manière différente selon les types de systèmes multi-agents :

- Pour certains systèmes, chaque agent dispose d'une rationalité propre et mesure de manière individuelle sa performance. Cette vision des choses est principalement compétitive, chaque agent cherche alors à maximiser ses gains (critères de performance) éventuellement au détriment des autres agents du système.
- D'autres systèmes s'intéressent à une mesure de performance globale : le système est caractérisé par une fonction globale de performance (éventuellement calculée à partir d'une combinaison de mesures de performance locales). La rationalité est forcément limitée puisque l'agent ne connaît pas le monde ni surtout le comportement des autres agents. Nous soulignons la présence de capacité sociale permettant d'estimer les comportements des autres agents pour agir au mieux par la suite. Les systèmes de cette catégorie seront intelligents, collectifs et coopératifs du fait qu'une mesure de performance globale caractérise les performances du système.

1.1.1 Autonomie

La notion d'autonomie d'un agent s'exprime elle aussi de manière différente dans un système multi-agents :

- La définition de **Russel** et **Norvig** [12] correspond à la capacité de l'agent à s'adapter à partir de ses expériences passées. Dans un système multi-agents, cette capacité inclut la capacité à s'adapter à l'environnement global du système mais aussi aux autres agents présents dont le comportement est inconnu et peut évoluer au cours du temps.

- Celle de Jennings [11], implique que chaque agent décide de manière autonome de son action sans intervention extérieure. Cette définition implique que toute action émise dans le système est uniquement à l'initiative d'un agent.

Nous pouvons dire que l'autonomie dans un système multi-agents coopératif doit permettre à l'ensemble du système d'évoluer.

1.6 Typologie des systèmes multi-agents

De la même manière qu'il existe deux types principaux d'agents, on distingue les systèmes multi-agents en fonction du type des agents qui les composent. On compte ainsi deux grandes classes de systèmes à savoir les systèmes multi-agents cognitifs et les systèmes multi-agents réactifs.

1.6.1 Les systèmes multi-agents cognitifs

Ce sont des systèmes constitués d'agents cognitifs généralement en faible nombre. Leur étude est fondée sur l'idée que la construction de systèmes multi-agents « intelligents » peut se faire à partir d'agents dont les capacités de représentation sont complexes et pouvant disposer de processus de communication élaborés. De tels systèmes peuvent mettre en œuvre et tirer profit de mécanismes complexes de représentation des autres, de négociation et d'échanges d'information. Néanmoins cette approche se heurte au problème de la représentation des connaissances déjà présent pour des agents isolés. Toutefois, la représentation complexe des autres agents du système limite le nombre d'agents et peut mener à des raisonnements souvent coûteux. A cela, s'ajoute enfin la complexité des processus de communications entre agents.

1.1.2 Les systèmes multi-agents réactifs

Ces systèmes sont composés d'agents réactifs souvent en grand nombre. De tels systèmes se basent sur l'hypothèse qu'il est possible de produire des comportements collectifs intelligents complexes malgré la simplicité des comportements individuels. Dans un système multi-agents, un comportement global complexe peut apparaître comme le résultat des interactions entre des composants simples en grande quantité. De nombreux phénomènes naturels peuvent s'expliquer de cette manière (par exemple systèmes inspirés des sociétés d'insectes comme les colonies de fourmis, les réseaux neuronaux, etc.). Les systèmes multi-agents réactifs présentent habituellement des intérêts en termes d'émergence de comportements intelligents, qui lui-même présente des difficultés de prédiction du comportement global non représenté explicitement dans

le système et de contrôle des comportements individuels par rapport à un objectif non représenté explicitement.

Nous distinguons de plus les systèmes multi-agents homogènes et les systèmes multi-agents hétérogènes, le premier type de systèmes concerne les systèmes composés d'agents de même type et donc de même architecture interne, le second type comme son nom l'indique inclue les systèmes composés d'agents d'architectures internes différentes. Notons que les systèmes actuels sont généralement hétérogènes [16].

1.7 Espace SMA

Les systèmes multi-agents constituent un espace qui s'articule autour de trois axes principaux et nécessaires pour la conception et la mise en œuvre de systèmes complexes :

- L'axe cognition fait référence aux aptitudes cognitives d'un agent telles que l'apprentissage, le raisonnement, la prise de décision, la planification autonome, etc... Ces aptitudes permettent aux agents de s'adapter aux changements de leur environnement. Cependant, les agents étant dans un environnement multi-agents, leurs capacités doivent être considérées dans un contexte collectif (apprentissage multi-agents, décision distribuée, connaissances communes et/ou distribuées, etc.) car c'est bien le passage du niveau individuel au niveau collectif qui caractérise ces systèmes.
- L'axe interaction fait référence aux mécanismes de communication entre agents qui leur permettent d'échanger l'information, de partager des objectifs et de coopérer. Les interactions peuvent être de haut niveau utilisant des langages sophistiqués (comme KQML, FIPA-ACL) manipulant des théories de la communication comme les actes de langage, les théories de discours ou encore des modèles symboliques d'argumentation, ou elles peuvent se dérouler via l'environnement partagé par les différents agents.
- L'axe concurrence signifie que les agents peuvent s'exécuter en parallèle et mener des activités concurrentes. Ceci permet aux agents de distribuer la résolution des tâches et de l'optimiser, de concourir à l'affectation de tâches, ou enfin concourir à l'allocation de ressources.

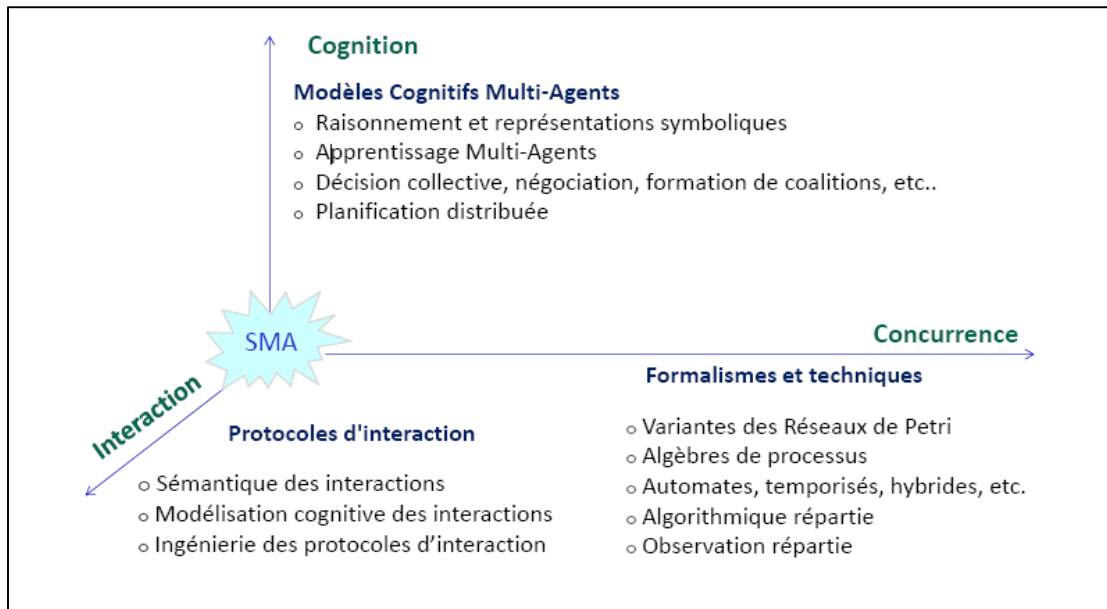


Figure 1-5 : Espace SMA [7]

Les SMA peuvent être placés tous dans cet espace [7], certains peuvent avoir des agents très forts en cognition tels que les agents joueurs aux échecs ou d'autres très faibles tels que les agents fourmis. Certains peuvent avoir des agents très performants en interaction (agents utilisant les ACLs) ou d'autres très faibles (agents interagissant via l'environnement).

1.8 Domaines d'application des SMA

Les applications des systèmes multi-agents couvrent de plus en plus de domaines. Citons les systèmes d'information coopératifs, la simulation sociologique, les outils documentaires adaptés au Web, les robots autonomes coopératifs, jeux vidéo (multi-joueurs), résolution distribuée de problèmes, etc. Néanmoins les systèmes multi-agents développés actuellement peuvent être classés en trois catégories [2], [18], [26] :

- Les simulations dont l'objectif est la modélisation de phénomènes du monde réel, afin d'observer, de comprendre et d'expliquer leur comportement et leur évolution. Les systèmes multi-agents ont trouvé rapidement un champ extrêmement propice à leur développement dans le domaine de la modélisation de systèmes complexes ne trouvant pas de formalisation mathématique adaptée. Dans le domaine des sciences du vivant d'abord, ensuite dans celui des sciences humaines et sociales, les SMA ont montré qu'il était possible de modéliser au niveau micro les comportements d'entités élémentaires et d'étudier au niveau macro le résultat global de l'interaction de ces entités. En effet, la

simulation multi-agents permet de tester rapidement le changement de certaines hypothèses, elle permet aussi d'intégrer de nouveaux agents et d'éditer, sur un plan pratique, les résultats pour comparer les expérimentations les unes aux autres, de plus elle permet de préserver l'hétérogénéité du système à simuler.

- Les applications dans lesquelles les agents jouent le rôle d'êtres humains. La notion d'agent simplifie la conception de ces systèmes et amène de nouvelles problématiques centrées utilisateur telles que la communication, la sécurité... Les systèmes de ventes aux enchères dans laquelle les agents jouent les rôles de commissaire-priseur et d'acheteurs, représentent une classe d'applications de cette catégorie.
- La résolution de problèmes : telle qu'elle avait été définie en Intelligence Artificielle, étendue à un contexte distribué. Dans ce cadre, l'objectif est de mettre en œuvre un ensemble de techniques pour que des agents, pertinents pour la résolution d'une partie ou l'ensemble du problème, participent de manière efficace et cohérente à la résolution du problème global.

1.9 Communication dans les systèmes multi-agents

1.9.1 Définition

« **La communication** est l'échange intentionnel d'informations occasionné par la production et la perception de signes issus d'un système partagé de signes conventionnels » [12]

Les communications dans les SMA comme chez les humains sont à la base des interactions et de l'organisation sociale. Sans communication, l'agent n'est qu'un individu isolé. C'est parce que les agents communiquent qu'ils peuvent coopérer, coordonner leurs actions et réaliser des tâches en commun [18] [27].

1.9.2 Types de communication

Les communications dans les SMA peuvent être classées en deux classes distinctes, à savoir communications directes et indirectes.

1.9.2.1 Communications indirectes (ou implicites)

Ce sont des mécanismes fondés sur l'usage de signaux qui prennent leurs sources en biologie et en éthologie. En effet, les agents dans de tels systèmes peuvent laisser des traces (signaux) de leur présence ou de leur action qui sont perçues et interprétées par d'autres agents (tels que les phéromones chez les fourmis). Dans d'autres systèmes, les agents dans l'environnement émettent des signaux ou des champs de potentiels guidant les autres agents dans un but de compétition ou de coopération (tels que les agents robots qui peuvent émettre des signaux pour attirer ou repousser d'autres agents de leur environnement).

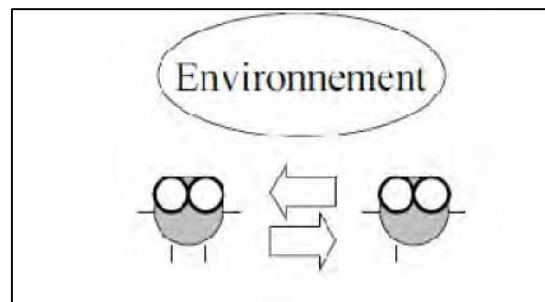


Figure 1-6 : Communication indirecte

1.9.2.2 Communications directes (ou explicites)

Ce modèle de communication entre agents repose sur la communication par message. L'échange direct est réalisé volontairement en direction d'un individu ou d'un groupe d'individus. Dans ce type de communication les agents sont en liaison directe, les messages sont envoyés directement et explicitement de l'émetteur au destinataire et ce via un canal de communication.



Figure 1-7 : Communication directe

1.10 Interactions dans les SMA

La notion d'interaction constitue l'essence d'un système multi-agents puisque c'est grâce à elle que les agents vont pouvoir produire des comportements collectifs complexes et dépendants les uns des autres. En effet, la fonction interactionnelle d'un agent porte sur l'ensemble des mécanismes lui permettant de faire le lien avec ce qui l'entoure (son environnement ainsi que l'ensemble des autres agents). L'interaction peut être vue comme une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques [28]. Elle représente de plus un élément nécessaire à la constitution des organisations.

1.10.1 Définition

« *On appelle situation d'interaction un ensemble de comportements résultant du regroupement d'agents qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent et de leurs compétences individuelles* » [1].

De cette définition on peut faire sortir une classification des situations d'interaction selon plusieurs critères :

- la présence d'objectifs *communs* ou *compatibles*
- l'accès à des ressources communes
- la répartition des compétences au sein des agents.

En fonction de ces critères et de l'objectif du système, la notion de situation d'interaction peut s'exprimer comme les attitudes adoptées par les agents vis-à-vis des autres agents. **Ferber** [1] distingue trois grandes catégories d'interactions :

- **l'antagonisme entre agents** : les agents ont des objectifs conflictuels (compétition) ou ont besoin de ressources communes (conflit sur les ressources).
- **l'indifférence entre agents** : les agents n'ont pas besoin les uns des autres pour atteindre leurs objectifs et ne sont pas gênés par ceux-ci.
- **la coopération entre agents** : les agents doivent s'entraider pour atteindre leurs objectifs qui peuvent être éventuellement communs.

En effet, un des aspects importants de la dynamique d'un SMA est la nature des interactions entre ses entités, puisqu'elles constituent un support effectif de la coopération [29]. Les agents interagissent en vue de coopérer et de coordonner leurs actions afin d'atteindre des buts locaux (individuels) ou globaux (collectifs). L'interaction est réalisée à travers un langage compréhensible et commun à tous les agents et peut être :

- sélective sur un nombre restreint d'agents (par exemple en fonction des accointances c'est-à-dire les agents qu'il connaît).
- ou étendue à l'ensemble des agents c'est-à-dire par diffusion.

De plus, les interactions permettent dans les systèmes multi-agents la combinaison des fonctionnalités des agents pour faire émerger le comportement global du système [30].

D'autre part, une interaction du point de vue d'un agent peut être dans à l'une des trois catégories suivantes selon le gain perçue :

- incidence nulle.
- incidence positive : l'agent perçoit l'interaction comme une aide.
- incidence négative : l'agent perçoit l'interaction comme une gêne.

1.10.2 Coopération

Les études dans le domaine de la sociologie ont montré que les humains doivent coopérer à cause de leurs capacités limitées. Nous coopérons parce que nous ne pouvons pas atteindre nos buts, ou parce que nous ne pouvons pas les atteindre de manière efficace et rapide.

En effet, la coopération est la forme générale d'interaction la plus étudiée dans les SMA, elle représente l'attitude sociale qui permet l'augmentation des performances de groupe [19], [31], [32]. De plus la coopération se fonde à la fois sur la complémentarité d'intérêt et la confiance [20]

Nous nous intéressons particulièrement à cette forme d'interaction, nous présentons dans ce qui suit deux définitions qui nous semblent répondre à ce dont nous avons besoin.

Définition 1 :

Les caractéristiques de la coopération idéale sont aussi celles d'une coopération totale où la moindre activité est bénéfique pour autrui [18] :

- **Compréhension** : un signal perçu doit être interprétable par un système coopératif. La compréhension mutuelle n'a pas à être postulée mais doit émerger de l'ajustement mutuel entre le système et son environnement.
- **Raisonnement** : toute information (un signal interprété) doit avoir des conséquences logiques dans le système. En d'autres termes, toute information doit apporter de la nouveauté (en différence avec les informations actuellement mémorisées).
- **Action** : les conclusions du processus de raisonnement doivent être utiles à l'environnement du système.

Définition 2 :

On dira que plusieurs agents coopèrent ou encore qu'ils sont dans une situation de coopération si l'une des conditions suivantes est vérifiée :

- l'ajout d'un nouvel agent permet d'accroître différenciellement les performances du groupe
- l'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels [1]. propose un certain nombre de méthodes permettant de mettre en œuvre cette attitude coopérative entre agents comme :
- la communication pour échanger des informations
- le regroupement physique des agents
- la spécialisation pour rendre certains agents plus adaptés à leur tâche
- la répartition des tâches, des informations et des ressources
- la coordination d'actions qui correspond à l'exécution de tâches supplémentaires permettant d'exécuter d'autres actions critiques dans les meilleures conditions.

De plus, certains chercheurs distinguent la coopération indirecte qui est due aux actions individuelles émises par les agents faisant évoluer l'environnement et la coopération directe résultante des signaux directs émis par les agents.

De ces définitions nous déduisons qu'un système coopératif doit avoir les caractéristiques suivantes :

- c'est un système composé d'agents coopératifs, ce sont des agents qui s'entraident, se complètent et se comprennent.

- ce sont des agents qui doivent accomplir la tâche pour laquelle ils ont été conçus dans un milieu coopératif.
- le maintien d'un degré de coopération élevé assure la survie du système.

1.11 Organisations dans les SMA

Lorsqu'on parle d'organisations on suppose l'existence d'un ensemble d'entités en relation formant une unité.

Dans un système multi-agents, l'organisation permet aux agents de savoir quels sont leurs partenaires et quels rôles ils jouent de façon à répondre à un objectif donné. C'est un arrangement des agents et de leurs comportements conditionné par les contraintes imposées par l'environnement [17], [21]. C'est une structure du système [22].

1.11.1 Définition

Plusieurs définitions du terme organisation existent dans la littérature, naturellement ce terme possède différents sens dans différents domaines de recherche tels que les sciences humaines, les sciences économiques. Nous présentons dans ce qui suit deux définitions que l'on rencontre dans le domaine des systèmes multi-agents :

L'organisation est agencement de relations entre composants ou individus qui produisent une unité ou système dotée de qualités inconnues au niveau des composants ou individus. L'organisation lie de façon interrelationnelle des éléments ou événements ou individus divers qui dès lors deviennent les composants d'un tout. Elle assure solidarité et solidité relative, donc assure une certaine possibilité de durée en dépit de perturbations aléatoires [1].

Une organisation est un ensemble d'individus regroupés au sein d'une structure régulée, ayant un système de communication pour faciliter la circulation de l'information, dont le but est de répondre à des besoins et d'atteindre des objectifs déterminés [16].

Dans un système multi-agents, l'organisation est le facteur structurant, elle permet aux agents de savoir quels sont leurs partenaires et quels rôles ils jouent de façon à répondre à un objectif donné. Ferber [1] insiste sur la dualité entre l'aspect statique et l'aspect dynamique de l'organisation en précisant que « l'organisation est à la fois le processus d'élaboration d'une structure et le résultat même de ce processus ».

[33] précisent que l'organisation est le *résultat d'un comportement* social des agents et elle ne peut être créée explicitement par le concepteur.

1.11.2 Types d'organisation

Dans la littérature [23], [24], [25] on distingue plusieurs types d'organisations :

- **Groupe** : plusieurs types de groupes existent :
 - **Groupe simple** : dès qu'un groupe existe, on peut avoir une coordination coopérative afin d'atteindre un but commun et partagé.
 - **Équipe** : une collection d'individus qui ont été rassemblés pour travailler ensemble. Dans une organisation plusieurs équipes sont formées pour des raisons de communication. Dans cette définition, des individus appartenant à une équipe doivent nécessairement communiquer entre eux, ce qui entraîne l'introduction de la notion d'environnement (cadre dans lequel les agents existent et évoluent).
 - **Groupe d'intérêts** : chaque membre a les mêmes intérêts, ils partagent les informations et coopèrent pour réaliser un but commun.
 - **Communauté de pratique** : se constitue lorsque des professionnels se regroupent et s'organisent pour partager des informations et des expériences relatives à leurs activités. Les membres de ces communautés peuvent ainsi échanger et coopérer afin de résoudre ensemble les problèmes auxquels ils peuvent être confrontés, apprendre ainsi les uns des autres et construire ensemble des connaissances et des pratiques communes.
- **Hiérarchie** où l'on distingue :
 - **Hiérarchie simple** : basée sur une relation maître/esclave, ce type d'organisation n'est plus utilisé.
 - **Hiérarchie multi niveaux** : les liens d'autorité forment un arbre. Le contrôle dans ce type d'organisation est très complexe, citons par exemple le problème d'allocation de ressources ou de planification qui doivent être pris en compte.
- **Organisation décentralisée** : c'est une hiérarchie multi divisions où chaque sommet d'une branche est une organisation à part entière. La difficulté principale dans ce type d'organisations est l'intégration des différents résultats provenant des différentes divisions.

- **Marché** : ce type d'organisation se base sur la relation consommateur/producteur. Une instance spéciale du marché est le Contract Net Protocol qui est un protocole permettant l'élaboration et l'exécution d'un contrat entre un agent *manager* et un agent *contractant*. Il fait intervenir des agents interagissant entre eux pendant l'élaboration et l'exécution du contrat au moyen de performatifs.
- **Coalitions** : Une coalition est une organisation à court terme basée sur des engagements spécifiques et contextuels permettant aux agents de bénéficier de leurs compétences respectives de façon opportuniste.

1.12 Conclusion

Les SMA proposent aujourd'hui une nouvelle technologie effective de mise en œuvre de systèmes complexes dès lors que ceux-ci requièrent distribution, ouverture, coopération et autonomie ajustable. Ces systèmes forment une communauté de recherche issue de plusieurs influences. Les principales sont :

- l'intelligence artificielle distribuée.
- la vie artificielle ou les systèmes inspirés d'organismes vivants.
- les sciences sociales et les sciences cognitives.

Chapitre 2

2 Les mécanismes de coordination de tâches dans les SMAs

Contenu

2	Les mécanismes de coordination de tâches dans les SMAs.....	29
2.1	Introduction.....	30
2.2	L'importance de la coordination	30
2.3	Les techniques de coordination	32
2.4	Critères pour choisir une technique de coordination	40
2.5	Organisations dans les systèmes multi-agents	42
2.6	Organisation et modèle organisationnel.....	43
2.7	Modèles organisationnels	43
2.8	Théorie des jeux dans les systèmes multi-agents	49
2.9	La coordination avec l'application de la théorie des jeux	53
2.10	Conclusion	55

2.1 Introduction

Dans les sociétés humaines, il existe trois types d'interrelations : coopération, coordination et concurrence. La coopération peut être perçue comme une coordination entre des personnes non antagonistes, tandis que la négociation est une coordination entre des concurrents ou des personnes auto-intéressées. Tout comme il existe une myriade de constructions de coordination dans les sociétés humaines telles que les entreprises, les gouvernements, les marchés, les comités, etc., de même il n'existe pas de stratégie de coordination unique qui fonctionne bien dans toutes les circonstances. Ce scénario indique ce qui se passe dans un environnement multi-agents.

La coordination des agents est la capacité fondamentale d'un agent de décider de ses propres actions dans le contexte des activités d'autres agents. [36] Par conséquent, dans les systèmes d'agents logiciels peuplés de nombreux agents hétérogènes, la coordination des agents est une préoccupation centrale. Les avancées en matière de génie logiciel orienté agent permettent de développer des systèmes complexes et distribués. Sans coordination, les agents gaspilleront involontairement leurs efforts et leurs ressources, échouant souvent dans la réalisation de leurs objectifs individuels et collectifs, et éventuellement dans l'arrêt du système. La coordination est essentiellement la propriété d'une agence intelligente qui veille à ce que chaque agent agisse de manière cohérente.

2.2 L'importance de la coordination

Dans un système à agents multiples, il est souhaitable de pouvoir considérer les agents collectivement, en tant que société. Tout comme dans la société humaine, nous avons le dicton selon lequel «aucun homme n'est une île», indiquant le fait qu'aucune personne ne suffit à elle seule, il en va de même pour le monde des agents. Avec l'augmentation des réseaux informatiques, il devient de plus en plus rare de trouver des agents autonomes. Le désir de faire en sorte que les agents communiquent entre eux et se partagent le fardeau des tâches a conduit à la création d'un nouveau domaine de recherche et d'application appelé intelligence artificielle distribuée (DAI). DAI s'appuie sur des concepts issus d'un large éventail de domaines, notamment l'informatique, la sociologie, l'économie, la philosophie et les sciences de gestion, et peut être défini comme «l'étude, la construction et l'application de systèmes multi-agents, systèmes dans lesquels plusieurs agents intelligents en interaction poursuivre un ensemble d'objectifs et accomplir un ensemble de tâches. »[2] L'une des tâches de DAI consiste à développer des mécanismes et des méthodes permettant aux agents d'interagir de manière cohérente en tant qu'unité.

La coordination n'implique pas la coopération : un concurrent efficace coordonnera ses décisions afin de maximiser son avantage sur un adversaire, par exemple une entreprise programmant une promotion de produit pour saper un concurrent. Cela n'implique pas non plus de réciprocité : un agent peut coordonner ses actions avec un autre agent inconscient de sa présence, par exemple lorsqu'un automobiliste croise un deuxième conducteur dont l'esprit est entièrement ailleurs [36]. Ainsi, la coordination peut être réalisée sans communication [37], à condition que les agents impliqués possèdent des modèles de comportement de chacun. Cependant, plus fréquemment, les agents doivent communiquer entre eux - ils doivent faire connaître leurs objectifs, leurs intentions, leurs résultats et leur état à d'autres agents. S'ils refusent de faire connaître leurs objectifs (partiellement ou totalement) ou leurs intentions, la situation sera clairement celle d'une concurrence obligeant les autres agents participants à postuler sur les objectifs des concurrents. La coordination a été utilisée pour résoudre plusieurs problèmes d'informatique distribuée et DAI, notamment :

- Cohérence du réseau : optimiser la collaboration entre un système d'agents répartis ;
- Répartition des tâches et des ressources entre les agents ;
- Reconnaître et résoudre les disparités ou les conflits d'objectifs, de faits, de croyances et de comportements d'agents.
- Détermination de la structure organisationnelle d'une configuration d'agent, c'est-à-dire définition des rôles, des responsabilités et des chaînes d'autorité entre agents. [34]

Bien que la liste ci-dessus ne soit pas exhaustive, elle donne une idée des types de problèmes résolus à l'aide de techniques de coordination.

La coordination ne concerne pas uniquement les agents de recherche. À mesure que les tendances évoluent vers les systèmes informatiques distribués, il est nécessaire de coordonner les éléments des systèmes distribués ouverts, analogues aux agents. En termes simples, si les agents doivent devenir le nouveau paradigme des systèmes informatiques, nous ne pouvons ignorer le besoin de recherche sur les techniques de coordination des agents.

2.3 Les techniques de coordination

Nwana, Lee et Jennings [34] ont classé les techniques de coordination dans grandes catégories :

1. Structuration organisationnelle
2. Coordination par protocoles
3. Planification multi-agents
4. Négociation
5. Formation de la coalition

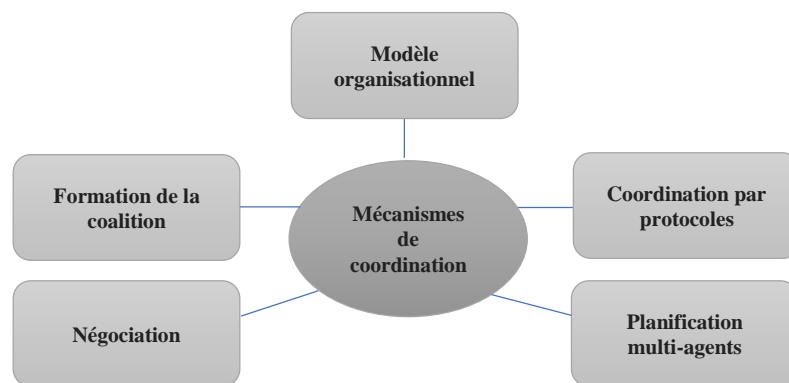


Figure 2-1 : Approches des mécanismes de coordination

Voici un aperçu et une brève critique de chaque catégorie :

2.3.1 *Modèle organisationnel*

Le moyen le plus simple d'assurer la coordination consiste à prédéfinir les responsabilités, les capacités, la connectivité et le flux de contrôle de chaque agent du système dans son ensemble. **Durfee et al** définissent cela comme les relations prédéfinies à long terme entre agents [38]. Les organisations hiérarchiques abondent, donnant lieu à la technique classique de coordination maître/esclave ou client/serveur, généralement utilisée pour l'attribution de tâches et de ressources parmi les agents esclaves par certains agents maîtres. Cette technique est implémentée de différentes manières :

- L'agent principal planifie et distribue des fragments du plan aux esclaves. Les esclaves peuvent ou non communiquer entre eux, mais doivent finalement communiquer leurs résultats à l'agent maître. Un exemple de ceci est un système de gestion de réseau, dans lequel des agents subordonnés résident sur chaque périphérique géré, rapportent les cas d'échec à un agent de gestion.

- La négociation Blackboard exploite l'architecture classique Blackboard [35] pour fournir une base de coordination. Dans ce schéma, les sources de connaissances du tableau sont remplacées par des agents qui affichent et lisent sur le tableau. L'agent de planification (ou l'agent principal) planifie les lectures / écritures de l'agent sur / à partir du tableau. Cette approche peut être utilisée lorsque le problème est distribué (diversité des techniques de résolution de problèmes), qu'un agent de planification central est présent, que des tâches ont déjà été attribuées, a priori, à des agents (indépendance de l'expertise), qu'il existe un langage d'interaction commun. il existe une génération de solution incrémentale [34],[35].

Durfee et al. [39] soulignent qu'un tel contrôle centralisé, comme dans la technique maître/esclave, est contraire aux hypothèses de base de DAI. Cela suppose qu'au moins un agent ait une vision globale de l'agence dans de nombreux domaines ; il s'agit là d'une hypothèse irréaliste.

2.3.2 Coordination par protocoles

Le protocole de contrat de réseau est une technique de coordination classique pour la répartition des tâches et des ressources entre les agents. Dans de nombreux domaines, certaines tâches sont intrinsèquement synergiques et doivent donc être gérées par un seul agent, tandis que d'autres sont mieux attribuées à différents agents. En effet, les agents ont des capacités et des ressources différentes et, par conséquent, des coûts différents pour la gestion des tâches. Cette approche de la répartition des tâches suppose une structure de marché avec un seul médiateur centralisé. Les agents peuvent assumer l'un des deux rôles suivants :

- Un médiateur chargé de diviser un problème en sous-tâches et de rechercher des sous-traitants pour le résoudre, ainsi que de surveiller la solution globale du problème.
- Un entrepreneur qui exécute une sous-tâche. Notez toutefois que les sous-traitants peuvent décomposer davantage la sous-tâche et la sous-traiter à d'autres agents, jouant ainsi le rôle de gestionnaires dans ce cas.

Le processus utilisé par les gestionnaires pour localiser les entrepreneurs est décrit ci-dessous :

- Un responsable annonce une tâche : elle prend généralement la forme d'un message diffusé.
- Les contractants évaluent la tâche en fonction de leurs capacités et leurs engagements mettent une offre sur la tâche s'il le souhaite.

- Le responsable évalue les offres reçues, choisit un contractant et lui attribue le contrat.
- Manager Le responsable attend le résultat du contrat.

Le protocole Contract-net classique supposait un système multi-agents d'agents coopératifs. Les améliorations récentes ont conduit à de meilleurs résultats viables parmi les agents qui s'intéressent aussi bien que chez les coopératifs [34].

Huhns et Singh [37] ont noté que le filet contractuel est mieux utilisé lorsque :

- la tâche d'application à une nature hiérarchique bien définie ;
- le problème à une décomposition à grains grossiers ;
- Le couplage entre les sous-tâches est minimal.

Les avantages du contrat net sont les suivants :

- attribution dynamique des tâches via l'auto-enchère, ce qui conduit à de meilleurs accords ;
- les agents peuvent être introduits et supprimés de manière dynamique, il fournit un équilibrage de charge naturel (car les agents occupés n'ont pas besoin de faire une offre)
- il s'agit d'un mécanisme fiable de contrôle distribué et de reprise sur incident [34].

La plupart des limitations initiales du protocole de contrat de réseau découlant de la supposition que les agents participants sont non antagonistes et bienveillants sont traitées par de nouvelles méthodes telles que les protocoles de négociation. Cependant, le protocole Contract-net utilise beaucoup de communication et nécessite une décomposition correcte des tâches. Cette approche a été utilisée dans de nombreuses applications [42],[43],[44],[54].

2.3.3 Planification multi-agents

Une autre approche de la coordination dans les systèmes à base d'agents consiste à faire participer les agents à la conception d'un plan qui guidera leur fonctionnement. Afin d'éviter des interactions et des actions conflictuelles, les agents élaborent un plan multi-agents détaillant toutes les actions futures nécessaires pour atteindre leurs objectifs, puis intercalent ces plans pour élaborer un plan de travail.

La planification multi-agents se divise en deux classes principales : la classe de planification multi-agents centralisée traditionnelle et la classe de planification multi-agents distribuée.

Dans la planification multi-agents centralisée, il existe généralement un agent de coordination, qui est chargé de recevoir des plans partiels ou locaux de tous les autres agents, d'analyser ces plans pour identifier les points de conflit et de les modifier, en les combinant dans un seul plan multi-agents. C'est exempt de conflits. Dans le plan final multi-agents, des commandes de communication sont insérées pour synchroniser les interactions des agents de manière appropriée [34]. Cammarata et al. [45] ont également utilisé la planification multi-agents centralisée dans un domaine de contrôle du trafic aérien simulé. Dans ce démonstrateur, deux agents (des avions) dans un scénario de conflit potentiel ont choisi l'un d'eux pour jouer le rôle d'agent de coordination. L'autre agent transmet ses plans à l'agent de coordination, qui modifie ensuite son propre plan pour éviter la collusion potentielle. Cette approche présente les mêmes inconvénients que la stratégie maître/esclave : le contrôle est central.

- Planification distribuée pour les plans centralisés : La formulation d'un plan complexe peut nécessiter la collaboration de divers spécialistes de la planification coopérative. tout comme générer la solution à tout problème complexe serait [35]. L'idée est de partager la tâche de former le plan et de partager les résultats des sous-tâches pour élaborer un plan global. Si l'un de ces planificateurs ne peut pas exécuter sa sous-tâche de planification avec le plan partiellement construit, il peut revenir en arrière et trouver d'autres choix. Une approche de partage de plan nécessiterait que les agents de planification génèrent des plans en parallèle, puis les partagent et les fusionnent pour converger vers un plan final. Cette façon de concevoir un plan nécessite toujours que la tâche de formation du plan soit correctement décomposée, ce qui peut ne pas toujours convenir à tous les domaines.
- Planification distribuée pour les plans distribués :

C'est la version la plus difficile de la planification distribuée : le processus de planification et les résultats sont destinés à être distribués [35]. L'idée est de fournir à chaque agent un modèle des plans d'autres agents. Les agents communiquent afin de construire et de mettre à jour leurs plans individuels et leurs modèles d'autrui jusqu'à ce que tous les conflits soient résolus [34]. L'exemple de cette approche est l'approche de planification globale partielle (PGP) de Durfee [46]. Le Framework PGP donne à un agent individuel la possibilité de :

1. représente ses propres activités prévues
2. communiquer sur ces plans avec d'autres.
3. modéliser les activités collectives de plusieurs agents.

4. proposer des modifications à une ou plusieurs activités globales pour améliorer les performances du groupe et modifier ses activités locales planifiées conformément à la proposition coordonnée. [35],[46].

En général, la planification multi-agents distribuée est plus complexe que la forme centralisée de planification : elle nécessite des ressources informatiques et de communication complexes. En outre, certaines techniques de planification, telles que le PGP, sont progressives et leur champ d'application peut donc être limité.

2.3.4 Négociation

La négociation est une partie importante du travail de coordination. Il existe une littérature abondante sur le sujet, car la négociation est une technique de coordination essentielle utilisée pour résoudre plusieurs problèmes de DAI. Une définition de base de la négociation est donnée par **Bussman** et **Muller** [47] :

“... La négociation est le processus de communication d'un groupe d'agents afin de parvenir à un accord mutuellement accepté sur certaines questions.” Sycara ajoute que pour négocier efficacement, les agents doivent raisonner sur les convictions, les désirs et les intentions d'un autre agent [48], ce qui a conduit à la mise au point de techniques pour :

- représenter et maintenir des modèles de croyance ;
- raisonnement sur les croyances des autres agents ;
- influencer les intentions et les convictions des autres agents.

Ces derniers sujets ont conduit à l'utilisation de toutes sortes d'intelligence artificielle et de techniques mathématiques, y compris la théorie des jeux, la logique, la révision des croyances, le maintien de la vérité, etc. La négociation est utilisée en tant que technique dans toutes les autres approches déjà évoquées. Cette vue d'ensemble n'est pas exhaustive. Cependant, il fournit une bonne base pour comprendre les techniques de négociation. Les techniques de négociation peuvent être classées dans les grandes catégories suivantes :

- négociation basée sur la théorie des jeux ;
- négociation basée sur un plan
- Diverses approches de négociation.

2.3.4.1 Négociation basée sur la théorie des jeux

Les développeurs de mécanismes de négociation centrés sur les agents se concentrent sur le problème suivant : «Dans un environnement dans lequel mon agent doit opérer, quelle

est la meilleure stratégie à suivre?» [35]. Lorsque les agents interagissent, il est nécessaire de réfléchir aux préférences et aux stratégies de l'autre agent. Ainsi, chaque fois que l'on est confronté à une situation dans laquelle la prise de décision rationnelle d'un agent dépend de ses attentes sur ce qu'un ou plusieurs autres agents vont faire, et la leur dépend également de ce qu'il va faire, la théorie des jeux devient très utile pour l'analyste [49]. Les concepts clés d'une approche basée sur la théorie des jeux incluent des fonctions utilitaires, un espace de transactions, des stratégies et des protocoles de négociation.

Un accord est un plan commun entre agents qui répondrait à tous leurs objectifs. Chaque transaction avait une valeur d'utilité. L'utilité est définie comme la différence entre la valeur de la réalisation d'un objectif et le prix payé pour l'atteindre. Chaque agent veut maximiser sa propre utilité. Les agents discutent d'un ensemble de négociations, qui est l'ensemble de toutes les offres qui ont une utilité positive pour chaque agent. Il existe une hypothèse implicite selon laquelle chaque agent de la négociation est un élargissement d'utilité attendu.

Étant donné que les agents ne sont pas tenus de rester véridiques à tout moment, le processus de négociation peut être considéré comme un processus en deux étapes : la négociation proprement dite et l'exécution des plans conjoints [50]. Les techniques de la théorie des jeux doivent également prendre en compte le temps au cours du processus de négociation, car le temps influence le résultat des négociations.

L'approche de la théorie des jeux doit encore être élargie pour intégrer des situations de connaissance partielle ou incomplète, lorsque les agents n'ont pas une connaissance complète des préférences de leur adversaire ou de la matrice des gains. Une grande partie du travail suppose que deux agents négocient, bien que des recherches soient en cours sur les négociations entre agents [1 *]. À mesure que de plus en plus d'agents s'impliquent dans le processus de négociation, la matrice de gains devient plus large et peut devenir intraitable et incomplète. Il est difficile de voir comment la technique de la théorie des jeux peut être utilisée de manière fiable dans des applications réelles sans explorer les techniques de résolution d'incertitude pour les systèmes multi-agents [49].

2.3.4.2 Négociation basée sur un plan

Dans la section sur la planification distribuée, nous avons considéré les agents ayant des plans locaux différents et devant coordonner leurs plans pour éviter les conflits dans le système. La décision de quel agent doit en attendre un autre est assez aléatoire et arbitraire dans certains cas, mais ne doit pas nécessairement l'être. Une grande quantité de travail en

négociation concerne la résolution de ce type de conflits. Parfois, par exemple, la décision de l'agent qui doit réviser ses plans locaux peut être basée sur l'espace de possibilités alternatives ouvert à cet agent. Cela indique le niveau de flexibilité de l'agent. Il existe plusieurs autres plans de négociation ouverts aux agents, mais presque toutes ces techniques supposent que les agents sont honnêtes quant à l'importance de leurs objectifs et de leur degré de flexibilité.

2.3.4.3 Approches de négociation diverses

Comme les interactions humaines semblent toujours nécessiter une forme de négociation, de nombreux chercheurs s'inspirent des stratégies de négociation humaine.

Un des documents les plus utiles sur la négociation provient du cadre de négociation de **Bussmann** et **Muller** pour les agents coopérants [47]. S'appuyant sur les théories socio-psychologiques de la négociation, ils développent un modèle de négociation cyclique à la fois général et simple. La stratégie générale est que la négociation commence avec un ou plusieurs agents faisant une proposition. D'autres agents évaluent la proposition et dressent une liste de leurs préférences qui ont été violées par la proposition. Tout le monde met à jour ses connaissances sur les préférences des autres et le cycle de négociation reprend.

Il existe plusieurs autres techniques de négociation similaires à celles de Bussmann et Muller qui utilisent la stratégie de tableau pour la communication et le partage des connaissances, par exemple. Werkman propose un modèle basé sur la connaissance d'une forme de négociation progressive [40]. Le modèle DFI (Designer Fabricator Interpreter) de Werkman est basé sur divers modèles de négociation humains.

2.3.5 Formation de la coalition

Le concept à est fréquemment utilisé pour décrire les systèmes organisationnels. A pourrait être défini comme un agent impliqué dans la communauté qui prend une décision en utilisant des protocoles de négociation pour coopérer les uns avec les autres dans le but d'atteindre un objectif spécifique [53]. Amélioration de ce SMA non seulement focalisé recherche aspects théoriques coopérative Il permet également aux organisations virtuelles des agents de création via des stratégies coordonnées d'application, avec leur logique temporaire d'être dépendante des conditions de changements dynamiques. Les principaux avantages sont résumés comme suit :

- Le développement des capacités individuelles de résolution de tâches des agents et l'amélioration de leur qualité.
- Réduire la complexité de la coordination et de l'organisation dans MAS et simplifier le processus de communication lors du passage des interactions locales (entre les agents) aux interactions globales (entre les coalitions).
- Structuration du domaine de connaissance, chaque sous-domaine peut être différencié par une coalition spécifique. Ainsi, les liens entre les différentes coalitions reflètent les conséquences du même modèle décrivant le domaine de la connaissance.
- Réorganisation dynamique du SMA, lorsque les agents doivent réagir rapidement aux modifications de l'environnement dans le but de reconfigurer leur comportement et de s'adapter à de nouvelles conditions, y compris la nécessité d'agir dans des scénarios imprévus ou émergents, en utilisant efficacement leurs capacités et leurs capacités de communication.

Les recherches sur la formation de coalitions tentent d'optimiser la formation de coalition, ce qui nécessite des algorithmes extrêmement complexes. Dans [53], ils ont proposé une solution basée sur la coalition d'agents au problème de l'optimisation d'un transport modal afin d'établir une solution d'itinéraire optimisée pour les voyageurs par rapport à leurs critères. Précisément, la coalition des segments de route pour composer un itinéraire complet ainsi que la combinaison des différents itinéraires représentés par les agents de route sont abordés.

S'agissant des réseaux de distribution d'énergie [54], ils ont proposé un mécanisme décentralisé de formation de coalitions multi-agents permettant aux agents de nouer de manière autonome des partenariats via des négociations en vue de résoudre le problème de la répartition de l'énergie.

En ce qui concerne l'apprentissage à travers la formation de partenariat, un nouveau modèle bayésien basé sur l'apprentissage par renforcement est présenté dans [60]. Ce modèle permet aux agents d'améliorer leurs convictions quant aux types d'autres agents appartenant à la même coalition. De plus, cela permet aux agents de faire des compromis explicites dans le but de former de nouvelles coalitions et d'en savoir plus sur les types d'autres partenaires potentiels.

En ce qui concerne l'application de la formation de coalition dans les jeux PE, les auteurs dans [61] ont proposé de former un algorithme de formation de coalition des poursuivants afin d'améliorer le calcul en temps réel de la coalition optimale. Notant qu'une

coalition de poursuites est créée afin de capturer un évadé détecté d. Plus précisément, cet algorithme est basé sur des gains de cupidité optimaux permettant l'affectation des poursuivants aux différentes coalitions créées.

2.4 Critères pour choisir une technique de coordination

Après avoir examiné toutes les techniques ci-dessus, même s'il ne s'agit pas d'une liste exhaustive, il est utile d'énumérer certains critères à prendre en compte avant d'adopter une technique pour tout développement de système multi-agents. La liste ci-dessous peut être étendue et rendue plus complète :

- **Le nombre d'agents**

En ignorant tous les autres facteurs, tels que l'hétérogénéité et la distributivité dans l'environnement, nous constatons que plus le nombre d'agents dans le système augmente, plus la communication entre ces agents devient un problème central, que le contrôle soit centralisé ou non. Si chaque agent interagit avec tous les autres agents, le nombre d'interactions appariées augmentera de manière quadratique avec le nombre d'agents et la complexité du problème de coordination augmentera de manière exponentielle : si chaque agent peut choisir parmi b actions, chacune ayant potentiellement des impacts différents sur l'autre agent, l'espace des combinaisons possibles serait « b » pour n agents [36]. Aucune stratégie ne résoudra le problème de la quantité, mais certaines stratégies seront plus efficaces que d'autres. Par exemple, au lieu de demander à tous les agents de demander de l'aide à tous les autres agents lorsqu'ils en ont besoin, un agent intermédiaire disposant d'informations sur les services offerts par tous les agents du domaine peut facilement acheminer une demande à l'agent approprié.

- **Les classes d'agents**

Par classes d'agents, « les types d'agents ». En plus d'avoir des objectifs, des croyances et une expertise différents, les agents peuvent également utiliser différents langages de communication, ontologies ou architectures internes. Il est important de prendre en compte l'hétérogénéité du système lors du choix d'une technique de coordination. La stratégie Blackboard, par exemple, ne suffira pas dans un environnement où les agents en interaction ont des langages de communication et des ontologies différents.

- **La polyvalence de chaque agent**

La polyvalence se réfère ici à la capacité des agents à décider de manière flexible des objectifs à poursuivre à un moment donné et de la manière de les atteindre. Les agents spécialisés chargés de la coordination qui s'acquittent d'une tâche donnée consacrent moins de ressources

à la stratégie de coordination. Pour gérer les problèmes de polyvalence, la technique doit être capable de gérer la coordination des plans d'exécution. L'une de ces techniques est la technique PGP [46].

- **L'environnement de la tâche - distributive ou non**

Si les agents sont regroupés de manière à ce que les tâches proviennent d'un point et soient coordonnées à partir de ce point, la technique de coordination est centrale. Dans d'autres environnements de tâches, lorsque les agents sont fortement distribués et que les tâches sont réparties de manière inhérente entre eux, la technique de coordination est contrainte par des problèmes d'incertitude, tels que - qui fait ou devrait faire quoi [36].

- **L'environnement de la tâche - Présence d'incertitude**

L'incertitude persiste à bien des égards dans tous les systèmes multi-agents : il peut y avoir une incertitude dans les données et les capteurs reçus par les agents, la matrice des gains d'un agent dans un environnement basé sur la théorie des jeux peut être incomplète, les agents auto-testés ne nécessairement être honnête, etc. La stratégie de coordination devrait prévoir, implicitement ou explicitement, l'éventail des conditions dans lesquelles la solution fournie sera suivie. Durfee [36] a déclaré à juste titre que «dans un environnement de tâches dans lequel des déviations mineures peuvent avoir des conséquences graves, il peut être impératif de trouver des solutions solides».

- **Limites de frais généraux**

Certaines stratégies de coordination sont plus coûteuses que d'autres. Le coût ici inclut les exigences de calcul, la surcharge de communication, le temps passé, etc. La rentabilité entre le coût et la qualité est une décision à prendre par le développeur d'un système multi-agents. Par exemple, pour prendre des décisions de coordination de haute qualité, les échanges d'informations entre agents seront très sollicités, sans quoi des décisions seront prises sans informations complètes ou exactes.

2.5 Organisations dans les systèmes multi-agents

Dans les systèmes multi-agents, l'interaction est contrôlée par un type d'organisation implicite ou explicite. Ces interactions contrôlent généralement certains comportements du système, tels que les structures d'autorité, les relations entre exigences et les flux de données. **Horling** et **Lesser** [55] définissent l'organisation d'un système multi-agents comme suit : l'ensemble des rôles, relations et structures d'autorité qui régissent son comportement. La conception organisationnelle permet de contrôler la sélection et le déplacement entre différents ensembles d'agents, de formes architecturales et de moyens pour les tâches. Avoir une structure organisationnelle peut être utile pour réduire le nombre d'interactions entre les agents, diminuer ou augmenter la redondance du système, gérer les incertitudes du système et former des objectifs globaux que les agents individuels peuvent être ignorés [52]. Sinon, une structure organisationnelle peut réduire la flexibilité globale du système, influencer négativement le temps de calcul ou de communication, et accentuer la complexité du système [56] [57].

L'organisation du système affecte ses performances à court et à long terme [51] [52] [57] [85] [56] [59] [62]. L'application de différents modèles organisationnels à un même problème conduira à plusieurs caractéristiques de performance [55]. Toute méthode pouvant être utilisée pour comparer différents modèles organisationnels doit comprendre leurs comportements et réorganiser le système selon la configuration appropriée sera bénéfique.

Il existe de nombreux paradigmes organisationnels adaptés aux systèmes multi-agents, tels que : marchés, sociétés, hiérarchies, coalitions, équipes, congrégations, organisations composites, fédérations et sociétés. Chacune de ces structures a ses propres caractéristiques, avantages et limites, telles que le soutien individuel en relation avec la rationalité du groupe, le comportement myope (un agent ne prend pas en compte les besoins critiques des autres agents, tandis que d'autres ressources peuvent toujours être satisfaisantes pour l'agent), de structure à long terme ou à court terme, et convenant aux populations d'agents homogènes ou hétérogènes. Il est généralement admis qu'il n'existe pas une seule structure organisationnelle qui conviendrait pour diverses applications et domaines [52] [55]. Il peut même être essentiel d'appliquer une combinaison de différentes structures au sein d'un même système. Les avantages et les inconvénients de ces structures organisationnelles sont décrits dans une enquête réalisée par **Horling** et **Lesser**.

2.6 Organisation et modèle organisationnel

Comme le soulignent *J. Ferber* [1] et *Y. Demazeau et al.* [63] [64], il est important de distinguer un modèle d'organisation de son implantation. Cette distinction met l'accent sur le fait que les propriétés d'un modèle organisationnel, et donc du système, dépendent essentiellement de la façon dont il est implanté. Pour qu'un modèle organisationnel fonctionne correctement, il faut par exemple que chaque rôle soit attribué à un agent au moins (capable de remplir ce rôle). En outre, pour qu'elle exhibe des propriétés spécifiques (ex., un certain degré de robustesse), l'implantation doit en plus respecter des contraintes qui ne sont pas forcément dictées par le modèle organisationnel (ex., chaque rôle est attribué à plusieurs agents).

2.7 Modèles organisationnels

Modèle organisationnel se base sur les concepts de plus haut niveau d'abstraction tel que les groupes, les rôles, les protocoles d'interactions, les responsabilités, les permissions...etc. Elle peut être vue comme une technique qui sert à contraindre les comportements individuels des agents vers les objectifs globaux à satisfaire. On part alors d'une organisation conçue par les développeurs du SMA pour arriver aux comportements individuels de chaque agent.

La formation d'organisations peut s'effectuer de différentes manières. En théorie des organisations, une organisation peut être spontanée ; émergeant via les interactions d'un ensemble de décideurs individuels, ou il pourrait s'agir d'une structure prédéfinie dans laquelle les membres devraient essayer de s'intégrer. Dans les systèmes multi-agents, ces deux tendances existent également. Dans certains cas, la structure d'une organisation est décrite comme un ensemble de règles qui répondent aux questions suivantes : quoi, qui et comment organiser la communication. Dans d'autres cas, l'intérêt porte sur la formation émergente d'une organisation dans laquelle son objectif est d'intégrer la structure organisationnelle aux agents. Une classification générale des méthodes disponibles de formation d'organisation dans les systèmes multi-agents suit [55] :

- Scripté : certaines instructions statistiques prédéfinies déterminent la formation de l'organisation.
- Contrôlé : une organisation est explicitement projetée sur une population par un individu ou un groupe d'individus.

- Emergent : Dans cette technique, il n'existe aucune orientation centrale ou globale. L'organisation apparaît via les actions des agents individuels. Ces méthodes sont considérées comme auto-dirigées.

Il n'est pas toujours possible de définir une classe pour un système dans des applications pratiques. Un mélange combinatoire des méthodes ci-dessus est très souvent introduit. Les chercheurs se concentrent davantage sur les structures émergentes dans la modélisation de systèmes complexes intégrés dans des environnements dynamiques et incertains. L'inconvénient des approches émergentes ou ascendantes de l'auto-organisation est celui d'une qualité inférieure à celle d'une organisation soigneusement modelée.

On va présenter dans la partie qui suit quelques modèles qui s'inscrivent dans ce cadre organisationnel.

2.7.1 AALAADIN (AGR) [GUT, 2001]

2.7.1.1 Présentation du modèle

Le modèle AGR (Agent, Groupe, Rôle) [66], est l'évolution du modèle AALAADIN (Figure 2-2) qui trouve son origine dans une réflexion sur les différents écueils des modèles centrés agents.

AALAADIN [66] est sans doute la plus connue des approches organisationnelles comportementalistes des SMA. Menée par O. *Gutknecht* et *J.Ferber*, cette démarche s'articule selon quatre axes :

- 1) Un axe conceptuel présentant un modèle s'articulant autour des trois concepts d'agent, de groupe et de rôle.
- 2) Un axe proposant une sémantique formelle au modèle Agent-Groupe-Rôle.
- 3) Un axe d'implémentation fournissant la plate-forme MadKit.
- 4) Un axe étudiant des pistes pour une méthodologie de conception des SMA organisationnels.

Le modèle est basé sur l'association de trois concepts clés : l'agent, le groupe et le rôle, utilisés simultanément pour décrire des organisations concrètes d'agents. La Figure 2-2 présente un diagramme UML de ce modèle de base.

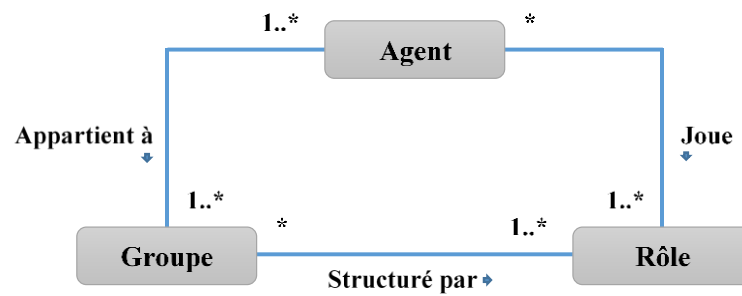


Figure 2-2 : Agent – Group - Rôle méta modèle [66]

Sur [65] se fait le commentaire suivant : « Un agent peut intervenir dans plusieurs communautés (que nous appellerons 'groupes' dans notre modèle) en parallèle. Il peut jouer dans chacun de ces groupes un ou plusieurs rôles correspondant à ses activités ou interactions. Ces rôles peuvent être portés par un nombre d'agents arbitraire, dépendant de la situation et des normes de l'organisation. »

2.7.1.2 Agent

Dans ce modèle, un agent est une entité capable d'agir et de communiquer, qui peut jouer un ou plusieurs rôles dans un ou plusieurs groupes. Il n'existe aucune contrainte sur la structure interne de l'agent.

2.7.1.3 Groupe

Un groupe est un ensemble d'agents qui partagent des caractéristiques. Il est utilisé pour diviser l'organisation en groupant des agents dont l'activité se rejoint. Deux agents peuvent communiquer uniquement s'ils appartiennent au même groupe.

2.7.1.4 Rôle

Un rôle est la représentation abstraite de la fonction d'un agent dans un groupe. Un rôle est local à un groupe et n'a plus aucun sens en dehors de ce contexte. Pour jouer un rôle dans le groupe, l'agent doit postuler pour ce rôle, et l'affectation se fait en fonction de ses compétences. Ces rôles peuvent être portés par un nombre d'agents arbitraire, dépendant de la situation et des normes de l'organisation.

Dans AGR, la présentation de l'organisation est basée sur un méta-modèle qui clarifie une relation structurelle entre des collections d'agents. Ce modèle est défini à l'aide d'agents, de groupes et de rôles (AGR). Dans ce modèle, un agent peut appartenir à plusieurs groupes. Un agent aura un rôle précis dans chaque groupe. La création de groupes est effectuée par agent, et l'agent qui crée un groupe joue le rôle de gestionnaire de groupe.

Chaque groupe est organisé au moyen de deux composants principaux : un tuple qui regroupe tous les rôles possibles et un graphe d'interaction qui clarifie les interactions entre les rôles du groupe. La vision minimaliste basée sur la structure des organisations d'agents de ce modèle en est la caractéristique principale. La structure de l'organisation est constituée de l'ensemble des groupes et des interactions possibles entre les rôles appartenant à différents groupes. Ce modèle procure une présentation rationnelle pour une organisation à plusieurs groupes dans laquelle la structure organisationnelle décrit l'interaction entre les membres.

2.7.2 Le modèle AGRMF (Agent-Group-Role-Membership Function)

Dans notre extension, nous nous intéressons au changement de vision des groupes, en fournissant des conseils méthodologiques qui permettent leur flexibilité au sein d'un système multi-agents. Étant donné que dans AGR, il n'existe aucun mécanisme définissant l'accès à un groupe spécifique. À cette fin, nous avons introduit certains principes de la logique floue. Nous devons imaginer nos groupes comme un ensemble flou que nos agents pourraient intégrer à un degré spécifique de membres.

2.7.2.1 Agent

Dans ce modèle, l'agent est une entité capable d'agir et de communiquer :

- L'identifiant de l'agent.
- Attributs de l'agent interne.
- Coefficients des attributs de chaque agent interne

2.7.2.2 Group

Le groupe est équipé d'une fonction d'appartenance qui gère sa flexibilité (accès de l'agent au groupe). Il représente également la notion primitive de combinaison d'agents. Chaque agent peut être membre d'un ou de plusieurs groupes. Dans sa forme la plus simple, un groupe peut être considéré comme un moyen d'identification en regroupant un ensemble d'agents.

- Gr: <ID, Rol, Atg, Mf>
- ID : l'identifiant du groupe. Chaque groupe a un identifiant spécifique.
- Rol : Tous les rôles proposés par le groupe à un moment donné.
- Atg: attributs de groupe internes.
- Mf: La fonction d'appartenance définit le degré d'appartenance d'un agent du groupe à un moment donné.

2.7.2.3 *Le rôle*

La différence entre le rôle dans le modèle AGRMF et le rôle dans le modèle AGR est qu'il ne peut être joué par aucun agent. Pour jouer ce rôle, l'agent doit appartenir à ce groupe avec un degré de membre spécifique, ce qui améliorera automatiquement la résolution de la tâche.

- **Ro**: <Gr, Nm, Md, Atr>
- **Gr** : Le groupe Id qui contient le rôle.
- **Nm** : le nom du rôle.
- **Md** : définit le degré minimum de membre de l'agent requis pour pouvoir être capable de jouer le rôle.
- **Atr** : attribut de rôle interne.

2.7.2.4 *La fonction d'adhésion (The membership function)*

Au lieu d'appartenir à l'un des ensembles «vrai» ou «faux» dans la logique binaire traditionnelle, la logique floue admet des degrés d'appartenance à un ensemble donné. Le degré d'appartenance à un ensemble flou est matérialisé par un nombre compris entre 0 et 1. Une valeur spécifique de la fonction d'appartenance liée à une valeur de la variable est notée μ et appelée «facteur d'appartenance». Notre fonction d'appartenance représente le mécanisme d'accès pour les différents groupes. Il faut des paramètres comme indicateurs de capacité principaux d'un agent et il est ensuite nécessaire de vérifier si l'agent est capable de jouer ce rôle (c'est-à-dire de rejoindre un groupe). Un agent peut jouer un rôle s'il appartient au groupe avec un degré d'appartenance spécifique. En plus de la distance qui sépare le poursuivant de l'objectif et de la vente liée, d'autres facteurs d'aptitude concernant les agents jouant le rôle de poursuivant seront ajoutés.

2.7.3 *Le modèle YAMAM (YET ANOTHER MULTI-AGENT MODEL)*

Le modèle organisationnel YAMAM [83] est une alternative aux autres modèles basés sur des agents tels qu'AGR [84] et AGRMF [85]. Les principaux avantages de ce modèle se résument à sa bonne modularité et à son éventuelle évolutivité. En outre, il peut être interfacé avec des outils de planification de tâches, comme il sera utilisé dans ce document. YAMAM est basé sur quatre concepts différents : agent, compétence, rôle et tâche. Dans ce modèle, l'organisation est décrite par sa structure inhérente. Par conséquent, les relations entre les agents sont primordiales par rapport aux agents et à leur comportement.

2.7.3.1 Agent

Dans cette structure organisationnelle, un agent est défini comme une entité autonome et communicante située dans un environnement explicite ou implicite. En fait, l'agent est construit sur des propriétés réactives, ce qui implique que les compétences ne peuvent pas être ajoutées de manière dynamique. Cependant, il est possible d'instancier un agent cognitif capable d'utiliser des compétences réactives et cognitives afin d'évaluer au fil du temps.

2.7.3.2 Rôle

Le rôle reflète un formulaire d'identification de service ou d'agent. L'agent peut gérer avec un ou plusieurs rôles concernant un environnement spécifique. De plus, le rôle est basé sur un ensemble de tâches à exécuter de différentes manières. On suppose qu'un agent ne peut jouer un rôle que s'il est capable d'exécuter les tâches impliquées, et donc s'il possède les compétences requises. Généralement, le rôle nécessite une capacité d'exécution de plusieurs tâches. Par conséquent, un agent doit être caractérisé par les compétences liées aux tâches concernées. Dans cet article, ce concept est utilisé pour attribuer le rôle "poursuivant" qui diffère d'un groupe de poursuites à un autre. De plus, nous notons que chaque agent ne peut pas jouer le rôle de "poursuivant" dans plus d'un groupe de poursuites.

2.7.3.3 Compétence (Skill)

La compétence est une unité de connaissance nécessaire au traitement d'une tâche donnée. Il existe deux types de compétences : les compétences cognitives, qui ne peuvent être reçues que par les agents cognitifs, et les compétences réactives, qui peuvent également être reçues par les agents réactifs. Un agent peut regrouper plusieurs compétences afin d'exécuter l'ensemble des tâches attendues requises pour jouer un rôle spécifique. Dans cet article, les compétences du poursuivant sont représentées par le degré de traitement de la tâche (Ω), la distance qui le sépare de la cible (Dist) et la connaissance de la tâche (Liaison).

2.7.3.4 Tâche (Task)

Pour être réalisée, la tâche peut être considérée comme l'opération d'une compétence ou comme une action nécessitant une ou plusieurs compétences. Dans le jeu poursuite-évasion, les différentes poursuites représentent les tâches à exécuter. En outre, chaque évadé nécessite la capture d'un nombre et d'un type de poursuivants spécifiques (compétences). La Figure 2-3 détaille les différentes relations entre les concepts constituant le modèle organisationnel YAMAM. Cependant, contrairement au modèle organisationnel AGR, le groupe de concepts n'est pas implémenté physiquement dans ce modèle. En d'autres termes, les agents n'ont aucune

vision des groupes existants et ignorent totalement leurs informations d'adhésion. Dans notre proposition, les groupes seront représentés par les différentes formations de coalition formées pour capturer les évadés détectés.

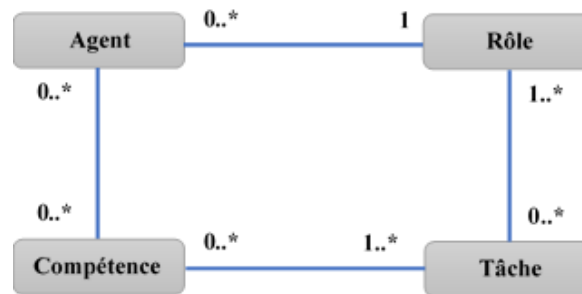


Figure 2-3 : YAMAM méta modèle

2.8 Théorie des jeux dans les systèmes multi-agents

La théorie des jeux est une branche de l'économie qui étudie les interactions entre des agents auto-différenciés. Tout comme la théorie de la décision, avec laquelle elle partage de nombreux concepts, la théorie des jeux trouve ses racines dans les travaux de *Von Neumann et Morgenstern* [68]. Comme son nom l'indique, les concepts de base de la théorie des jeux sont issus de l'étude de jeux tels que les échecs et les dames. Cependant, il est rapidement apparu que les techniques et les résultats de la théorie des jeux pouvaient également être appliqués à toutes les interactions entre agents indépendants.

La question classique de la théorie des jeux posée lors de chaque rencontre multi-agents est la suivante : quelle est la meilleure chose - la plus rationnelle - qu'un agent puisse faire ? Dans la plupart des rencontres multi-agents, le résultat global dépendra de manière décisive des choix faits par tous les agents du scénario. Cela implique que pour qu'un agent puisse faire le choix qui optimise ses résultats, il doit raisonner de manière stratégique. En d'autres termes, il doit prendre en compte les décisions que l'autre agent peut prendre et supposer qu'il agira de manière à optimiser ses propres résultats. La théorie des jeux nous donne un moyen de formaliser et d'analyser de telles préoccupations.

Aux débuts de la recherche sur les systèmes multi-agents, il était généralement admis que les agents étaient bienveillants : en d'autres termes, les agents pouvaient être supposés partager un objectif commun et seraient donc heureux de "donner un coup de main" à chaque demande. L'accent était mis sur les systèmes distribués de résolution de problèmes, dans lesquels des groupes d'agents bienveillants travaillaient ensemble pour résoudre des problèmes d'intérêt commun [71]. Il semblait exister une hypothèse implicite selon laquelle cette classe

de systèmes était la plus courante et que les scénarios dans lesquels les agents sont en concurrence étaient au mieux inhabituels, les aberrations au pire. Au fil du temps, on a fini par reconnaître que la bienveillance était l'exception ; l'intérêt personnel est la norme. La reconnaissance de ce fait semble avoir été motivée, du moins en partie, par la croissance rapide d'Internet et la tendance constante à la généralisation croissante des systèmes informatiques.

Parallèlement à cette reconnaissance croissante du fait que les agents d'intérêt personnel sont la norme, on observe une croissance croissante de l'intérêt pour les applications de la théorie des jeux aux systèmes multi-agents. La théorie des jeux est entrée dans la littérature sur les systèmes multi-agents en grande partie grâce aux travaux de Jeffrey Rosenschein et de ses collègues [72], [73], [74]. Dans sa thèse de doctorat de 1985 [68], Rosenschein a utilisé des techniques de la théorie des jeux pour analyser une gamme de scénarios d'interaction multi-agents.

Par exemple, il a montré comment certains types de coopération et de conclusion d'opérations pourraient se dérouler sans communication : les deux agents calculent simplement le meilleur résultat et savent que la partie avec laquelle ils traitent fera de même. Étant donné que les agents peuvent utiliser des techniques de la théorie des jeux pour prédire ce que les autres vont faire, cela évite le besoin de communication explicite - la coordination découle de l'hypothèse d'une rationalité mutuelle [73].

Les applications les plus convaincantes de la théorie des jeux pour les systèmes multi-agents ont sans doute été dans le domaine de la négociation [67],[74],[75]. En termes simples, la négociation est le processus par lequel les agents peuvent parvenir à un accord sur des questions d'intérêt commun. La négociation et la négociation ont été étudiées dans la littérature sur la théorie des jeux bien avant l'émergence des systèmes multi-agents en tant que discipline de recherche et même avant l'avènement du premier ordinateur numérique. Cependant, l'informatique apporte deux considérations importantes à l'étude théorique de la négociation des négociations :

1. Les études de la théorie des jeux sur le choix rationnel lors de rencontres multi-agents supposaient généralement que les agents étaient autorisés à choisir la meilleure stratégie parmi toutes les stratégies possibles, en prenant en compte toutes les interactions possibles. Il s'avère que «l'espace de recherche» des stratégies et des interactions à prendre en compte connaît une croissance exponentielle, ce qui signifie que le problème de la recherche d'une stratégie optimale est en général insoluble en calcul. En informatique, l'étude de tels problèmes relève de la théorie de la complexité

computationnelle [70]. Une littérature importante est consacrée au développement d'algorithmes efficaces (temps polynomiaux) pour des problèmes apparemment insolubles, et l'application de telles techniques à l'étude de rencontres multi-agents est un domaine de travail continu et fructueux.

2. L'émergence d'Internet et du World Wide Web a créé un énorme impératif commercial pour le développement ultérieur des techniques de négociation et de négociation informatiques [69].

Dans un scénario de négociation particulier impliquant des agents automatisés, les techniques de la théorie des jeux peuvent être appliquées à deux problèmes clés :

- 1- La conception d'un protocole approprié qui régira les interactions entre les participants à la négociation. Le protocole définit les «règles de rencontre» entre agents [74]. Formellement, un protocole peut être compris comme une fonction qui, sur la base d'un historique de négociation préalable, définit les propositions autorisées par les participants à la négociation. Il est possible de concevoir des protocoles de manière à ce que tout historique de négociation ait certaines propriétés souhaitables - il s'agit de la conception de mécanismes, qui est traitée plus en détail ci-dessous.
- 2- La conception d'une stratégie particulière que chaque agent peut utiliser lors de la négociation - un agent visera à utiliser une stratégie qui optimise son bien-être individuel. Une difficulté majeure réside dans le fait qu'en règle générale, les stratégies qui fonctionnent le mieux en théorie ont tendance à être intraitables sur le plan des calculs et sont donc inutilisables en pratique. Comme indiqué ci-dessus, la conception de mécanisme implique la conception de protocoles pour régir les interactions multi-agents, de sorte que ces protocoles présentent certaines propriétés souhaitables. Les propriétés possibles incluent, par exemple [75, p204] :
 - Succès garanti. Un protocole garantit le succès s'il garantit qu'un accord sera finalement atteint.
 - Maximiser le bien-être social. Intuitivement, un protocole maximise le bien-être social s'il garantit que tout résultat maximise la somme des utilités des participants à la négociation. Si l'utilité d'un résultat pour un agent était simplement définie en termes de montant d'argent reçu par l'agent, le protocole qui maximiserait le bien-être social maximiserait alors le montant total d'argent «versé».
 - Efficacité de Pareto. Un résultat de négociation est dit efficace selon Pareto s'il n'y a pas d'autre résultat qui améliorera au moins un agent sans aggraver au moins un autre

agent. Intuitivement, si un résultat de négociation n'est pas efficace selon Pareto, il existe un autre résultat qui rendra au moins un agent plus heureux tout en rendant tous les autres au moins aussi heureux.

- **Rationalité individuelle.** Un protocole est dit individuellement rationnel si le fait de respecter les règles est dans l'intérêt des participants aux négociations. Les protocoles individuellement rationnels sont essentiels car sans eux, rien n'inciterait les agents à engager des négociations.
- **Stabilité.** Un protocole est stable s'il incite tous les agents à se comporter d'une manière particulière. Le type de stabilité le plus connu est l'équilibre de Nash : deux stratégies s et s' sont dites à l'équilibre de Nash si, en supposant qu'un agent utilise s , l'autre ne peut pas faire mieux que l'utilisation s' , et inversement.
- **Simplicité.** Un protocole «simple» est un protocole qui rend «évidente» la stratégie appropriée pour un participant à la négociation. En d'autres termes, un protocole est simple s'il est utilisé, un participant peut facilement déterminer la stratégie optimale.
- **Distribution.** Un protocole devrait idéalement être conçu pour garantir qu'il n'y a pas de «point de défaillance unique», et idéalement, afin de minimiser la communication entre les agents.

Le fait que même des protocoles de négociation assez simples puissent avoir des propriétés aussi souhaitables qu'elles expliquent en grande partie le succès des techniques de la théorie des jeux pour la négociation [67].

Malgré ces avantages très évidents, l'utilisation de la théorie des jeux dans les problèmes de négociation pose un certain nombre de problèmes :

La théorie des jeux suppose qu'il est possible de caractériser les préférences d'un agent en fonction des résultats possibles. Les humains, cependant, ont beaucoup de mal à définir systématiquement leurs préférences par rapport aux résultats en général. Ces préférences ne peuvent même pas être caractérisées par un simple classement par ordre de résultats, encore moins par les utilités numériques.

La plupart des techniques de négociation basées sur la théorie des jeux tendent à supposer la disponibilité de ressources informatiques illimitées pour trouver une solution optimale - elles présentent les caractéristiques des problèmes NP-difficiles. (Un exemple bien connu est le problème de la détermination du gagnant dans les enchères) Dans de tels cas, une approximation des solutions théorie des jeux peut être plus appropriée.

Malgré ces problèmes, la théorie des jeux est extrêmement convaincante en tant qu'outil de négociation automatisée. Dans les cas où il est possible de caractériser les préférences et les stratégies possibles des participants à la négociation, la théorie des jeux a beaucoup à offrir.

2.9 La coordination avec l'application de la théorie des jeux

L'utilisation initiale de la théorie des jeux mathématique dans la conception d'un mécanisme de coordination multi-agents revient à Rosenschein [76]. Le chercheur a utilisé des techniques basées sur la théorie des jeux pour analyser un ensemble de scénarios d'interactions entre les agents. Par exemple, Rosenschein a montré comment on peut réaliser une certaine coordination sans recourir à la communication : les agents calculent simplement le meilleur gain et savent que les autres suivent la même approche. La coordination apparaît à travers l'hypothèse de la rationalité mutuelle des agents.

Par conséquent, les outils de la théorie des jeux sont utilisés pour coordonner des agents rationnels autonomes sans utiliser de mécanisme de coordination explicitement intégré dans le modèle des agents. L'application la plus courante de la théorie des jeux pour les systèmes multi-agents est certainement la négociation. Dans ce cas, la fonction d'utilité utilisée est dérivée de la théorie des jeux et repose sur une matrice de paiements relative au gain et au coût induit par chaque action possible. Cette matrice est une connaissance commune aux parties aux négociations. Le processus de négociation met en œuvre un processus interactif d'offres et de contre-offres pour lequel chaque agent choisit une action qui maximise la valeur d'utilité attendue. À chaque étape de la négociation, chaque agent évalue l'offre de l'autre dépendant de sa propre stratégie de négociation [77] [78].

Parallèlement à ce travail, d'autres ont été amenés à prendre en compte le fait qu'un agent ne fait pas nécessairement confiance à un autre. En effet, si un agent négociateur veut maximiser ses chances personnelles, il est tenté d'induire en erreur et de manipuler les autres en retenant des informations ou en les désinformant [79]. D'autres travaux ont également été réalisés à partir de la théorie des jeux, notamment en ce qui concerne la prise en compte de la durée du processus de négociation via la mise en œuvre d'un modèle de stratégie spécifique [77]. Depuis lors, les recherches de plus en plus nombreuses en intelligence artificielle ont permis le développement et la mise en œuvre de nombreuses applications industrielles et commerciales reflétant les relations entre agents et principes de la théorie des jeux.

Ces dernières années, la coordination selon les principes de la théorie des jeux a été utilisée dans différentes activités de recherche telles que la simulation de la circulation routière [80]. Ils ont utilisé un mécanisme basé sur la théorie des jeux qui vise à coordonner le comportement des agents afin de traiter les problèmes générés dans des situations de trafic critiques telles que les jonctions. Ce mécanisme résout les conflits en permettant aux agents de réaliser la jonction un à un. Dans un contexte étroit (**Road pricing**) [81], un mécanisme de coordination de la théorie des jeux a été utilisé pour prouver qu'il existait au moins une stratégie vérifiée de l'équilibre de Nash pour le jeu de la congestion à l'étude. En outre, dans le cas de « Engineering Asset Management » (EAM) [82], la stratégie de coordination de la théorie des jeux a été utilisée pour fournir les modèles de décision pour la prise de décision opérationnelle, tactique et stratégique lors de pratiques de maintenance en collaboration.

2.10 Conclusion

Ce chapitre a fourni un aperçu de la coordination dans les systèmes multi-agents. La coordination a été jugée très importante dans la conception et la mise en œuvre de tout système multi-agents. Sans une bonne coordination, de nombreux avantages du paradigme multi-agents disparaîtront. Des recherches approfondies ont été effectuées dans ce domaine, et ces recherches se poursuivent même maintenant. Le défi, est de pouvoir fournir un cadre guidant le choix d'une technique de coordination basée sur les propriétés d'un système multi-agents particulier et de concevoir un moyen d'évaluer la complexité et la pertinence de chaque technique.

A partir de ce chapitre, il est possible d'identifier quatre composants principaux qui doivent être présents dans toute technique de coordination complète :

1. Il est important d'essayer d'anticiper toutes les actions potentielles que les agents d'un système pourraient éventuellement entreprendre et d'imposer des restrictions interdisant les interactions indésirables. Ces «lois sociales» garantissent que le système multi-agents ne deviendra pas ingérable et erratique. C'est la même chose dans les sociétés humaines.
2. La structure doit être flexible pour que les agents puissent opérer dans un environnement dynamique : autrement dit, les lois ne doivent pas être trop restrictives.
3. Il doit exister des structures sociales ou des étiquettes décrivant comment les agents devraient se comporter les uns envers les autres lorsqu'ils participent à des processus de coordination - c'est vraiment comme des protocoles.
4. Les agents doivent avoir des connaissances et des capacités de raisonnement suffisantes pour exploiter à la fois la structure disponible et la flexibilité. Cependant, je ne pense pas qu'une technique complète soit ce qui est nécessaire pour chaque système. Il existe des systèmes dans lesquels les structures ne nécessitent pas d'excellentes techniques de coordination (la structure et l'organisation sont définies de manière stricte) et des compromis doivent être faits le cas échéant.

Chapitre 3

3 Une nouvelle approche de la poursuite en coopération multi-agents basée sur le modèle de l'élimination itérative des stratégies dominées avec changement de priorités d'élimination (IEDSEPC).

Contribution

Contenu

3	Une nouvelle approche de la poursuite en coopération multi-agents (IEDSEPC)	56
3.1	Introduction.....	57
3.2	Description du problème de poursuite-évasion.....	57
3.2.1	Le poursuivant	58
3.2.2	L'évadé.....	58
3.3	Élimination itérative des stratégies dominées (IEDS)	59
3.4	Notre contribution	60
3.4.1	Proposition d'un algorithme de formation de coalition basé sur IEDSEPC.....	60
3.4.2	Processus Décisionnel de Markov	65
3.5	Conclusion	68

3.1 Introduction

Dans ce chapitre, nous proposons un mécanisme de coordination des tâches flexible basé sur les principes de la théorie des jeux. L'objectif principal de cette approche est de fournir une stratégie d'agents répondant aux attentes d'un équilibre conditionné. En d'autres termes, nous avons basé sur le model d'élimination itérative des stratégies dominant avec une priorité d'élimination dynamique IEDSEPC pour obtenir l'équilibre avec la récompense moyenne la plus élevée et le degré de dynamisme le plus faible. Afin de refléter l'amélioration apportée par cette méthode, nous l'avons appliquée au problème Poursuit-Evasion par rapport à une autre approche de la théorie des jeux dans laquelle les joueurs choisissent de jouer à la stratégie fournie par IEDS avec une priorité d'élimination statique. Ainsi, nous étudions le temps moyen de capture des évadés, le développement de l'acquisition de récompenses lors de l'exécution des tâches, ainsi que le degré de stabilité des groupes de poursuite. De plus, nous avons utilisé les principes du processus de décision de Markov (MDP) afin de contrôler la stratégie de déplacement de chaque agent. Ce processus (MDP) fournit un formalisme de modèle et résout les problèmes de planification de trajectoire et d'apprentissage dans des conditions d'incertitude [86], [87].

3.2 Description du problème de poursuite-évasion

Dans cette partie, nous nous concentrons sur le problème de coopération dans lequel n poursuivants situés dans un environnement de grille toroïdal restrictif doivent capturer m évadés de types différents. Les expressions de $P = \{P_1, \dots, P_n\}$ et $E = \{E_1, \dots, E_m\}$ représentent la collection de n poursuivants et de m évadés, respectivement. Les poursuivants et les évadés représentent les rôles que les agents peuvent jouer. Nous supposons ici que ceux-ci peuvent évaluer les types des évadés après la localisation. Il existe des obstacles fixes de formes et de tailles différentes dans l'environnement, La position pourrait être destinée à la cartographie mp :

$X \rightarrow \{0,1\}$, tel que $\forall x \in : mp(x) = 1$, bien alors x est un obstacle.

Dans cette partie, nous ne présenterons que les principaux éléments de notre proposition. Prenons un tel scénario dans un environnement rectangulaire bidimensionnel et limité X avec n cellules carrées et deux types d'agents.

3.2.1 Le poursuivant

L'ensemble des poursuivants est désigné par : $P = \{P_1, \dots, P_n\}$

3.2.1.1 Paramètres de capacité d'un poursuivant :

- *Niveau de confiance* : dans les systèmes multi-agents, chaque agent doit pouvoir répondre aux services demandés par tout autre agent. Le degré de confiance en soi est l'évaluation du succès de l'agent pour la tâche affectée. Il est noté et calculé de la manière suivante :

$$\forall Conf \in [0.1, 1] : Conf = \max\left(0.1, \frac{C_s}{C_t}\right)$$

C_s est le nombre de tâches terminées par l'agent. C_t est le nombre de tâches auxquelles l'agent a participé.

- *Le crédit*. Dans le cas où l'agent ne peut pas exécuter une tâche, son crédit sera affecté. Le crédit d'un agent est désigné et calculé comme suit :

$$\forall Credit \in [0, 1] : Credit = \min\left(1, 1 - \frac{C_b}{C_t - C_s}\right).$$

C_b est le nombre de tâches abandonnées par l'agent.

- *Position de l'environnement*. La position de l'agent dans l'environnement est un critère essentiel pour les séquences de poursuite, car la capture sera plus facile si le poursuivant est plus proche de l'évité. La position Pos est calculée comme suit :

$$Pos = Dist(S_P, S_E).$$

S_P est l'état (cellule) du poursuivant, S_E est l'état (cellule) de l'évité. Dist est la distance entre le poursuivant et l'évité.

$$Dist(S_P, S_E) = \sqrt{(CC_{P_i} - CC_{E_i})^2 + (CC_{P_j} - CC_{E_j})^2}.$$

3.2.2 L'évité

L'ensemble des évités est désigné par : $E = \{E_1, \dots, E_m\}$

Chaque évité est caractérisé par un type Re, avec $Re \in \{I, II, III, IV\}$ pour indiquer combien de poursuivants sont nécessaires pour le capturer ;

- Re = I : signifie que cet évité a besoin d'un poursuivant pour être capturé.
- Re = II : signifie que cet évité a besoin de deux poursuivants pour être capturé.
- Re = III : signifie que cet évité a besoin de trois poursuivants pour être capturé.
- Re = IV : signifie que cet évité a besoin de quatre poursuivants pour être capturé.

Afin de distinguer les différentes coalitions, chaque poursuivant appartenant à une coalition calcule la valeur qui lui est retournée grâce à cette stratégie. Ce calcul est basé sur les facteurs caractérisant les poursuivants. Par exemple, un poursuivant (P_1) appartient à la coalition (Co) ; la valeur de cette coalition par rapport à ce poursuivant est calculée comme suit :

$$Co_{(val,P_1)} = \frac{Coef_1 \times Conf_1 + Coef_2 \times Credit_1 + Coef_3 \times Pos_1}{\sum_{k=1}^3 Coef_k}$$

$$+ \sum_{i=2}^{Re} \frac{Coef_1 \times Conf_i + Coef_2 \times Credit_i + Coef_3 \times Pos_i}{Re \times \sum_{k=1}^3 Coef_k} \dots \dots \dots (3 - 1)$$

$Coef$ est le coefficient de chaque facteur.

Sur la base de ces valeurs et en utilisant la méthode IEDS, notre mécanisme sera en mesure de sélectionner la coalition de poursuites optimale pour chaque évadé détecté. Comme détaillé dans l'algorithme de formation de coalition basé sur IEDS

3.3 Élimination itérative des stratégies dominées (IEDS)

La coalition est un ensemble de poursuivants nécessaires pour capturer les évadés détectés. Dans la coalition, chaque poursuivant doit correspondre à une stratégie spécifique. Dans notre proposition, une stratégie pure s_i définit l'intégration d'un groupe de poursuites spécifique que le poursuivant suivra dans toutes les situations possibles et réalisables au cours de la poursuite. De telles coalitions peuvent ne pas être aléatoires, ni tirées d'une distribution, comme dans le cas de stratégies mixtes. Une stratégie str_i domine une autre stratégie str'_i si et seulement si pour chaque combinaison potentielle des actions des autres joueurs str_{-i} :

$$\mu_i(str_i, str_{-i}) \geq \mu_i(str'_i, str_{-i})$$

μ : fonction renvoyant le résultat obtenu via une stratégie spécifique.

Considérez le jeu stratégique présenté dans le tableau 3-1, où le joueur de colonne a trois stratégies pures et le joueur de rangée seulement deux (a). Sachant cela, les valeurs affichées dans chaque cas représentent les gains attendus retournés aux joueurs dans le cas de la sélection de la stratégie actuelle. Jouer au centre est toujours mieux que de jouer du côté droit pour le joueur de colonne. Par conséquent, nous pouvons supposer qu'il finira par arrêter de jouer à droite car il s'agit d'une stratégie dominée (b). Nous pouvons donc ignorer la colonne de droite après son élimination. Maintenant, le joueur de ligne a une stratégie dominée, Haut. Finalement, le joueur de ligne cesse de jouer Haut, puis la ligne - Haut est éliminée (c). Enfin, nous avons deux choix restants Bas-

Gauche et Centre-Bas, le joueur de colonne s'aperçoit qu'il ne peut gagner qu'en jouant à gauche (d). On peut donc en déduire que la solution IEDS est (Bas, Gauche) avec le gain suivant : (6, 6).

Tableau 3-1: Application de la technique IEDS

(a)	Gauche	Centre	Droite
Haut	5,4	3,8	1,5
Bas	6,6	6,0	-3,-1
(b)	Gauche	Centre	Droite
Haut	5,4	3,8	1,5
Bas	6,6	6,0	-3,-1
(c)	Gauche	Centre	Droite
Haut	5,4	3,8	1,5
Bas	6,6	6,0	-3,-1
(d)	Gauche	Centre	Droite
Haut	5,4	3,8	1,5
Bas	6,6	6,0	-3,-1

3.4 Notre contribution

3.4.1 Coordination de tâches : Proposition d'un algorithme de formation de coalition basé sur IEDSEPC

Dans notre proposition d'un Algorithme de formation de coalition basé sur l'élimination itérative des stratégies dominées avec changement de priorités d'élimination (IEDSEPC), Un certain nombre d'algorithmes de formation de coalitions ont été développés pour définir laquelle des coalitions potentielles devrait réellement être formée. Pour ce faire, ils calculent généralement une valeur pour chaque coalition, appelée valeur de coalition, qui fournit une indication des résultats attendus pouvant être obtenus si cette coalition est constituée. Ensuite, après avoir calculé toutes les valeurs de la coalition, la décision concernant la coalition optimale à former peut être sélectionnée.

Dans notre contribution, nous utilisons un algorithme itératif afin de déterminer les coalitions optimales d'agents. Il commence par un ensemble complet de coalitions (combinaisons agent-stratégie) et élimine de manière itérative les coalitions, cette élimination respecte une priorité absolue. Le pseudocode de notre algorithme est montré ci-dessous :

Tableau 3-2 : Algorithme de coalition de poursuite basé sur IEDSEPC

Algorithme de coalition de poursuite basé sur IEDSEPC

Repeat

Move (evaders);

Reset(agents_positions);

While (priority_possibility \neq true) **do**

Reset (list_coalitions);

Change (elimination_priority);

While (number of coalitions $>$ 1) **do**Eliminate the dominated strategy of P_i ; $i \leftarrow i \bmod n + 1$;**End_while**Equilibrium \leftarrow last_coalition;List_equilibrium \leftarrow Equilibrium;**End_while****While** (list_equilibrium $\neq \emptyset$) **do****If** (equilibrium.reward = max **and** equilibrium.dynamism = min) **then**Selected_equilibrium \leftarrow equilibrium;**End_if**

Delete(equilibrium);

End_while

Role_attribution (selected_equilibrium);

Pursuit_iteration(pursuers);

If (move (evaders) = false) **then**

Capture = true;

Until (capture=true)

Le calcul des coalitions possibles pour capturer les évadés détectés est effectué après l'itération des évadés (mouvement des évadés). Les valeurs des poursuivants formant ces coalitions sont calculées en fonction de leurs dernières positions par rapport aux positions des évadés.

Le nombre attendu de coalitions possibles à former est calculé selon :

$$\begin{aligned}
 Nbr_{cl} &= \frac{n!}{(n-R_{e1})!R_{e1}} \times \frac{(n-R_{e1})!}{(n-(R_{e1}+R_{e2}))!R_{e2}} \times \dots \times \frac{(n-(R_{e1}+R_{e2}+\dots+R_{eN-1}))!}{(n-(R_{e1}+R_{e2}+\dots+R_{eN}))!R_{eN}} \\
 &= \prod_{j=1}^N \frac{(n - \sum_{k=0}^{k=j-1} R_{ek})!}{(n - \sum_{k=0}^{k=j} R_{ek})! R_{ej}} \dots \dots \dots (3 - 2)
 \end{aligned}$$

- n : nombre de poursuivants dans l'environnement.
- N : nombre d'évader détectés.
- $R_{e0} = 0$

Afin de répartir le calcul des coalitions possibles entre les poursuivants, les coalitions générales possibles (Ω) seront calculées. Une coalition générale inscrit tous les poursuivants nécessaires pour capturer l'ensemble des évadés détectés :

$$\Omega = \frac{n!}{(n - \lambda)! \lambda!} \dots \dots \dots (3 - 3)$$

$$\lambda = (R_{e1} + R_{e2} + \dots + R_{eN}) \dots \dots \dots (3 - 4)$$

Les coalitions générales générées seront réparties équitablement entre les agents jouant le rôle de poursuivant. Plus précisément, chaque coalition générale sera composée de N groupes de poursuite. A partir de chaque coalition générale générée par le précédent calcul équation (3), un nombre de formations de coalition possibles (\mathcal{U}) sera calculé :

$$\mathcal{U} = \prod_{j=1}^N \frac{(\lambda - \sum_{k=0}^{k=j-1} R_{ek})!}{(\lambda - \sum_{k=0}^{k=j} R_{ek})! R_{ej}} \dots \dots \dots (3 - 5)$$

$$Nbr_{cl} = \Omega \times \mathcal{U} = \frac{n!}{(n - \lambda)! \lambda!} \times \prod_{j=1}^N \frac{(\lambda - \sum_{k=0}^{k=j-1} R_{ek})!}{(\lambda - \sum_{k=0}^{k=j} R_{ek})! R_{ej}} \dots \dots \dots (3 - 6)$$

Cette technique décentralisée vise à équilibrer le calcul des formations de coalition possibles parmi les poursuivants. En outre, cette méthode est plus détaillée dans le chapitre suivant via son application à l'étude de cas.

Suivant notre algorithme, les poursuivants éliminent alternativement les pires coalitions qu'ils puissent jouer selon l'algorithme IEDSEPC jusqu'à ce qu'ils atteignent l'équilibre. La pire coalition est celle qui offre la récompense la plus basse au joueur concerné. Sachant cela, cette élimination respecte une priorité absolue. Le nombre de priorités d'élimination de n agents est calculé comme suit :

$$\Psi = n!$$

Ensuite, les priorités d'élimination sont modifiées en fonction de la priorité suivante, les valeurs des coalitions sont réinitialisées et l'élimination est répétée pour obtenir un autre équilibre. Ce processus est répété en utilisant toutes les priorités d'élimination possibles pour obtenir tous les équilibres possibles. Enfin, le meilleur équilibre est choisi selon ces deux conditions :

- ✓ La maximisation de la récompense moyenne des poursuivants formant le meilleur équilibre.
- ✓ La minimisation des changements de rôle entre l'agent formant le nouvel équilibre optimal et ceux constituant le précédent.

L'opération de sélection à l'équilibre est répétée avant toute itération de poursuite (mouvement des poursuivants) et après la mise à jour des positions des agents. Les poursuivants sont immédiatement affectés en fonction du meilleur équilibre sélectionné. La capture est réalisée lorsque les évadés détectés ne peuvent plus se déplacer.

La Figure 3-1 montre l'organigramme de cet algorithme de poursuite reprenant les différentes étapes expliquées dans cette partie, de la détection à la capture des évadés existants.

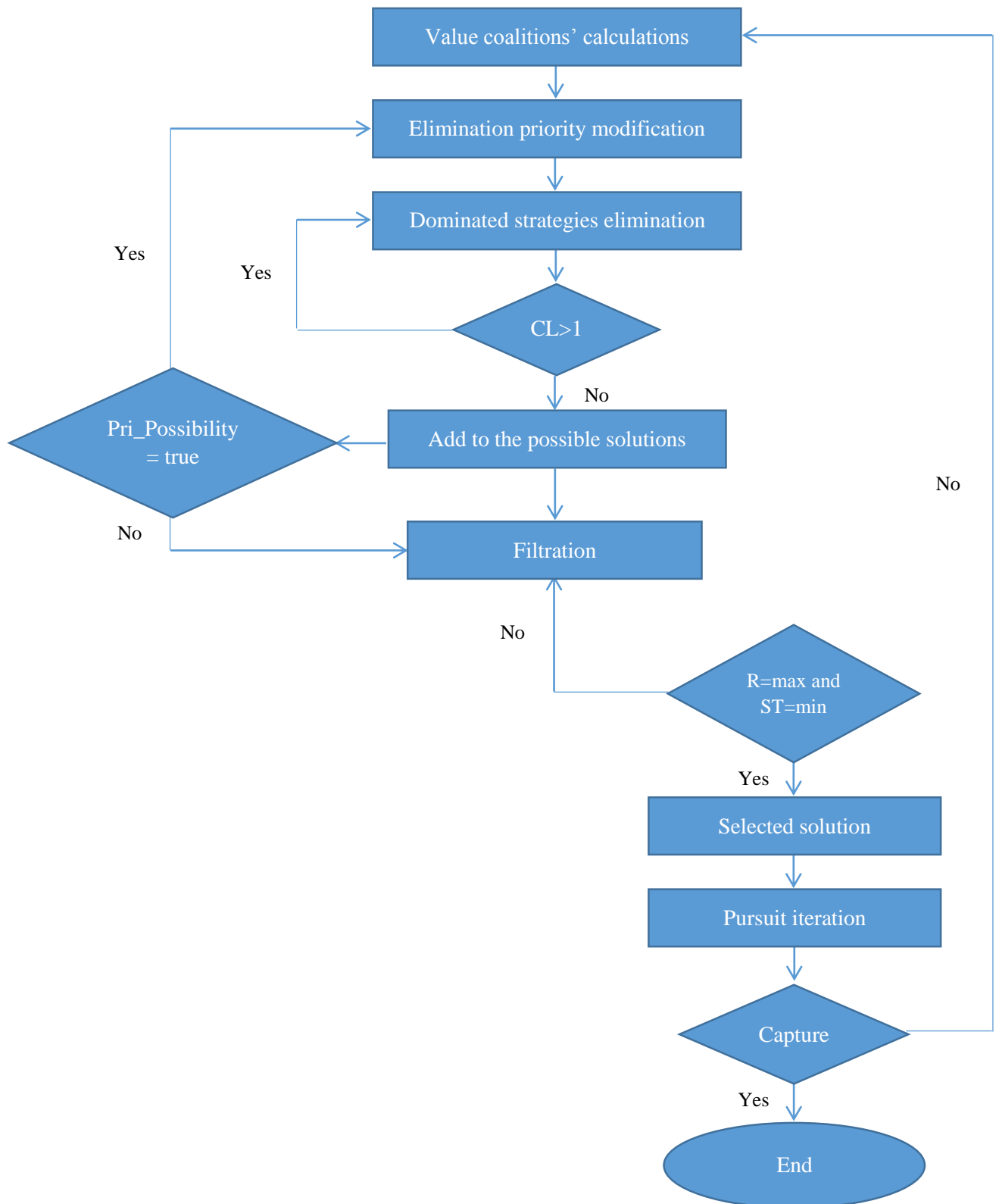


Figure 3-1 : Organigramme de l'algorithme de poursuite basé sur IEDSEPC

3.4.2 Coordination du mouvement : Processus Décisionnel de Markov

Les processus de décision de Markov (MDP) fournissent un cadre mathématique pour la prise de décision d'un modèle dans des situations où les situations sont quelque peu aléatoires et en partie sous le contrôle d'un décideur. Dans les systèmes multi-agents coopératifs, MDP permet de formaliser des problèmes de décision séquentiels. Ce processus ne modèle que les systèmes coopératifs dans lesquels la fonction de récompense est partagée par tous les acteurs. MDP est défini par (N, S, A, T, R) comme suit :

- N est le nombre d'agents Ag_i dans le système, $i \in \{1, \dots, N\}$
 - S correspond à l'ensemble des états des agents s .
 - $A: A = A_1 \times A_2 \times \dots \times A_N$: définit l'ensemble des actions conjointes des agents, où A_i est l'ensemble des actions locales de l'agent Ag_i
 - T est la fonction de transition. Il renvoie la probabilité $T(s, a, s')$, ce qui signifie que l'agent passe à l'état s' s'il exécute l'action conjointe $a \in A$ à partir de l'état s .
 - R définit la fonction de récompense.
 - $R(s, a, s')$ représente la récompense obtenue par l'agent lorsqu'il passe de l'état s à l'état s' par l'exécution de l'action a .
- *Fonction de récompense.* Dans le problème MDP, les prochains états sélectionnés sont les états retournant le maximum de récompense définitive. Dans notre proposition, nous avons utilisé des fonctions heuristiques afin de calculer la récompense immédiate de chaque état. La fonction de récompense définit les objectifs que les poursuivants doivent atteindre et identifier les obstacles environnementaux. Pour calculer cette fonction, nous avons utilisé la position environnementale des agents, qui permet une répartition équitable des récompenses sur les cellules environnementales. Le calcul des récompenses dans chaque état concerné est effectué comme suit

$$R(s, a) = \begin{cases} \gamma & \text{if } E_i \subseteq s \\ 0 & \text{if } mp(x) = 1 \\ \gamma - Val(Dist(CC_P, CC_E)) & \text{else.} \end{cases}$$

- γ C'est la récompense maximale.
- $Val(Dist(CC_P, CC_E))$: représente la valeur de la distance.

En ce qui concerne la distribution des récompenses dans les cellules standard, notons que la fonction de récompense est inversement proportionnelle à la fonction de distance.

La Figure 3-2 illustre une partie de notre environnement de simulation. Les valeurs affichées dans les différentes cellules $[V_1, V_2, V_3]$ représentent les gains générés par la fonction de récompense. Les récompenses dynamiques seront attribuées à tout poursuivant situé dans la cellule concernée pendant la poursuite.

V_1 : la récompense pourrait être obtenue si le poursuivant concerné suit le premier évadé.

V_2 : la récompense aurait pu être obtenue si le poursuivant concerné suit le deuxième évadé.

V_3 : est l'indice de la cellule (est égale à 1 si la cellule est occupée ou 0 si la cellule est libre).

- *Fonction de transition.* Les probabilités de transition (p) décrivent le dynamisme de l'environnement. Ils jouent le rôle de fonction de l'état suivant dans une recherche de résolution de problème, sachant que chaque état peut être le prochain état possible en fonction de l'action entreprise dans l'état actuel. Notre approche est développée dans une grille de cellules où chaque agent peut se déplacer dans quatre états différents : S_{haut} , S_{bas} , S_{gauche} et S_{droite} .

Les probabilités de transition des poursuivants sont basées sur le degré de récompense, comme indiqué :

$$\sum_{s'} p(s' | s, a) = 1,$$

$$p(s' | s, a) = \frac{R(s', a)}{\gamma},$$

$$p(s' | s, a) = \max(p(s | s, a), p(s_{up} | s, a),$$

$$p(s_{down} | s, a), p(s_{right} | s, a), p(s_{left} | s, a)),$$

$$\forall s, a$$




									
[41 97 0]	[40 98 1]	[39 97 0]	[38 96 0]	[37 95 0]	[36 94 0]	[35 93 0]	[34 92 0]	[33 91 1]	
[42 96 0]	[41 97 0]	[40 96 0]	[39 95 0]	[38 94 0]	[37 93 0]	[36 92 0]	[35 91 0]	[34 90 0]	
[43 95 0]	[42 96 0]	[41 95 0]		[39 93 0]	[38 92 0]		[36 90 0]	[35 89 0]	
[44 94 0]	[43 95 0]		[41 93 0]	[40 92 0]	[39 91 0]	[38 90 0]	[37 89 0]	[36 88 0]	

Figure 3-2 : fonction de récompense appliquée à l'environnement de grille

Cellules avec cadre rouge : les états sélectionnés ; agents bleus : poursuivants ; agents verts : évadés ; cellules noires : cellules contenant des obstacles.

Les liens entre l'évadé et chaque poursuivant illustrés à la Figure 3-3 reflètent les trajectoires optimales fournies par l'application du procédé proposé dans cette partie au cours de chaque étape de poursuite différente.

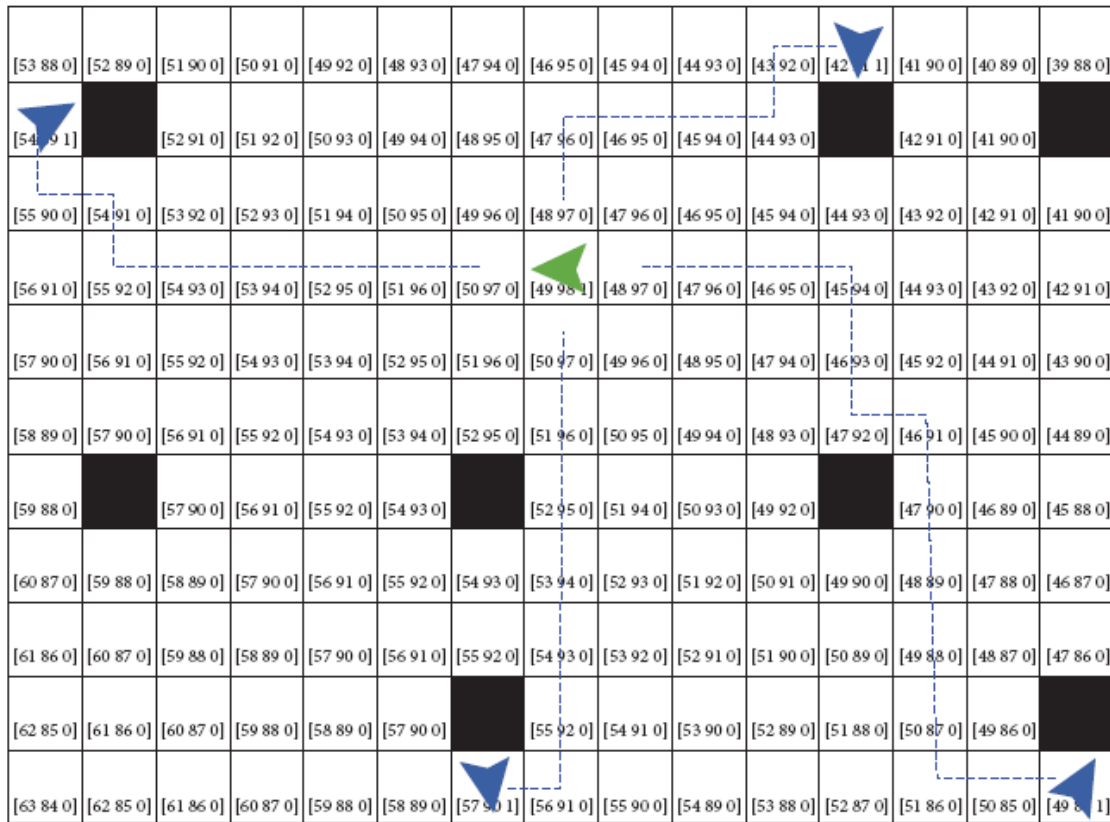


Figure 3-3 : Prédiction du comportement des poursuivants après l'application de la fonction de transition

3.5 Conclusion

Ce chapitre présente une sorte de méthode de coalition décentralisée basée sur les principes de la théorie des jeux pour différents types de poursuites ; la méthode proposée démontre l'impact positif imposé par le dynamisme des formations de coalition. Premièrement, nous avons étendu notre Algorithme de formation de coalition basé sur l'élimination itérative des stratégies dominées avec changement de priorités d'élimination (IEDSEPC). Ce processus nous permet de déterminer la stratégie de coalition de poursuite optimale conformément aux principes de la théorie des jeux. Deuxièmement, nous nous sommes concentrés sur le processus de décision de Markov en tant que stratégie de déplacement de nos poursuivants dans l'environnement (grille de cellules). Pour mettre en évidence notre proposition, nous avons développé dans le chapitre suivant (chapitre4) une étude comparative entre notre algorithme et une stratégie de coalition décentralisée récent basée sur le modèle IEDS. Les résultats de simulation présentés dans le chapitre4 démontrent que l'algorithme basé sur IEDSEPC est réalisable et efficace.

Chapitre 4

4 Expériences de simulation

Etude des cas

Contenu

4	Etude des cas : Expériences de simulation.....	69
4.1	Introduction.....	70
4.2	La programmation orientée agents.....	71
4.3	Expériences de simulation.....	78
4.4	Conclusion	82

4.1 Introduction

Les systèmes multi-agents (SMA) ont montré leur pertinence pour la conception d'applications distribuées, complexes et robustes. Le concept d'agent est aujourd'hui plus qu'une technologie efficace, il représente un nouveau paradigme pour le développement de logiciels dans lesquels l'agent est un logiciel autonome qui a un objectif, évolue dans un environnement et interagit avec d'autres agents au moyen de langages et de protocoles. Souvent, l'agent est considéré comme un objet « intelligent » ou comme un niveau d'abstraction au-dessus des objets et des composants. Les méthodes de développement orientées objet – au vu des différences entre les objets et les agents – ne sont pas directement applicables au développement de SMA. Il est alors devenu nécessaire d'étendre ou de développer de nouveaux modèles, de nouvelles méthodologies et de nouveaux outils adaptés au concept d'agent.

Après la révolution de la conception et de la programmation orientée objet, nous sommes donc à l'aube d'une nouvelle révolution qui serait celle de la conception et de la programmation orientée agent/interaction/organisation. Le constat concernant la conception de SMA, dans les années 1998-1999, est que le développement de SMA est coûteux en temps à cause de leur complexité, mais aussi par le fait que pour chaque application, il faut créer tout le SMA adéquat. En effet, la majorité des applications existantes en SMA ont été développées de manière ad hoc [88]. Ce foisonnement a conduit en parallèle à de multiples propositions de modèles d'agents, notamment par le rapprochement effectué avec l'approche objet. De là, plusieurs formalismes sont apparus, chacun mettant en avant une représentation de l'agent et de son système. En 1999, se fait donc sentir le besoin de fournir des modèles, des méthodologies, des plates-formes pour faciliter la prise en compte de la complexité des systèmes à concevoir [89] et pour aider les concepteurs qui ne sont pas nécessairement spécialistes des SMA. Parallèlement au développement des méthodes, des modèles ont été définis pour aider les concepteurs tels que AGR [90] ou BDI [91]. Pour le développement, les concepteurs disposent de langages orientés agent dont les précurseurs ont été Agent0 et son évolution PLACA, ou LALO et METATEM. Des approches impératives (comme JAL1), déclaratives (comme CLAIM [92]) ou hybrides (comme 3APL2) ont vu le jour depuis [93], liées ou non à des plates-formes de développement dont les plus célèbres sont NetLogo, Madkit3 ou JADE4. Bien entendu, l'utilisation de langages de programmation classiques (C++, Java, etc.) reste envisageable.

4.2 La programmation orientée agents

La programmation orientée agents a été proposée par Yoav Shoham en 1993 comme un nouveau paradigme de programmation, que l'on peut voir comme une spécialisation de la programmation orientée objets. Dans cette approche, les agents sont les éléments centraux du langage, de la même façon que les objets sont centraux pour les langages orientés objets. La perspective sur les agents est cognitive : les agents sont caractérisés par des notions mentales comme leurs croyances, leurs décisions et leurs obligations. De plus, chaque agent est associé un ensemble d'habiletés qui représentent ce que l'agent sait faire. En même temps, la programmation orientée agents suppose qu'on va développer des programmes dans lesquels plusieurs agents interagissent, ce qui met l'accent sur la dimension sociale des agents. Le langage de programmation proposé par Shoham comme démonstration de ce nouveau paradigme s'appelle AGENT0.

La différence principale entre un tel langage et un langage de programmation classique que l'on pourrait utiliser pour développer des agents, vient du fait que les notions mentales qui caractérisent les agents apparaissent dans le langage lui-même, et que la sémantique du langage est intimement liée à la sémantique de ces notions mentales. La programmation orientée agents peut être vue comme une spécialisation de celle orientée objets parce que les modules du programme sont maintenant des agents, c'est-à-dire des objets avec un état qui définit les notions mentales associées, et que les messages entre objets sont remplacés par des messages entre agents. En quoi les messages entre agents diffèrent-ils de ceux entre objets ? Premièrement, parce que ces messages sont modélisés en partant de la théorie des actes de parole, qui s'intéresse aux actions de communication comme informer, demander, offrir, accepter, rejeter et d'autres. Deuxièmement, puisque les agents sont autonomes et dotés de capacités mentales, ils ont la liberté de décider s'ils vont ou non exécuter l'action spécifiée dans le message. Par contraste, un objet recevant un message va toujours exécuter l'action spécifiée dans le message.

Dans cette partie, nous ne présenterons la plateforme de notre simulation NetLogo, ainsi que de brèves définitions sur deux plateformes les plus célèbres JADE et MadKit.

4.2.1 La Plateforme JADE

4.2.1.1 Définition (Java Agent DEvelopment Framework)

JADE (Java Agent DEvelopment framework) est une plate-forme multi-agent créé par le laboratoire TILAB et décrite par Bellifemine et al. Dans [96],[97]. JADE permet le développement de systèmes multi-agents et d'applications conformes aux normes FIPA [98]. Elle est implémentée en JAVA et fourni des classes qui implémentent « JESS » pour la définition du comportement des agents. JADE possède trois modules principaux (nécessaire aux normes FIPA).

- DF « Director Facilitator » fournit un service de « pages jaunes » à la plate-forme ;
- ACC «Agent Communication Channel » gère la communication entre les agents ;
- AMS « Agent Management System » supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système. Ces trois modules sont activés à chaque démarrage de la plate-forme.

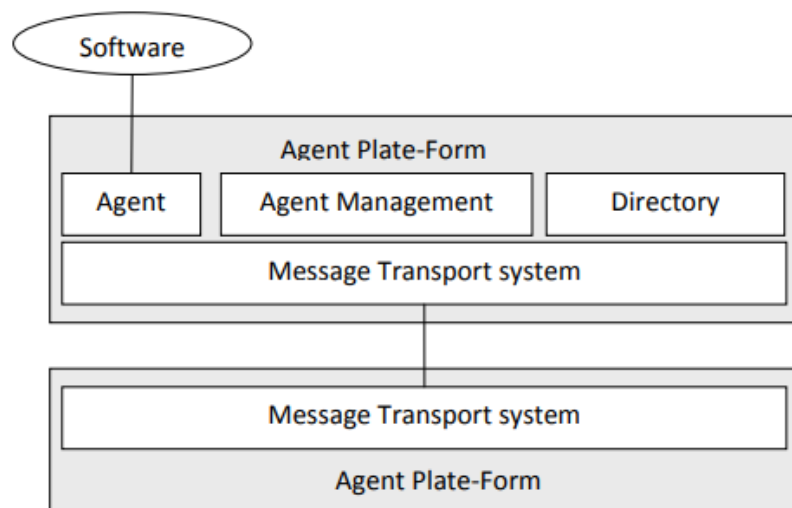


Figure 4-1 : Architecture logiciel de La plate-forme JADE

Les langages de communication de la plateforme jade sont deux :

- Le langage de Communication de la plate-forme JADE est FIPA-ACL (Agent Communication language).
- La classe ACLMessage représente les messages qui peuvent être échangés par les agents. La communication de messages se fait en mode asynchrone.

JADE utilise deux protocoles de communication :

- Les protocoles FIPA-Query et FIPA-Request.
- Le protocole Contract-Net de JADE

4.2.2 La Plateforme MADKIT

4.2.2.1 Définition (*Multi agent development Kit*) :

MadKit (Multi agent development Kit), est une plate-forme multi-agents, implémentée en java et à l'originalité d'être basé sur un modèle organisationnel « modèle AGR » plutôt qu'une architecture d'agent ou un modèle d'interaction spécifique [100]. L'utilisation de groupes et de rôles associés à des agents est mis en œuvre tant en tant qu'outil de modélisation et de conception pour les développeurs de systèmes multi-agents, que de principe d'architecture de la plate-forme elle-même. Cette architecture est basée sur un noyau agent minimal découplé de tout modèle individuel d'agent. Elle permet d'intégrer au sein d'un même outil une grande variété d'architectures d'agents et de modèles de communication. De plus, les outils de communication sont déjà implémentés et gèrent les envois de messages distants (en utilisant le protocole TCP/IP). Les agents MadKit sont des Threads Java, ils sont donc indépendants les uns des autres et évoluent de manière asynchrone.

La structure organisationnelle est implémentée au cœur de la plate-forme MADKIT tant pour fournir un modèle organisationnel aux SMA exécutés que pour le fonctionnement interne du système. Cette plate-forme est basée sur trois principes :

- Architecture à micronoyau
- Agentification systématique des services.
- Découplage applicatif entre noyau, agents et application d'accueil.

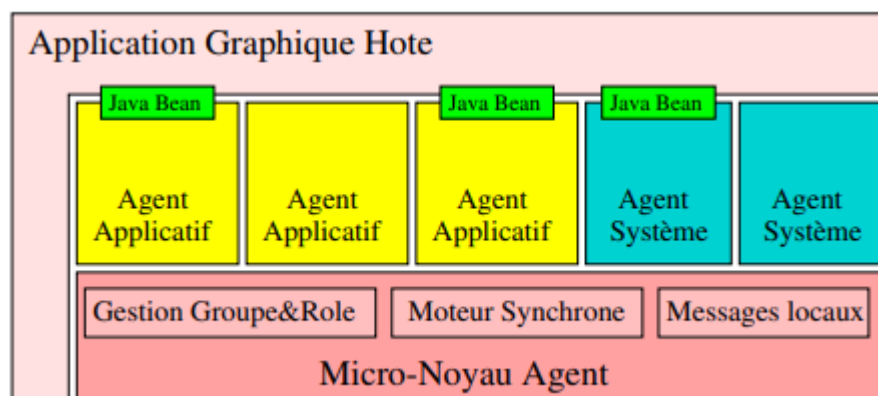


Figure 4-2 : Architecture Générale du MadKit

Concrètement, MADKIT est un ensemble des packages Java qui implémentent le noyau agent, diverses bibliothèques de base de messages, d'agents et de sondes.

Le principe essentiel de MADKIT est d'utiliser partout où cela est possible la plateforme pour son propre fonctionnement. Tous les services hors ceux assurés par le micronoyau sont implémentés par des agents à part entière. Ceci vient à la fois d'un souci d'unicité de modèle, mais

également de mise à l'épreuve des SMA comme modèle de programmation. Cela le rapproche également fortement des approches méta en architectures objets [94] à ceci près que nous plaçons la relation au méta entre agents applications et "systèmes" (communication, ordonnancement, surveillance...).

MADKIT n'est pas une plate-forme agent dans le sens classique, centrée autour d'un modèle particulier d'agent ou d'interaction. La taille réduite et les fonctionnalités du noyau de base, la neutralité des types agents, associée à ce principe de services agentifiés et de découplage par rapport aux applications "hôtes" permet en fait d'obtenir toute une gamme d'environnements d'exécutions aux finalités parfois complètement opposées.

D'après [95], la principale caractéristique de MadKit est son micro- noyau agent. Ce centrage sur l'infrastructure simplifie les phases de développement et de déploiement. Les phases d'analyse et de conception sont encore peu documentées, le modèle AGR étant plutôt un modèle organisationnel descriptif qu'une méthode de conception. Le principal point faible de MadKit est son manque de modèle d'agent. La construction d'agent complexe nécessite l'écriture de beaucoup de code Java. De plus, cette propriété est aussi un point fort sous certains aspects, car elle offre une plus grande flexibilité, et permet à tout programmeur Java de commencer à utiliser la plate-forme, ce qui est un atout dans un contexte pédagogique.

4.2.3 La plateforme NetLogo

Des plates-formes de simulation permettant un prototypage et des tests rapides des idées de conception initiales et des alternatives peuvent améliorer de manière significative le processus de mise en œuvre de systèmes complexes agents et multi-agents (A-SMA). Une telle plate-forme devrait idéalement se caractériser par une courte période d'apprentissage, une mise en œuvre simple et l'affichage de l' A-SMA en cours de développement, tout en tenant compte des fonctionnalités de programmation orientées agent permettant de transférer les alternatives de conception à un environnement de développement d'agent complet sans aucune difficulté. Cependant, ces exigences spécifiques font de cette plateforme de simulation un outil d'apprentissage parfait.

NetLogo [99] a été créé par le Dr Uri Wilensky au Centre pour l'apprentissage connecté et la modélisation assistée par ordinateur (CCL), situé à l'Université Tufts, Medford, MA. En 2000, le CCL a été transféré à l'université du nord-ouest d'Evanston, dans l'Illinois, où le traitement de NetLogo s'est poursuivi jusqu'à maintenant.

NetLogo est un langage de programmation multi-agents simple et primitif basé sur le langage de programmation Logo. Cette plateforme est distribuée en open-source dans sa version actuelle. La conception et le développement de NetLogo ont été effectués dans le but de modéliser et de simuler des systèmes complexes évoluant dans le temps. NetLogo a été utilisé pour concevoir et mettre en œuvre plusieurs systèmes complexes dans différents domaines tels que l'économie, la biologie, la physique, la chimie, la psychologie, la dynamique des systèmes et de nombreux autres phénomènes naturels et sociaux.

Les agents mobiles et immobiles peuvent être modélisés par NetLogo. Il traite de plusieurs agents appartenant au même espace physique et peut également inclure des milliers d'agents dans la même simulation. En outre, il existe d'autres capacités, notamment les capacités de liaison d'agent et de réseau, les options bidimensionnelles et tridimensionnelles (2D et 3D) concernant les modes d'affichage, ainsi que la commutation facile entre une exécution en une étape et des simulations continues. Les caractéristiques distinguant NetLogo pourraient être résumées comme suit [101] :

- Une documentation claire et complète, contenant un large et riche ensemble d'exemples (exemples de modèles) concernant différents domaines pour guider et renforcer les objectifs des utilisateurs.
- NetLogo propose une interface utilisateur graphique (GUI) facile à comprendre, grâce à laquelle un ensemble d'outils simples est disponible pour produire un modèle, le superviser et contrôler son comportement.
- Cette plateforme offre un langage de modélisation facile à apprendre, puissant et flexible pour la création de modèles.
- Un outil de simulation participatif HubNet permettant à plusieurs utilisateurs exécutant des programmes clients distincts d'interagir via un modèle NetLogo. En outre, il contient plusieurs outils facilitant l'exécution de plusieurs programmes en arrière-plan.
- L'espace comportemental, un outil qui facilite l'exécution d'un modèle à plusieurs reprises grâce à l'utilisation de différentes entrées pour étudier les impacts de scénarios alternatifs.
- Éditeurs d'icônes et possibilités d'importation de liens et d'agents.
- Un environnement distinct doté d'outils appropriés pour développer graphiquement des modèles de dynamique de système plutôt que des modèles basés sur des agents.

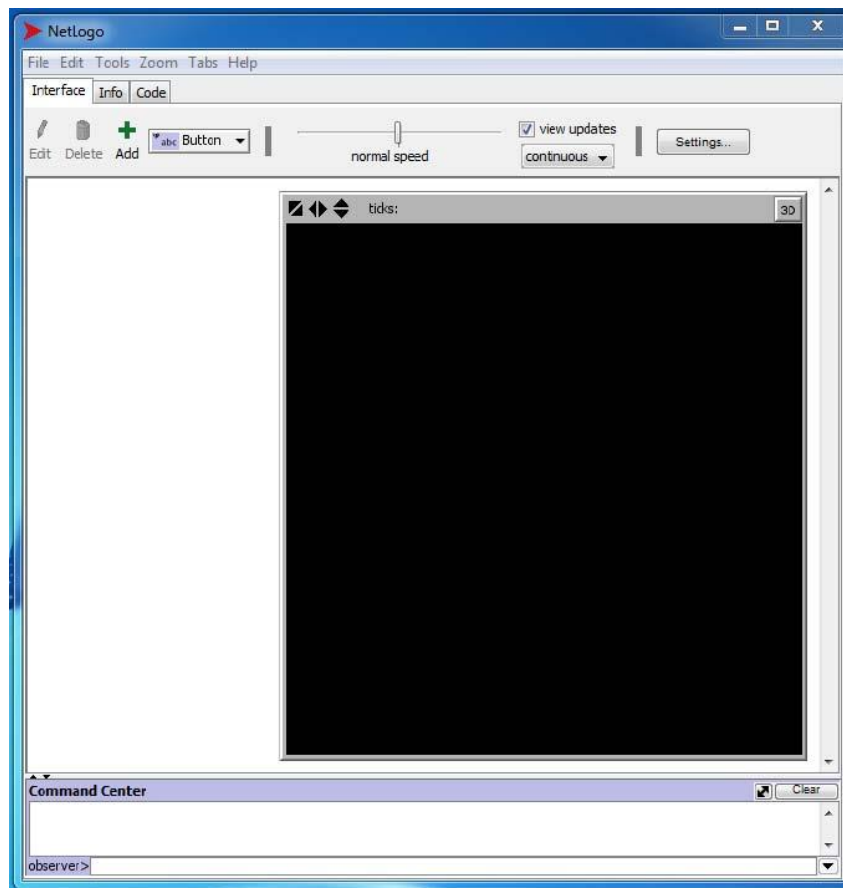


Figure 4-3 : Interface NetLogo

Afin d'étudier l'application de NetLogo concernant la modélisation et la simulation de réseaux ad hoc mobiles simulant en particulier des phénomènes plus élevés de réseaux ad hoc mobiles, les auteurs dans [102] ont analysé le NetLogo du langage de programmation et déterminé plusieurs caractéristiques et supériorités de ce dernier par rapport aux simulateurs de réseau classiques. Ils ont réalisé la conception et la mise en œuvre d'un modèle réaliste de réseau ad hoc mobile dans NetLogo, puis ils ont prouvé son applicabilité à la modélisation et à la simulation de réseaux ad hoc mobiles.

En outre, grâce à des recherches, ils ont conclu dans [103] que NetLogo était capable de simuler efficacement les comportements de l'agriculteur. Il est prouvé dans un cas que lorsque les subventions financières augmentent considérablement le revenu des agriculteurs, la proportion de celles qui enfreignent les politiques diminuera considérablement, et dans l'autre cas, lorsque les subventions financières réduiront largement le revenu des agriculteurs, la proportion de ceux qui suivent les politiques va considérablement diminuer.

Dans les activités de recherche récentes [104], NetLogo a été utilisé pour étudier l'impact d'une optimisation de colonies de fourmis sur le problème du trafic, le problème du contrôle du trafic étant modélisé comme un système multi-agents-multi-usages (MAMP).

Au cours de notre travail, nous avons constaté l'utilité d'utiliser NetLogo dans le but de décrire l'environnement de poursuite-évasion ainsi que le processus de poursuite des coalitions d'agents jusqu'à la capture des évadés détectés. En fait, cette plate-forme offre la possibilité de créer différents types d'agents qui pourraient être distingués par plusieurs caractéristiques telles que le degré de mobilité définissant la vitesse du mouvement et les chemins à suivre disponibles, la forme et la couleur définissant le rôle attribué au cours du processus de la poursuite.

Un programme NetLogo peut être composé de différents types d'agents :

- les tortues (Turtles) : ce sont nos agents individuels, ils ont des caractéristiques, ils se déplacent dans le monde, ils naissent et meurent ; ils peuvent être de plusieurs types (Breeds, consommateurs, firmes)
- les patches : ce sont les composantes spatiales du « monde » qui est modélisé ; les agents se déplacent sur les patches et les patches peuvent stocker des variables. En d'autres termes, chaque patch caractérisé par des coordonnées cartésiennes est considéré comme une cellule pouvant être occupée par un agent ou un obstacle. Les couleurs des patches sont modifiables nous permettant de définir les cellules libres, les obstacles, ainsi que les limites de l'environnement.
- les liens (links) : un type particulier d'agent qui relie deux agents et qui est représenté comme une ligne dessinée entre ces agents. Ce lien peut être orienté ou non
- le monde (World) : il correspond à la représentation spatiale (en 2d ou 3D) de l'environnement modélisé.

NetLogo est la prochaine génération de la série de langages de modélisation multi-agents, notamment StarLogo et StarLogoT. NetLogo fonctionne sur la machine virtuelle Java et fonctionne donc sur toutes les principales plates-formes (Mac, Windows, Linux, etc.). Il est exécuté comme une application de bureau. Le fonctionnement en ligne de commande est également pris en charge.

4.3 Expériences de simulation

Afin d'évaluer l'approche présentée dans ce chapitre, nous réalisons notre jeu de poursuite-évasion sur un exemple prenant place dans une grille rectangulaire à deux dimensions de 100×100 cellules. En ce qui concerne les agents environnementaux, nos simulations sont basées sur dix (10) poursuivants et deux (02) évadés de type $Re = IV$ (quatre poursuivants pour être capturé) comme le montre la Figure 4-4. Il est expliqué en détail comment capturer un poursuivant de ce type. Chaque agent est marqué avec un numéro d'identification. Les poursuivants et les évadés ont une vitesse similaire (une cellule par itération). Les équipes des poursuivants sont tout à fait capables de déterminer leurs positions réelles et les évadés ont disparu après la capture. Si la capture de l'évadé est effectuée, la coalition créée pour améliorer cette poursuite sera automatiquement dissoute.

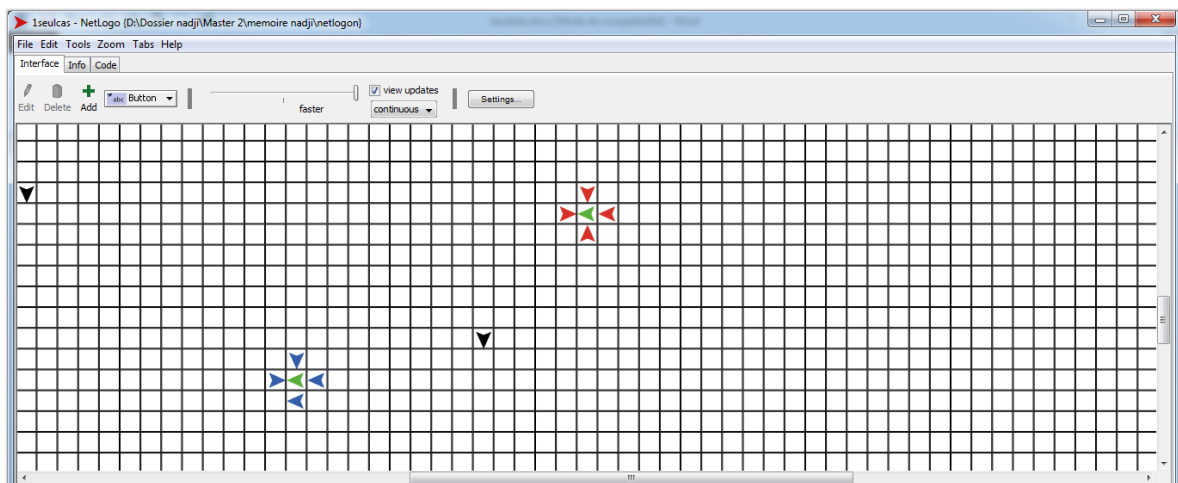


Figure 4-4 : Exemple : 02 évadés de type $Re = IV$ après la capture

Le Tableau 4-1 Figure 4-1 reprend les résultats obtenus après l'application du calcul décentralisé des coalitions possibles sur cette étude de cas.

Dans ce cas et selon l'équation (3-3), les coalitions générales possibles (Ω) sont égales à 45 coalitions, qui seront réparties équitablement entre les agents jouant le rôle de poursuivant, comme indiqué dans le Tableau 4-1. Chaque coalition générale génère un certain nombre de coalitions (= 70) selon l'équation (3-5).

$$\Omega = \frac{n!}{(n-\lambda)! \lambda!} = \frac{10!}{(10-8)! 8!} = \frac{3628800}{80640} = 45 \text{ Coalitions générale}$$

$$\lambda = (Re_1 + Re_2 + \dots + Re_N) = 4 + 4 = 8$$

Selon l'équation (3-5) :

$$\mathfrak{U} = \prod_{j=1}^N \frac{(\lambda - \sum_{k=0}^{k=j-1} Re_k)!}{(\lambda - \sum_{k=0}^{k=j} Re_k)! Re_j} = \frac{8!}{4!4!} * \frac{4!}{(8-8)!4!} = 70 \text{ Coalitions}$$

Tableau 4-1 : Répartition du calcul des coalitions possibles

Les poursuivants	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Coalitions générales	5	5	5	5	5	4	4	4	4	4
Coalitions possibles générées	350	350	350	350	350	280	280	280	280	280

Donc, dans notre étude de cas on a $\text{Nbr}_{cl} = \Omega \times \mathfrak{U} = 45 \times 70 = 3150$ coalitions possibles.

De plus, nous avons étudié le nombre de coalitions possibles générées en parallèle par les poursuivants par rapport au nombre de poursuivants existants, comme le montre dans le Tableau 4-2 par rapport à la méthode centralisée dans laquelle un seul poursuivant calcule les coalitions possibles, la méthode décentralisée diminue considérablement le temps concernant ce calcul par sa division sur le nombre des poursuivants existants.

$$\Omega = 5 + 5 + 5 + 5 + 5 + 4 + 4 + 4 + 4 + 4 = 45 \text{ coalitions générale}$$

$$\mathfrak{U} = 350 + 350 + 350 + 350 + 350 + 280 + 280 + 280 + 280 + 280 = 3150 \text{ coalitions}$$

Les résultats de 20 épisodes de notre simulation sont montrés dans le tableau suivant :

Tableau 4-2 : résultats de la poursuite

IEDS		IEDSEPC	
Temps de capture (Itération)	Changement de rôle (Stabilité)	Temps de capture (Itération)	Changement de rôle (Stabilité)
95	30	91	14
81	10	90	6
96	12	82	10
133	49	111	30
84	12	78	8
132	81	120	27
113	30	107	8
105	61	108	26
98	79	97	32
113	136	106	17
92	25	76	18
86	14	85	21
120	57	102	6
105	47	76	17
91	43	93	24
98	92	94	10
93	18	84	14
82	22	82	16
96	20	95	8
107	32	94	10

Le Tableau 4-2 résume les principaux résultats obtenus, nous en déduisons que l'algorithme de poursuite basé sur l'IEDSEPC (Élimination des stratégies dominées avec changement des priorités) est meilleur que l'algorithme basé sur le modèle IEDS (Élimination des stratégies dominées) en ce qui concerne le temps de capture ainsi que le changement de rôle (stabilité) des poursuivants. La cause principale de ce fait est le dynamisme de nos groupes de coalition. Ce mécanisme flexible améliore l'intelligence des poursuivants concernant les déplacements et l'acquisition des récompenses, sachant que la récompense de l'équipe est optimale dans le cas où chaque poursuivant emprunte le meilleur chemin.

Les résultats présentés à la Figure 4-5 représentent le temps de capture réalisé au cours de vingt (20) études de cas de simulation différentes (épisodes) du début à la fin de chacune.

Afin de mettre en évidence la différence entre les deux différents cas, nous avons vu l'utilité de prendre en compte le concept d'itération, qui détermine le nombre de changements d'état concernant chaque agent au cours des poursuites.

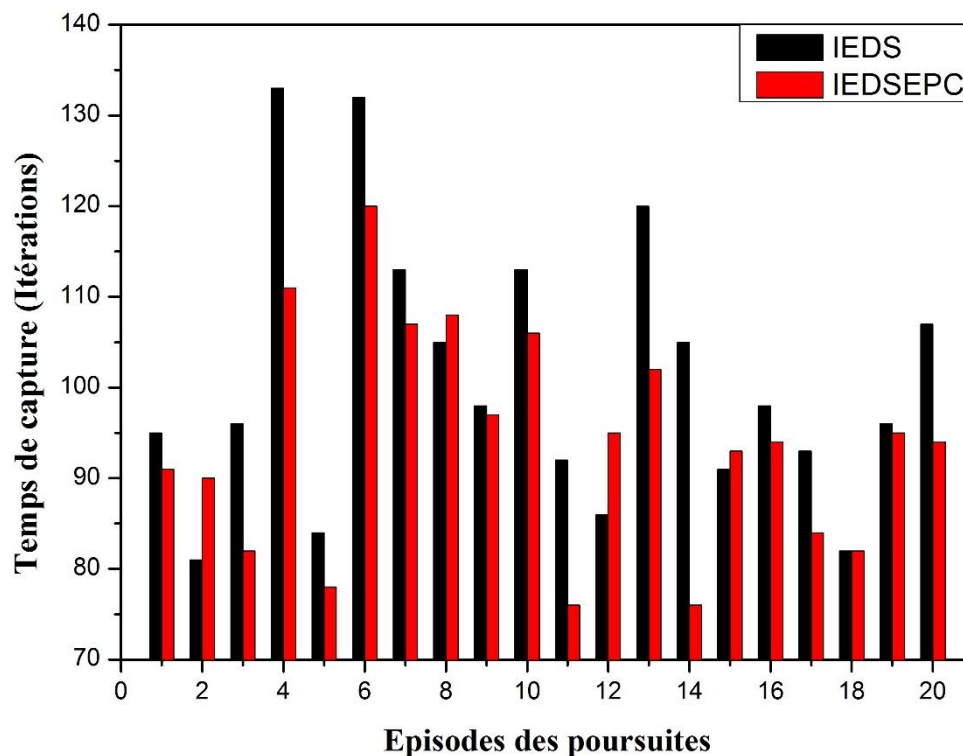


Figure 4-5 : Temps de capture (Itérations) après (20) épisodes des poursuites

Dans le premier cas (IEDS), le temps de capture moyen obtenu est égal à 101 itérations. En outre, nous notons une diminution intéressante jusqu'à 93.55 itérations après l'application de l'algorithme de formation de coalition IEDSEPC avec un choix de 03 changements de priorités, sachant que on a ($10! = 3628800$ choix possibles).

Les résultats de développement des récompenses des poursuivants par l'utilisation du model IEDS et IEDSEPC de la même période de la poursuite :

4.3.1 Utilisation du model IEDS

- **Le nombre d'itération :** Observer : 105
- **Listes des récompenses :**

4.3.2 Utilisation du model IEDSEPC avec 03 changement de priorités

- **Le nombre d'itération :** Observer : 89
- **Listes des récompenses :**

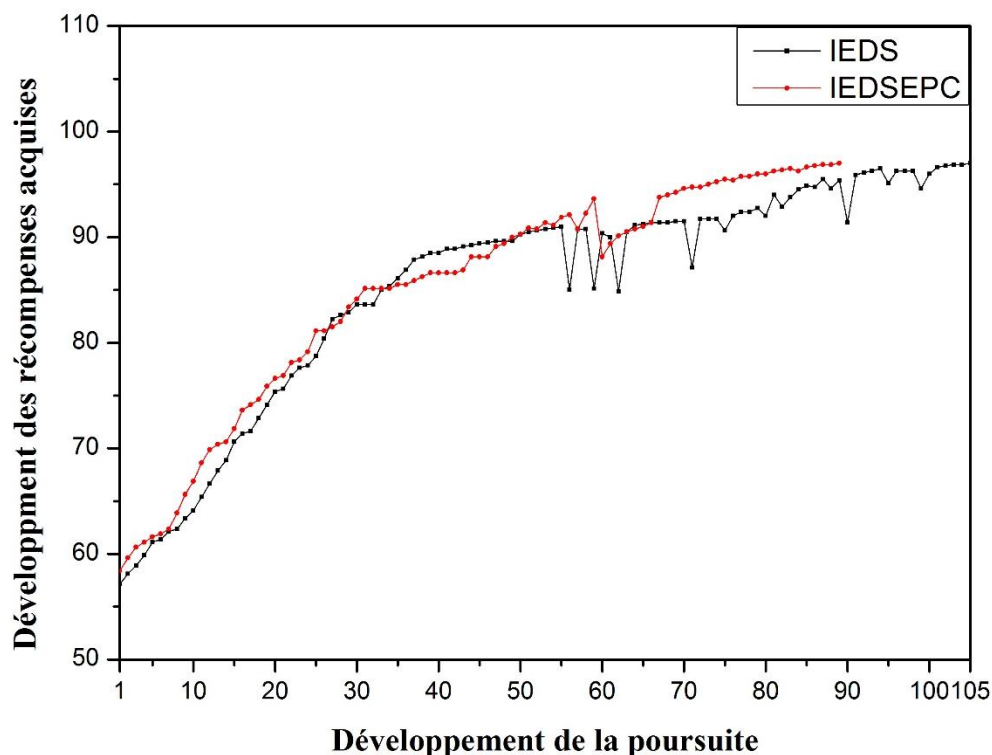


Figure 4-6 : Le développement des récompenses des poursuivants

La Figure 4-6 montre l'évolution de la fonction de récompense des poursuivants au cours de la même période de poursuite des deux cas (IEDS et IEDEPC), et les résultats reflètent l'amélioration apportée par les formations dynamiques et les reformatations des équipes de poursuite. Enfin, nous nous sommes concentrés sur l'étude des récompenses moyennes des poursuivants obtenues dans chaque cas d'itération de poursuite au cours de la poursuite complète. Nous remarquons que le maximum de récompenses est acquis durant l'itération 89 dans le cas IEDSEPC par rapport au cas IEDS ou le maximum de récompense est acquis durant l'itération 105.

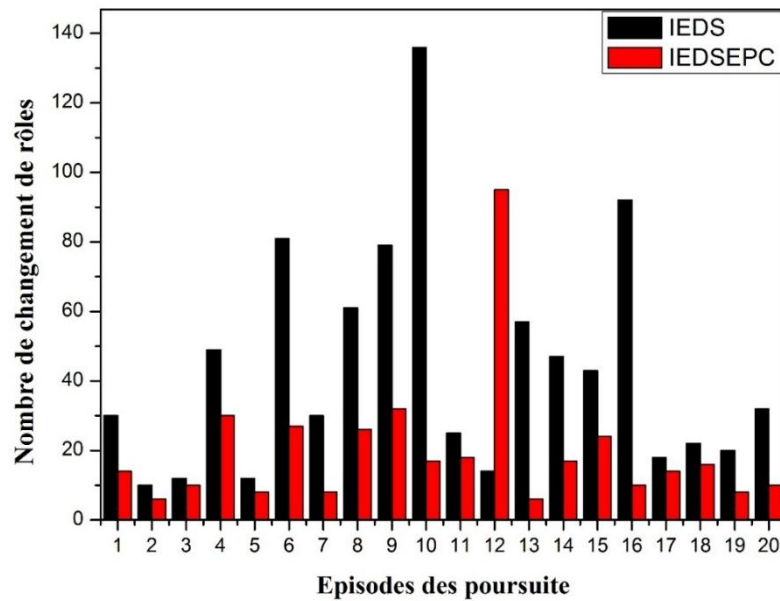


Figure 4-7 : Nombre de changement de rôle après (20) épisodes des poursuites

Les résultats présentés à la Figure 4-7 représentent le nombre de changement de rôle des poursuivants suivant les trois (03) stratégies (S_1, S_2, S_3) au cours de vingt (20) études de cas de simulation différentes (épisodes). Avec le model IEDSEPC on atteint une certaine stabilité (moyenne est égale à 16.1 changements) des poursuivants dans leurs groupe durant la poursuite meilleure que le model IEDS (moyenne est égale à 43.5 changements). Ces moyennes prouvent que la nouvelle méthode proposée procure une certaine stabilité aux groupes de poursuite. Ce fait influence positivement le temps de capture ainsi que l'acquisition des récompenses durant l'exécution des tâches.

Tableau 4-3 : moyen de résultat de la poursuite

	IEDS	IEDSEPC
Temps moyen de capture (itération)	101	93.55
Moyen de Changement de rôle (Stabilité)	43.5	16.1

4.4 Conclusion

Dans ce chapitre, nous nous sommes concentrés sur l'implémentation de l'algorithme de formation de coalition IEDSEPC afin de traiter le problème de poursuite-évasion. Pour cela, nous avons utilisé la plateforme NETLOGO que nous avons jugé comme la plus apte à faciliter cette implémentation. Pour démontrer l'efficacité de cette nouvelle approche, nous l'avons comparé à un algorithme de formation de coalition (IEDS) récent et basé sur la théorie des jeux.

Durant les simulations, nous nous sommes basés sur le temps de capture, la stabilité des groupes, et ainsi que l'acquisition des récompenses durant la poursuite. Les résultats expliqués dans cette partie de ce chapitre prouvent que cette approche améliore tous les paramètres étudiés.

Conclusion et perspectives

Contenu

Conclusion et perspectives.....	83
I Résumé conclusif.....	84
II Perspectives et tendances futures	84

I Résumé conclusif

L'étude des problèmes de coordination multi-agents du point de vue de la planification des tâches a permis de proposer une nouvelle solution appliqué au problème de poursuite-évasion.

Dans notre approche, nous avons utilisé une méthode de la théorie des jeux pour fournir un mécanisme de coordination des tâches de poursuite. Cet algorithme de coalition repose sur l'élimination itérative des groupes de poursuites dominés avec un changement de priorité d'élimination (IEDSEPC), ce qui crée un certain équilibre entre les groupes de poursuites sélectionnés, ainsi que l'acquisition des récompenses durant la poursuite. En outre, cette approche présente un calcul décentralisé intéressant des formations de coalition possibles.

En ce qui concerne la contribution à la coordination de mouvement des agents, notre approche se concentre sur la décomposition spatiale. L'environnement de poursuite se présente sous la forme d'une grille de cellules dans laquelle les principes du Processus Décisionnel de Markov (MPD) sont mis en œuvre pour guider les trajectoires des agents.

II Perspectives et tendances futures

Dans les travaux futurs, nous allons essayer d'appliquer une méthode d'apprentissage automatique supervisée à notre mécanisme de planification de tâches afin de trouver une sorte d'équilibre entre le degré de stabilité et celui du dynamisme des coalitions formées durant la poursuite.

Bibliographie

- [1] Ferber.J. (1995). Les systèmes multi-agents, vers une intelligence collective. Edition Inter-éditions.
- [2] Ferber.J. (2006). Concepts et méthodologies multi-agents. Dans Modélisation et simulation multi-agents, applications pour les sciences de l'homme et de la société. Edition Lavoisier.
- [3] Ganascia.J.G. (2006). Connaissance. Dans Encyclopédie de l'informatique et des systèmes d'information. Edition Vuibert Informatique.
- [4] Sycara.K. (1998). Multiagent systems. In AI magazine Volume 19, No.2 Intelligent Agents.
- [5] Nguyen.T.D., Thompson.S. (2007). Intelligent Agent Framework for Order Entry and Management. The Sixth Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'07) pp1393-1398.
- [6] Vidal.J.M. (2006). Fundamentals of multiagent systems with NetLogo examples. © 2006.
- [7] El Fellah Seghrouchni.A. (2006). Systèmes multi-agents. Dans Encyclopédie de l'informatique et des systèmes d'information. Edition Vuibert Informatique.
- [8] Jennings.N.R., Sycara,K., Wooldridge.M. (1998). A roadmap of agent research and development. International Journal of Autonomous Agents and Multi-Agent Systems (1)
- [9] Jamont. J.P, (2005), «DIAMOND: Une approche pour la conception de systèmes multi-agents embarqués», thèse de doctorat, INP de Grenoble, France.
- [10] Gleizes. MP, Bernon. C, Migeon. F, & Picard. G, (2008), « Méthodes de développement de systèmes multi-agents », Génie logiciel N° 86.
- [11] Jennings.N.R. (2000). On Agent-Based Software Engineering. Artificial Intelligence Journal 117 (2), pp277-296.
- [12] Russell.S., Norvig.P. (2006). Intelligence artificielle. Pearson Education 2e édition.
- [13] Zambonelli.F., Jennings.N. R., Wooldridge.M. (2001). Organisational rules as an abstraction for the analysis and design of multi-agent systems. Int J. of Software Engineering and Knowledge Engineering 11 (3), pp 303-328.
- [14] Cardon.A., Guessoum.Z. (2000). Systèmes multi-agents adaptatifs. In Actes des Huitièmes Journées Francophones d'Intelligence Artificielle Distribuée et Systèmes Multi-agents. Edition Hermès.
- [15] Jarras.I., Chaib-Draa.B. (2002). Aperçu sur les systèmes multi-agents. Rapport technique, CIRANO.
- [16] Moujahed.S. (2007). Approche multi-agents auto-organisée pour la résolution de contraintes spatiales dans les problèmes de positionnement mono et multi-niveaux. Thèse de doctorat de l'université de Belfort-Montbéliard.

- [17] Ferber.J., Stratulat.T., Tranier.J. (2009). Towards an integral approach of organizations in multi-agent systems, the MASQ approach. In *Multi-agent Systems: Semantics and Dynamics of Organizational Models in Virginia Dignum* (Ed), IGI.
- [18] Gleizes.M.P., (2004). Vers la résolution de problèmes par émergence. Thèse d'habilitation de l'Université de Toulouse.
- [19] Gleizes.M.P., Camps.V., Glize.P. (1999). A theory of emergent computation based on cooperative self-organization for adaptive artificial systems. In *4th European Congress of Systems Science, Valencia*.
- [20] Melaye.D., Demazeau.Y. (2005). Modèles et Réseaux de confiance. Analyse bibliographique. Les cahiers Leibniz. N° 142. ISSN : 1298-020X.
- [21] Deloach.S.A. (2009), Organizational model for adaptive complex systems. In *Virginia Dignum (ed.) Multi-Agent Systems: Semantics and Dynamics of Organizational Models*. IGI Global: Hershey, PA. ISBN : 1-60566-256-9.
- [22] Jiquan.S., Zhiqiang.W. (2008). Service-Oriented Organization Management and Coordination Control of MAS. In *IEEE International Conference on Intelligent Computation Technology and Automation*, pp 479-483.
- [23] Mkadmi.A, Saleh.I. (2008). Bibliothèque numérique et recherche d'informations. Edition Lavoisier.
- [24] Akola.J. and al. (2006). Encyclopédie de l'informatique et des systèmes d'information. Edition Vuibert.
- [25] Rahwan.T., Jennings.N. R. (2007). An algorithm for distributing coalitional value calculations among cooperating agents. In *Artificial Intelligence Journal* 171 (8-9). pp 535-567.
- [26] Rejeb.L. (2005). Simulation multi-agents de modèles économiques Vers des systems multi-agents adaptatifs. Thèse de doctorat de l'Université de Reims.
- [27] Dutta.P.S., Jennings.N.R., Moreau.L. (2005). Cooperative information sharing to improve distributed learning in multi-agent systems. *Journal of Artificial Intelligence Research* 24. pp 407-463.
- [28] Thomas.V. (2005). Proposition d'un formalisme pour la construction automatique d'interactions dans les systèmes multi-agents réactifs. Thèse de doctorat de l'université Nancy1.
- [29] Panait.L., Luke.S. (2005). Cooperative Multi-Agent Learning: The State of the Art. In *Autonomous Agents and Multi-Agent Systems*, 00, pp 1-48, Springer.
- [30] Sabouret.N. (2009). Interactions sur le fonctionnement dans les systèmes multi-agents ouverts et hétérogènes. Habilitation à Diriger des Recherches, Université Pierre & Marie Curie.
- [31] Doran.J. E., Franklin.S., Jennings.N.R., Norman.T.J. (1997). On cooperation in multi-agent systems. *The Knowledge Engineering Review* 12 (3), pp 309-314.
- [32] Velagapudi.P., Prokopyev.O, Sycara.K., Scerri.P. (2007). Maintaining Shared Belief in a Large Multiagent Team. In *Proceedings of FUSION*.

- [33] Kota.R., Gibbins.N. Jennings.N.R. (2009), A generic agent organization framework for autonomic systems. In Proc. 1st Int. Workshop on Agent-Based Social Simulation and Autonomic Systems, Limassol, Cyprus.
- [34] Nwana H. S., Lee L., and Jennings N. R., "Co-ordination in Multi-Agent Systems."Lecture Notes in Computer Science; 1198 (c) Springer-Verlag Berlin Heidelberg 1997.
- [35] G. Weiss, ed., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, Cambridge, Mass., 1999.
- [36] Durfee Edmund H., "Scaling Up Agent Coordination Strategies," University of Michigan. IEE July 2001.
- [37] Huhns M and Singh M. P., "CKBS-94 Tutorial: Distributed Artificial Intelligence for Information Systems." Dake Centre, University of Keele. 1994
- [38] Durfee E. H, Lesser V. R. and Corkill D. D., "Coherent Cooperation among Communicating Problem Solvers", IEE Trans Comput, 11, pp 1275-1291 (1987).
- [39] Durfee E.H. ,Lesser V. R., and Corkill D. D., "Trends in Cooperative Distributed Problem Solving", IEEE Knowledge and Data engineering, 1, No.1, pp 63-83, 1989.
- [40] Werman, K. J: "Knowledge-based model of negotiation using shareable perspectives", Proc of the 10th Int. Workshop on DAI, Texas. 1990.
- [41] Parunak H V-D, "Manufacturing Experience with the Contract Net", in Gasser L and Huhns M (Eds): "Distributed Artificial Intelligence 2", Morgan Kaufmann 1989.
- [42] Sandholm, Tuomas. "An implementation of the contract net protocol based on marginal cost calculations." In Proc. Of the National Confr. On AI, pages 256-262, Washington, D.C., July 1993.
- [43] Sandholm, Tuomas. "Negotiation among self-interested Computationally Limited Agents. "PhD thesis, University of Massachusetts, Amherst, 1996. Available at <http://www.cs.wustl.edu/~sandholm/dissertation.ps>.
- [44] Sandholm, T W., "Contract types for satisficing task allocation: I theoretical results.", In AAAI Spring Symposium Series: Satisficing Models, Pages 68-75, Stanford University, CA, March 1998.
- [45] Camarata S, McArthur D and Steeb R: "Strategies of Cooperation in Distributed Problem Solving", Proc of the 8th Int. Joint Conf on Artificial Intelligence, pp 767-770. 1983
- [46] Durfee E. H., Lesser V. R., "Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation", © IEEE 1991.
- [47] Bussmann S and Muller J., " A Negotiation Framework for cooperating Agents", in Deen S. M. (Ed): Proc CKBS_SIG, Dake Center, University of Keele, pp 1-17. 1992.
- [48] Sycara K: "Multi-agent Compromise via Negotiation", in Gasser L and Huhns M (Eds): "Distributed Artificial Intelligence 2", Morgan Kaufmann 1989.

- [49] Ekaette E.U. “A Survey on uncertain programming and uncertainty management techniques in agent-based systems”, University of Calgary. Aug. 2002.
- [50] Zlotkin G and Rosenschein J. S., “Blocks, Lies and Postal Freight: The Nature of Deception in Behavior”, Proc of the 10th Int Workshop on DAI. 1990.
- [51] M. Wooldridge, N. R. Jennings. Intelligent agents: Theories, Architectures and Languages. Lecture Notes in Artificial Intelligence, 1995, vol 890,
- [52] V. Lesser and D. Corkill. Functionally Accurate, Cooperative Distributed Systems. IEEE Transactions on Systems, Man, and Cybernetics, 1981, vol 11, pp. 81–96.
- [53] Z. Wang, K. Mesghouni, S. Hammadi. Agent-based Coalition Formation in a Co-modal Transport System. IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014, vol 3, pp. 310-317.
- [54] D. Ye, M. Zhang, and D. Sutanto. Decentralised dispatch of distributed energy resources in smart grids via multi-agent coalition formation. Journal of Parallel and Distributed Computing , 2015, vol 83 , pp. 30–43.
- [55] B. Horling and V. Lesser. Quantitative Organizational Models for Large-Scale Agent Systems. Massively Multi-Agent Systems, 2005, pp. 121–135.
- [56] B. Horling, R. Mailler, and V. Lesser. A Case Study of Organizational Effects in a Distributed Sensor Network. In Proceedings of the International Conference on Intelligent Agent Technology, 2004, pp. 51–57.
- [57] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohm. Coalition Structure Generation with Worst Case Guarantees. Artificial Intelligence, 1999, pp. 209-238.
- [58] E. Durfee, V. Lesser, and D. Corkill. Cooperation through Communication in a Distributed Problem-Solving Network. Distributed Artificial Intelligence, 1987, pp. 29–58.
- [59] E. Matson and S. DeLoach. Engineering Organization-Based Multiagent Systems. Software Engineering for Multi-Agent Systems IV, 2003, vol 3914, pp. 109-125.
- [60] G. Chalkiadakis, C. Boutilier. Sequentially optimal repeated coalition formation under uncertainty. Autonomous Agents and Multi-Agent Systems , 2012, vol 24, pp. 441-484.
- [61] W. Hao, D. Lei, F. Baofu, and Y. Hongliang. Pursuers-Coalition Construction Algorithm in Multi-robot Pursuit-Evasion Game. ROBOT, 2013, pp. 142-150.
- [62] Y. So and E. Durfee. Designing Organizations for Computational Agents. In Simulating Organizations: Computational Models of Institutions and Groups, MIT Press, 1998, pp. 47–64.
- [63] Yves Demazeau and A.C. Rocha Costa. Populations and Organisations in Open Multi-Agent Systems, 1st Symposium on Parallel and Distributed AI, Hyderabad, India, July 1996.
- [64] Eric Malville L’auto-organisation de groupes pour l’allocation de tâches dans les Systèmes Multi-Agents : Application à CORBA ; Thèse 1999 Université SAVOIE France.

- [65] Olivier Gutknecht Proposition d'un modèle organisationnel générique de systèmes multi-agents Examen de ses conséquences formelles, implémentatoires et méthodologiques ; Thèse 2001 Université de Montpellier II France.
- [66] Ferber J., Gutknecht O., A Meta-Model for the Analysis and Design of Organizations in Multi-Agents Systems, 1998 <http://www.madkit.org>
- [67]. S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1-2):79–98, July 1997.
- [68]. J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behaviour*.
- [69]. P. Noriega and C. Sierra, editors. *Agent Mediated Electronic Commerce (LNAI Volume 1571)*. Springer-Verlag: Berlin, Germany, 1999.
- [70] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley: Reading, MA, 1994. Princeton University Press, 1944.
- [71] J. S. Rosenschein. *Rational Interaction: Cooperation among Intelligent Agents*. PhD thesis, Computer Science Department, Stanford University, Stanford, CA 94305, 1985.
- [72] J. S. Rosenschein and M. R. Genesereth. Deals among rational agents. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 91–99, Los Angeles, CA, 1985.
- [73] J. S. Rosenschein, M. Ginsberg, and M. R. Genesereth. Cooperation without communication. In *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, Philadelphia, PA, 1986.
- [74] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. The MIT Press: Cambridge, MA, 1994. 250. S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995.
- [75]. T. Sandholm. Distributed rational decision-making. In G. Weiss, editor, *Multiagent Systems*, pages 201–258. The MIT Press: Cambridge, MA, 1999.
- [76] J. S. Rosenschein, *Rational interaction: cooperation among intelligent agents*. PhD Thesis, Stanford University, 1985.
- [77] S. Kraus and J. Wilkenfield. The function of time in cooperative negotiations, Preliminary report, Department of Computer Science, University of Maryland, 1991.
- [78] T. W. Sandholm. Distributed rational decision-making. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999, pp. 45-64.
- [79] G. Zlotkin, and J. S Rosenschein. Blocks, lies, and postal freight: the nature of deception in behaviour. *Proceeding of the 10th International Workshop on Distributed Artificial Intelligence*, 1990.
- [80] A. Champion, R. Mandiau, S. Espié, and C. Kolski. Multi-Agent Road Traffic Simulation: Towards Coordination by Game Theory Based Mechanism. *Proceedings of the Agents in Traffic and Transportation, workshop in conjunction with the ITS World Congress*, 2001.

- [81] N. Xiao, X. Wang, L. Xie, T. Wongpiromsarn, E. Frazzoli, and D. Rus. Road Pricing Design Based on Game Theory and Multi-agent Consensus. *IEEE/CAA Journal of Automatica Sinica*, 2014, vol 1, pp. 31 - 39.
- [82] A. J. C. Trappey, C. V. Trappey, and W. Ni. A multi-agent collaborative maintenance platform applying game theory negotiation strategies. *Journal of Intelligent Manufacturing* , 2013, vol 24, pp. 613-623.
- [83] M. Savall et al., "YAMAM – An Organisation Model for the Multiagent Systems", Implementation in the Phoenix platform, 3rd Francophone Conference of Modeling and Simulation "Conception, Analyze, Management of Industrial Systems" MOSIM, 2001, Troyes, France.
- [84] J. Ferber et al., "From Agents to Organizations, an Organizational View of Multi-Agent Systems", *Agent Oriented Software Engineering (AOSE) IV*, P., vol. 2935, pp. 214–230, 2004.
- [85] M. Souidi et al., "Multi-agent Cooperation Pursuit Based on an Extension of AALAADIN Organisational Model, *Journal of Experimental & Theoretical Artificial Intelligence (JETAI)*, vol. 28, pp. 1075–1088, 2016.
- [86] B.Khosravifar, F. Bouchet, R. Feyzi-Behnagh, R. Azevedo, and J. M. Harley, "Using intelligent multi-agent systems to model and foster self-regulated learning: a theoretically-based approach using Markov decision process," in *Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA '13)*, pp. 413–420, IEEE, Barcelona, Spain, March 2013.
- [87] L. Ting, Z. Cheng, and Z. Weiming, "Planning for target system striking based on Markov decision process," in *Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI '13)*, pp. 154–159, IEEE, Dongguan, China, July 2013.
- [88] Treur J., Report of the First SIG Meeting, Contribution to Methodologies/software engineering SIG, Rapport, First SIG Meeting – AgentLink, 1998.
- [89] Jennings N. R., « Agent-Oriented Software Engineering », Garijo F. J., Boman M. (dir.), *Multi-Agent System Engineering, 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99*, Valence, Espagne, juin-juillet 1999, vol. 1647 de LNAI, Springer, 1999.
- [90] Ferber J., Gutknecht O., « Aalaadin : a meta-model for the analysis and design of organizations in multi-agent systems », Demazeau, Y. Ed., *3rd International Conference on Multi-Agent Systems*, Paris, IEEE, p. 128-135, 1998.
- [91] Rao A. S., Georgeff M. P., *BDI-Agents: from Theory to Practice*, Rapport n° 56, Australian Artificial Intelligence Institute, Melbourne, Australie, 1995.
- [92] El Fallah Seghrouchni A., Suna A., « CLAIM : Un langage de programmation pour des agents autonomes, intelligents et mobiles », *Technique et Science Informatiques*, vol. 22, n° 4, p. 91-105, 2003.
- [93] Bordini R., Braubach L., Dastani M., El Fallah Seghrouchni A., Gomez-Sanz J., Leite J., O'Hare G., Pokahr A., Ricci A., «A Survey of Programming Languages and Platforms for Multi-

Agent Systems », Informatica, An International Journal of Computing and Informatics, vol.30, n° 1, p. 33-44, 2006.

[94] ZIMMERMANN C., Ed., Advances in object-oriented meta-level architectures and reflection, CRC, 1996.

[95] P.-M. Ricordel. Programmation Orientee Multi-Agents : Developpement et Déploiement de Systèmes Multi-Agents Voyelles.

[96] Bellifemine F., Giovani C., Tiziana T. Rimassa G., "Jade Programmer's Guide"(<http://sharon.csel.it/projects/jade/>), 2000.

[97] Bellifemine F., Poggi A., Rimassa G., "JADE -- A FIPA-compliant agent framework", 1999.

[98] Bellifemine F., Caire G., Trucco T., Rimassa G., "JADE : Programmer's Guide", 2002.

[99] U. Wilensky. NetLogo.Computer software. Northwestern University, 1999.

[100] Richard Rashid, Robert Baron, Alessandro Forin, David Golub, Michael Jones, Daniel Julin, Douglas Orr, and Richard Sanzi. Mach : A foundation for Open Systems. In Proceedings of the 34th Computer Society Ithe Second Workshop on Workstation Operating Systems(WWOS2), September 1989.

[101] D. Stigberg. An Introduction to the NetLogo Modeling Environment. Ecologist-Developed Spatially-Explicit Dynamic Landscape Models, Part of the series Modeling Dynamic Systems, 2012, pp. 27-41.

[102] M. Babiš, and P. Magula.NetLogo – an alternative way of simulating mobile ad hoc networks. Wireless and Mobile Networking Conference (WMNC), 2012, pp. 122–125.

[103] S. Xike, and L. Fengxian. An Agent-based Simulation of Farmer's Behaviors Adapting to Ecological Policies Using NetLogo. International Symposium on Information Science and Engineering (ISISE), 2012, pp. 426–429.

[104] K. Jerry, K. Yujun, O. Kwasi, Z. Enzhan, and T. Parfait. NetLogo implementation of an ant colony optimisation solution to the traffic problem. IET Intelligent Transport Systems, 2015, vol 9, pp. 862–869.