Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Planification Distribuée dans les Systèmes Multi-agents à l'aide de Processus Décisionnels de Markov

THÈSE

présentée et soutenue publiquement le Une date

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Iadine CHADÈS

Composition du jury

Rapporteurs :	R. Alami	Directeur de recherche, CNRS, Laas
	R. Schott	Professeur, Université Henri Poincaré Nancy 1
	R. Washington	Associate Director, RIACS, NASA
Examinateurs :	J-L. Deneubourg	Chercheur, Université Libre de Bruxelles
	A. Koukam	Professeur, Université de Technologie de Belfort-Montbéliard
Directeur de thèse :	F. Charpillet	Directeur de recherche INRIA, Nancy

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thloria.

Remerciements

Les remerciements.

Table des matières

Table des figures

Liste des Algorithmes

Introduction

L'Intelligence Artificielle (IA) s'est donnée pour objectifs la compréhension et l'adaptation des mécanismes inhérents aux comportements d'entités "intelligentes" pour les appliquer aux systèmes informatiques. Elle peut être présentée comme la discipline qui tente de construire des "systèmes informatiques intelligents".

A l'origine, les recherches pour concevoir des systèmes intelligents se sont concentrées sur la représentation du langage et le développement de l'Intelligence Artificielle symbolique. Mais les limites de l'IA symbolique pour représenter l'information (problèmes d'ancrage) et ses difficultés à s'adapter à la dynamique de la réalité, ont éveillé l'intérêt des chercheurs pour de nouvelles approches, et en particulier l'approche "agents". Un *agent* est une entité dotée de capacités de perception, de prise de décision et d'action. En y intégrant le développement des réseaux et du parallélisme, l'étude des systèmes distribués en IA est devenue un domaine de recherche à part entière.

Aujourd'hui, l'intelligence artificielle distribuée se préoccupe de construire des systèmes dans lesquels des agents (semi-) autonomes interagissent entre eux comme avec leur environnement. Dans le cadre des systèmes multi-agents (SMA), de tels systèmes sont caractérisés par la fréquente absence de perspectives globales, les agents ne possédant qu'une connaissance partielle sur le problème à résoudre et sa solution. La distribution du contrôle en est la principale caractéristique : une seule entité n'a pas le contrôle des autres. Le terme *localité* résume cette propriété de fonctionnement.

Qu'ils soient ou non inspirés des comportements biologiques, les systèmes multi-agents mettent en jeu des phénomènes de coopération pour la résolution collective d'une tâche. La *coordination* d'actions est un moyen de réaliser cette coopération. Les techniques comme la planification d'actions permettent l'amélioration des performances par rapport à un système dépourvu de coordination.

La coordination prend différentes formes selon les caractéristiques du système multi-agents étudié. On distingue les systèmes multi-agents cognitifs qui font appels à des agents "intelligents" dotés de compétences cognitives (coordination par synchronisation, planification, réglementation) et les systèmes multi-agents réactifs qui sont conçus à partir d'agents minimalistes (coordination réactive) dont un des avantages est la robustesse à l'évolution de l'environnement. Dans ces deux cas, les interactions de coopération entre les agents tiennent une place prépondérante et font que le système parvient à résoudre la tâche qu'on lui a assigné.

Souvent ignorée, l'*incertitude* de fonctionnement d'un système multi-agents est omniprésente dans des applications réelles et notamment dans le cas de perceptions localisées. La théorie de

la décision apporte des outils mathématiques de planification réactive qui prennent en considération cette incertitude, leurs relatives simplicité de résolution s'opposent à la complexité des mécanismes mis en œuvre par les méthodes de planification traditionnelles en IA. Les modèles décisionnels de Markov offrent ainsi un formalisme pour la représentation mathématique d'un problème de prise de décision dans l'incertain. Dans le cas d'une observabilité complète, la résolution d'un Processus Décisionnel de Markov (MDP[1]) se traduit par la conception d'un plan réactif optimal qui, utilisé par un agent, associera à chacune de ses perceptions une décision (ou action) en prenant en compte les incertitudes des actions.

Le sujet de cette thèse se situe à l'intersection de ces deux axes de recherches : les systèmes multi-agents et la théorie de la décision.

1 Thèse soutenue

Dans ce contexte générale, notre intérêt se porte sur les systèmes multi-agents coopératifs dont les agents ont à l'exécution un comportement réactif (absence de communication) et qui sont soumis aux conséquences probabilistes de leurs actions (incertitude).

Les systèmes multi-agents possédant ces caractéristiques réactives sont souvent conçus de manière empirique et ascendante. La démarche ascendante consiste dans un premier temps à concevoir le système multi-agents, puis d'adapter les paramètres afin de satisfaire le fonctionnement recherché. A cette approche ascendante, nous opposons l'approche descendante. La question devient alors : connaissant les caractéristiques d'un problème, comment concevoir le système multi-agents qui saura le résoudre ? C'est à cette interrogation que nous nous proposons de répondre dans cette thèse en utilisant un modèle décisionnel de Markov qui, en plus d'être adapté à la prise de décision dans l'incertain, nous apportera un formalisme théorique sans lequel il nous paraît vain d'étudier et de renouveler avec précision la qualité des agents ainsi conçus.

Les MDP sont dédiés à la formalisation de problèmes centralisés et complètement observables. Or, les SMA sont par essence décentralisés et ont des perceptions locales et souvent incomplètes de leur environnement. Du point de vue de la théorie de la décision, coordonner nos agents à travers l'élaboration de plans (politiques) individuels, c'est résoudre un problème de type DEC-POMDP dont la complexité est NEXP-complet dans le cas le plus favorable. Afin de contourner cette difficulté, nous proposons un ensemble d'algorithmes de conception d'agents fondé sur deux propriétés fondamentales que nous prêtons à nos agents : la *subjectivité* et l'*empathie*. La subjectivité prend en compte la localité des perceptions et des actions, tandis que l'empathie, définie comme la faculté de s'identifier à une personne et de ressentir ce qu'elle ressent, est ici utilisée pour coordonner les actions de nos agents par planification réactive.

2 Organisation du manuscrit

La présentation de notre travail nécessite tout d'abord de décrire les deux domaines de recherche que sont les systèmes multi-agents et les modèles décisionnels de Markov afin de mieux situer notre démarche scientifique.

[1] Acronyme de la traduction anglaise "Markov Decision Process"

Dans le premier chapitre, nous présenterons un état de l'art sur les systèmes multi-agents qui s'articule tout d'abord autour de l'analyse de définitions et de concepts sur lesquels cette discipline nouvelle repose. Puis, nous situerons la coopération et les différentes méthodes de coordination qui sont des axes de recherche importants du domaine. Tandis que les SMA réactifs favorisent l'émergence de phénomènes tels que la coordination réactive et l'auto-organisation, les SMA cognitifs s'emploient, quant à eux, à élaborer des agents intelligents se coordonnant de manière explicite (par exemple par planification). Enfin, à titre de comparaison sur une conception ascendante de systèmes multi-agents, nous présenterons le principe des systèmes multi-agents réactifs d'inspirations biologiques, puis nous étudierons lors d'expérimentations l'influence de l'incertitude sur les performances d'un SMA réactif qui utilise des interactions de type attraction et répulsion.

Le deuxième chapitre présentera les modèles décisionnels de Markov. Nous nous intéresserons alors aux processus décisionnels de Markov en tant qu'outil de conception d'un système multi-agents. Un processus décisionnel de Markov partiellement observable correspond à un modèle d'agent ayant des capacités d'observation incomplète. C'est le cas des agents qui constituent nos systèmes multi-agents. Nous verrons que de par leur complexité de résolution, nous n'avons pas choisi d'utiliser les processus décisionnels de Markov partiellement observables. De plus, si nous considérons le problème dans son ensemble, trouver les politiques individuelles des agents équivaut à résoudre un DEC-POMDP (Processus Décisionnel de Markov Décentralisé Partiellement Observable). Il a été montré que la complexité de sa résolution appartient à la classe NEXP-Complet lorsque le nombre d'agents est supérieur ou égale à deux. D'autres modèles décisionnels de Markov prennent en considération le contrôle décentralisé des systèmes multi-agents sous diverses hypothèses. Nous analyserons les avantages et inconvénients de chacun.

Le chapitre trois constituera notre première étude de modélisation d'un problème réel de nature non "Markovienne" dans un cas mono-agent. Nous présenterons un exemple de modélisation du comportement de gestion de ressource d'un agent situé dans un environnement complexe. Cette étude met en avant la gestion de l'énergie d'une araignée orbitèle (à toile géométrique) dont l'objectif est de maximiser son succès reproducteur. Pour cela, il lui faut gérer la construction successive de toiles qui lui permettent de capturer des proies et d'augmenter ses gains énergétiques. Cependant, le tissage d'une toile n'est pas sans conséquences sur l'état énergétique de l'araignée, cette action entraîne des dépenses énergétiques et l'expose à d'éventuels prédateurs. Nous verrons que bien que le comportement réel de l'araignée ne soit pas de nature "Markovienne", son approximation par l'utilisation d'un processus décisionnel de Markov apporte de bons résultats, validés par la simulation et l'interprétation biologique.

A l'issu de nos expérimentations dans un cas mono-agent, le quatrième chapitre exposera notre modèle formel développé pour la construction de systèmes multi-agents réactifs dédiés à une évolution dans un environnement complexe (incertitude), en utilisant des plans individuels. Dans ce chapitre, nous suivrons progressivement notre cheminement dans la recherche d'algorithmes de conception de nos systèmes multi-agents. Nous définirons les systèmes multi-agents que nous proposons de concevoir. Puis, nous nous intéresserons tour à tour aux deux propriétés sur lesquelles reposent notre approche : la **subjectivité** synonyme de respect de la localité et l'**empathie** synonyme d'adaptation, de prédiction et de coordination vis-à-vis de ses congénères. Enfin, prenant notre problème dans son ensemble, nous proposerons successivement des méthodes de conception centralisées ou décentralisées, dédiées à une population homogène ou hétérogène d'agents, et tirant profit de nos deux propriétés.

Le chapitre cinq validera notre approche de manière expérimentale. Nous présenterons l'application d'un de nos algorithmes sur un exemple académique : la poursuite de proie. Cet algorithme est de loin le plus facile à appliquer en terme de complexité mais le moins performant de par ses approximations. Cependant, nos algorithmes de conception d'agents proposés dans le chapitre précédent appartenant tous à la même famille, les résultats mettront en valeur leur fonctionnement.

Le chapitre six conclura cette thèse. Il résumera les apports et les perspectives qui sont les miennes à travers la réalisation de ce travail. Nous mettrons en valeur les résultats encourageants obtenus ainsi que les possibilités d'exploitation des résultats théoriques déjà obtenus. Enfin, nous terminerons ce manuscrit par des conclusions sur ce travail de conception descendante d'un système multi-agents coopératif dans un environnement complexe à l'aide de processus décisionnels de Markov.

Chapitre 1

Systèmes multi-agents

Les problèmes réels font de plus en plus appel à une diversité de systèmes logiciels qui se doivent de coopérer afin de résoudre les tâches qui leur sont confiées. Les systèmes multi-agents apparaissent alors comme une alternative de choix parmi les techniques traditionnelles de l'IA, et pallient les limites théoriques et applicatives des techniques centralisées. Apparus tout d'abord comme un sous-domaine de recherche de l'IA Distribuée, les systèmes multi-agents sont maintenant un domaine de recherche à part entière, qui multiplie les connexions avec d'autres axes de recherches, tels que les réseaux, la recherche opérationnelle, la biologie du comportement, les sciences cognitives, etc. Dans l'introduction de son ouvrage [Wooldridge, 2002], Wooldridge va plus loin, et évoque l'IA comme un sous-domaine des systèmes multi-agents : l'Intelligence Artificielle tente de concevoir un agent "intelligent" tandis que les systèmes multi-agents s'intéressent à "l'intelligence" d'un ensemble d'agents. Les systèmes multi-agents sont aujourd'hui utilisés dans de nombreux domaines d'applications par exemple les réseaux dynamiques de télécommunication, le commerce électronique ou le contrôle aérien.

Pour comprendre les systèmes multi-agents, il faut éclaircir le vocabulaire et les concepts utilisés qui, dans le cas d'une discipline nouvelle, sont particulièrement riches. Je propose, dans cette thèse, un modèle formel de conception descendante de SMA coopératifs capable de s'adapter à l'incertitude de l'environnement à l'aide d'une méthode de planification réactive. Dans ce chapitre, je m'efforce de présenter les principes fondamentaux des systèmes multi-agents qui me semblent nécessaires afin de situer mon travail dans son contexte de recherche.

Organisation du chapitre

Dans un premier temps, nous présentons et analysons les définitions d'agent et de système multi-agents, de la plus générale à la plus spécialisée. Il existe deux grandes communautés dans les systèmes multi-agents : celle qui s'intéresse à la résolution de problèmes complexes en utilisant des agents cognitifs très élaborés ; et celle qui s'attaque au défi de concevoir des agents minimalistes (réactifs) capables de résoudre collectivement des problèmes de grande complexité mais sans parvenir à en contrôler et à en expliquer le comportement intelligent émergeant.

Cependant, que les agents soient de nature cognitive ou réactive, le problème de la conception constitue une composante majeure des thèmes de recherches se rapportant aux systèmes multi-agents. Comment concevoir un système multi-agents à partir des agents qui le constituent (architecture), des relations (interactions) ou des rôles entre ces agents (organisations) ? Bien sûr, les réponses sont fortement dépendantes des caractéristiques des problèmes que l'on cherche

à résoudre.

Dans la dernière partie de ce chapitre, nous nous intéressons à un exemple de conception ascendante d'un système multi-agents réactif issu des travaux d'intelligence collective. Puis, nous exhibons sur une application académique "la poursuite de proie", les effets de l'incertitude sur les performances d'un système multi-agents réactif en utilisant un modèle de référence [Korf, 1992].

1.1 Agent

Le concept d'agent reste l'objet de recherches dans différentes disciplines de l'IA (dans les systèmes à base de connaissances, la robotique, le langage naturel comme dans d'autres domaines de l'IA), mais aussi dans des disciplines comme la philosophie et la psychologie. Il paraît alors difficile d'accorder toutes les définitions qu'on lui prête.

A titre d'introduction, dans [Franklin et Graesser, 1996], les auteurs publient un état de l'art sur les différentes significations du mot "agent" en informatique et proposent une classification des variétés d'agents existants selon leurs caractéristiques. La figure 1.1 reprend cette tentative de classification. Les agents sont, dans un premier temps, soit réels (agents biologiques, agents robotiques), soit artificiels (agents computationnels).

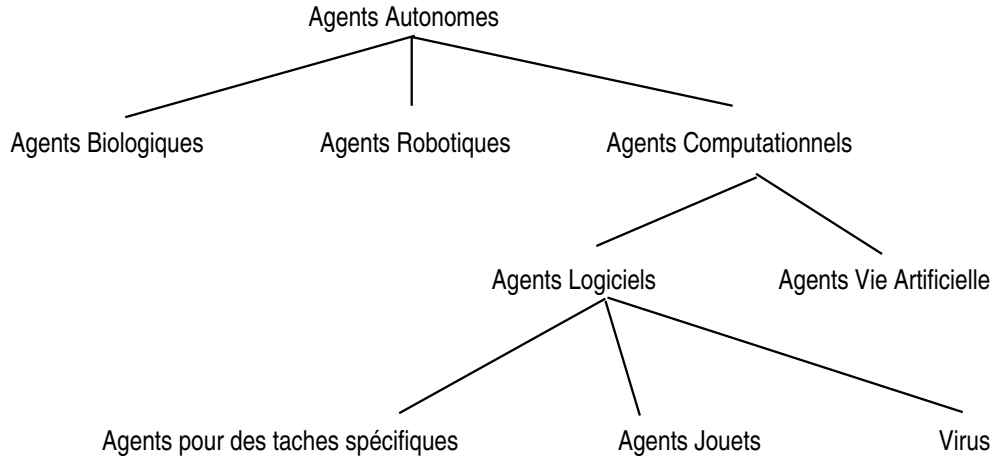


FIG. 1.1 – Taxonomie des variétés d'agents existants.

A en croire le succès de l'introduction de Russel et Norvig dans leur ouvrage "*Artificial Intelligence : A modern approach*" [Russel et Norvig, 1995], l'agent est le concept majeur qui permet d'expérimenter et de développer les théories et les techniques de l'Intelligence Artificielle. Ce mot "agent" devient, pour une partie de la communauté IA, un concept fédérateur.

Dans cette section, nous présentons tout d'abord les définitions d'agent telles qu'elles sont exprimées en IA de manière générale. Puis, nous différencierons les définitions que le mot agent inspire dans le domaine des systèmes multi-agents. Nous ne manquerons pas d'analyser, tant que faire se peut, la diversité et l'homogénéïté que cache ce petit mot de cinq lettres[2].

[2]Le lecteur attentif aura compté sur ses doigts les cinq lettres que compte le mot agent.

1.1.1 Qu'est-ce qu'un agent ?

Russel et Norvig définissent l'agent ([Russel et Norvig, 1995] page 33) :

> **Définition 1 (Agent) :**
> "An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors." □

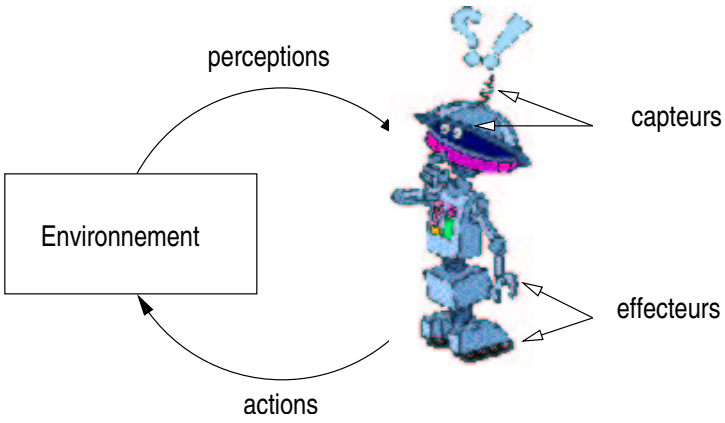


FIG. 1.2 – Exemple de fonctionnement d'un agent : la boucle sensori-motrice

Selon cette définition, un agent est un système qui décide par lui-même ce qu'il doit faire pour atteindre ses objectifs (déterminés à sa conception). De manière générale, quelle que soit la définition, un agent suit le comportement décrit par la figure 1.2 :

1. étant dans un état ou une configuration interne,
2. l'agent perçoit l'environnement par l'intermédiaire de ses capteurs, et
3. il analyse ses perceptions afin d'en élaborer un comportement, lequel provoque :
4. soit un changement de son état interne, soit une influence sur l'environnement grâce à l'action de ses effecteurs.

Ce processus se répète indéfiniment, on parle de boucle "sensori-motrice". Nous venons de décrire le fonctionnement général d'un agent, étudions à présent les propriétés que l'on peut lui attribuer.

Autonomie : définition de Russel et Norvig

Si un agent doit être robuste (résistant aux impondérables) dans un contexte en évolution, on parlera d'agent autonome. L'autonomie est la capacité d'un individu à déterminer son propre comportement. Un agent ne nécessite donc pas le concours d'un humain ou d'un autre agent [Russel et Norvig, 1995].

Rationalité : définition de Russel et Norvig

Russel et Norvig introduisent le concept d'agent rationnel [Russel et Norvig, 1995] : un agent qui agit avec raison, qui effectue les *bonnes* actions au bon moment. Autrement dit, l'agent rationnel est "intelligent". On ne peut s'empêcher une digression concernant le principe de rendre

un agent "intelligent" à l'image de l'être humain[3]. Chacun d'entre nous sait pourtant qu'il n'est pas rationnel en toute situation[4].

Les auteurs proposent de quantifier la qualité d'une action ou d'un comportement d'un agent. Pour cela, Russel et Norvig invoquent l'implication d'un observateur extérieur (par exemple le concepteur). Ce dernier doit être capable de définir le modèle d'évaluation des actions d'un agent selon les objectifs pour lesquels l'agent a été conçu.

Il est aussi important de se poser la question du "quand" évaluer. Faut-il évaluer une action ou le comportement de l'agent ? En effet, si l'évaluation se restreint à récompenser les actions immédiates de l'agent, une action qui dans un premier temps peut paraître inutile pourra se révéler indispensable par la suite. L'agent doit donc être évalué sur le principe du long terme, et non sur celui de l'évaluation immédiate.

En allant plus loin dans la notion d'agent rationnel, les auteurs définissent l'agent rationnel idéal.

> **Définition 2 (Agent rationnel idéal) :**
> Pour chaque séquence perceptive possible, l'agent rationnel idéal devrait effectuer l'action capable de maximiser sa mesure de performance, sur la base des preuves fournies par la séquence perceptive et compte tenu de la connaissance dont l'agent peut disposer. [Russel et Norvig, 1995] □

Ainsi, effectuer des actions afin d'obtenir des informations intéressantes pour prendre une décision constitue une part importante du comportement rationnel idéal.

Si nous analysons ce principe de rationalité, il ne fait aucun doute qu'il est dédié à qualifier le comportement d'une seule entité "intelligente", autrement dit d'un seul agent. Les agents des systèmes multi-agents apportent une nouvelle dimension, autrefois ignorée par les chercheurs et mise en valeur par Brooks dans [Brooks, 1986]. Ses propos révèlent pour la première fois l'importance de l'environnement dans la démarche de conception d'un agent "intelligent". Un agent n'est pas isolé de son environnement, il interagit. Ainsi, les capacités d'adaptation d'un agent, qui font partie du comportement rationnel, ne sont pas codées dans l'agent ni représentées explicitement. Elles apparaissent par l'échange, par les interactions avec un environnement ou, dans le cas des systèmes multi-agents, avec d'autres agents dans cet environnement.

Point de vue multi-agents : définition de Ferber

Jacques Ferber consacre un premier ouvrage aux systèmes multi-agents en 1995. On y découvre une définition du terme "agent" détaillée, qui allie le fonctionnement de l'agent et ses propriétés.

> **Définition 3 (Agent) :**
> On appelle "agent" une entité physique ou virtuelle :
>
> 1. qui est capable d'agir dans un environnement,
> 2. qui peut communiquer avec les autres agents de manière directe ou indirecte,

[3] Russel et Norvig distinguent d'ailleurs les approches imitant l'intelligence humaine aux approches rationnelles [Russel et Norvig, 1995]

[4] Et cela nous rend bien plus humains !

3. 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),
4. qui possède des ressources (temps CPU, mémoire...) propres,
5. qui est capable de percevoir (mais de manière limitée) son environnement,
6. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
7. qui possède des compétences et offre des services,
8. qui peut éventuellement se reproduire, et
9. 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. [Ferber, 1995] □

Tandis que les items 1 et 5 reprennent les principes d'action et de perception comme nous l'avons vu dans le paragraphe (1.1.1) concernant la définition générale d'un agent, Ferber intègre d'autres propriétés dans un agent de type SMA. Il distingue principalement la capacité de communiquer avec les autres agents (2), la prise de décision en accord avec ses objectifs (3) et en tenant compte de ses ressources propres (4). Enfin la propriété de localité de l'agent (6) influe sur ses connaissances de l'environnement.

Remarquons que Ferber définit avec plus de concision les qualités d'autonomie d'un agent. En effet, les décisions de l'agent sont dirigées par ses tendances, en accord avec les ressources disponibles et ses connaissances. L'agent est de nouveau autonome par rapport à une intervention humaine, mais son évolution est liée à celle de l'environnement ou des autres agents compte-tenu du contexte SMA.

Enfin, nous ne pouvons nous empêcher d'apprécier le vocabulaire utilisé par l'auteur qui, comme il le fait lui-même remarquer dans son ouvrage, tend à personnifier l'agent : lui donner une vie[5].

Point de vue multi-agents : définition de Wooldridge et Jennings

Une autre école conçoit les systèmes multi-agents sur des bases cognitivistes. Wooldridge et Jennings proposent une définition plus générale de l'agent. On y retrouve les propriétés de situation et d'autonomie.

> **Définition 4 (Agent) :**
> "An agent is a computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives."[6] [Jennings *et al.*, 1998]

Les auteurs précisent les trois mots clés de leur définition :

1. Situé : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement.
2. Autonomie : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne .

[5] Les lecteurs curieux se poseront la question de comment lui donner une mort ? ... Tandis que certains répondront "kill -9 agentPID", d'autres chuchoteront "Ctrl+Alt+Suppr".

[6] "Un agent est un système informatique, situé dans un environnement, et qui agit de façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu."

3. Flexible :
   - L'agent répond à temps. L'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requis.
   - L'agent doit exhiber un comportement pro-actif et opportuniste, tout en étant capable de prendre l'initiative au bon moment.
   - L'agent est social. L'agent doit être capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin de faire appel à des compétences complémentaires, ou au contraire mettre ses propres compétences à la disposition des autres agents.

Le terme "flexible" différencie la définition de Wooldridge et Jennings des précédentes. Il s'agit tout comme pour Ferber d'agents sociaux, les interactions sont le moyen de parvenir à cette flexibilité. Notons qu'il n'y a pas de restrictions sur la représentation des connaissances et en particulier de l'environnement. Les contraintes de ressources ne sont pas explicitement décrites, mais on peut les placer dans les propriétés d'autonomie. Les auteurs ne cherchent pas ici à incarner leur agent, il s'agit bien d'une entité logicielle élaborée possédant des capacités de raisonnement selon la tâche qu'elle doit accomplir.

1.1.2 Conclusion

Nous venons de faire un tour d'horizon des définitions possibles d'agent. Nous avons choisi de présenter une définition très générale de l'agent par Russel et Norvig. Ferber a précisé et cerné tout ce que l'agent était et ce que devait être son comportement à partir de capacités restreintes. Wooldridge et Jennings utilisent le terme flexible pour décrire les capacités de l'agent à résoudre la tâche qu'on lui a confiée.

Hormis les principes de fonctionnement fondamentaux (perceptions, actions ...), nous retiendrons de ces définitions, exprimées d'un point de vue SMA, les dénominateurs communs qui suscitent notre intérêt comme les *capacités d'interaction* (social, communication directe ou indirecte par le biais de l'environnement), les *compétences* et l'*autonomie*.

1.2 Systèmes multi-agents

Les systèmes multi-agents sont constitués de deux dimensions qui peuvent être vues de manière distincte : le collectif ("système multi-") et l'individu ("agents"). Chacune de ces dimensions est caractérisée par des propriétés spécifiques : la capacité, l'intention, l'autonomie au niveau individuel et le rôle, le groupe, la mission, les règles d'interaction au niveau collectif. Nous allons voir que, dans un système, la nature de chaque composant importe moins que les interactions ou relations qu'il entretient avec les autres.

En introduction de ce chapitre, nous avons parlé de la diversité des modèles de systèmes multi-agents exposés dans la littérature. Dans cette section, nous présentons trois définitions de systèmes multi-agents qui introduisent les notions d'interaction, d'organisation et d'environnement. Les deux premières ne laissent pas apparaître de formalisme et restent générales, tandis que la dernière est la seule à notre connaissance à énoncer le paradigme multi-agents sous la forme d'un formalisme "d'influence et de réaction".

1.2.1 Qu'est-ce qu'un système multi-agents ?

Pour Ferber, un système multi-agents est un système composé des éléments suivants [Ferber, 1995] :

1. Un environnement E, c'est-à-dire un espace disposant généralement d'une métrique.
2. Un ensemble d'objets O. Ces objets sont situés, on peut leur 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.
3. Un ensemble A d'agents, qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système.
4. Un ensemble de relations R qui unissent les objets (et donc des agents) entre eux.
5. Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O.
6. 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.

A partir de cette définition, l'auteur définit les systèmes à agents purement communicants. Dans ce cas, E est l'ensemble vide, A correspond à O, et R représente un réseau. L'auteur définit également les systèmes à agents purement situés qui ne possèdent aucune capacité de représentation et qui communiquent indirectement à travers l'environnement (agents réactifs).

Dans [Jennings *et al.*, 1998], Jennings, Sycara et Wooldridge décrivent les caractéristiques des systèmes multi-agents. Un système multi-agents est un ensemble d'agents en interaction qui cherchent à accomplir un ou plusieurs buts sous les conditions suivantes :

- Chaque agent a des informations ou des capacités de résolution de problèmes incomplètes. Ainsi chaque agent a un point de vue limité.
- Il n'existe pas de contrôle global du système.
- Les données sont décentralisées.
- Les calculs sont asynchrones.

En accord avec les applications pour lesquelles les systèmes multi-agents seraient idéalement adaptés, les auteurs exposent la notion d'interaction entre les agents (détaillée dans la section suivante). Tout comme pour Ferber, les informations perçues ne sont pas complètes. En résumé, les définitions précédentes mettent l'accent sur l'importance de la localité, de l'indépendance et des interactions, mais elles ne donnent pas de formalisme à l'univers multi-agents et restent très générales. Les systèmes multi-agents nous apparaissent comme une combinaison de deux facteurs déterminant le niveau de localité des connaissances des agents et la nature de leurs interactions.

1.2.2 Système multi-agents : le modèle influence-réaction

Dans [Ferber et Müller, 1996], Ferber et Müller tentent de pallier ce manque de théorie en modélisant les actions dans les systèmes multi-agents et les paramètres qui interviennent dans leurs conceptions. Cette théorie repose sur la distinction entre les influences du comportement des agents et la réaction de l'environnement.

Systèmes multi-agents

Dans cette approche, un système multi-agents est défini comme la composition de deux sous-systèmes dynamiques imbriqués que constituent l'environnement et le ou les agents :

1. L'environnement possède sa propre dynamique et évolue en fonction de ses lois propres et des actions du ou des agents qu'il héberge.
2. L'agent est un système dynamique dont l'état interne évolue en fonction de ses perceptions propres. Cet état interne influence directement la décision ou l'action que va entreprendre l'agent.

Environnement

Plus formellement, l'environnement est défini par le système $\langle E, \Gamma, \Sigma, R, RL \rangle$ dans lequel :
- E est l'environnement dans lequel évoluent les agents.
- Γ est l'ensemble des influences ou actions que peuvent entreprendre les agents dans cet environnement.
- Σ est l'ensemble des états possibles de l'environnement, et on note $\sigma(t)$ l'état de E à l'instant t.
- R définit les lois d'évolution de l'environnement :

$$\sigma(t+1) = R(\sigma(t), \Pi_{i \in Agents} Infl_i(s_i(t)))$$

et on note Π la résultante des actions effectuées par les agents.
- RL est la fonction de renforcement qui, à un état $\sigma(t)$ dans lequel se trouve le système, associe une valeur qui indique l'adéquation de se trouver dans cet état vis-à-vis des objectifs globaux du système.

Agent

Un agent est défini par le système $Ag =< P_a, Percept_a, F_a, Infl_a, S_a >$ dans lequel :
- $Percept_a$ est l'ensemble des perceptions que peut percevoir un agent.
- P_a est la fonction de perception de l'agent.
- S_a est l'espace des états internes de l'agent a, et on note $s_a(t)$ l'état interne de l'agent a à l'instant t.
- F_a définit la dynamique interne d'un agent en fonction de son état précédent et des perceptions qu'il a du monde :

$$s_a(t+1) = F_a(s_a(t), P_a(\sigma(t))) \text{ avec } \sigma(t) \in \Sigma$$

- $Infl_a$ est la fonction de décision d'un agent qui, à un état interne $s_a(t)$, associe une action de Γ.

Un système à base d'agents se décrit donc comme suit :

$$s_i(t+1) = F_i(s_i(t), P_i(\sigma(t))) \quad \forall i \in Agents \tag{1.1}$$

$$\sigma(t+1) = R(\sigma(t), \Pi_{i \in Ag} Infl_i(s_i(t))) \tag{1.2}$$

1.2.3 Problématiques de conception

En s'appuyant sur cette modélisation, le point de vue défendu par mon équipe de recherche, l'équipe MAIA, concevoir un système à agents consiste à résoudre un certain nombre de problèmes :

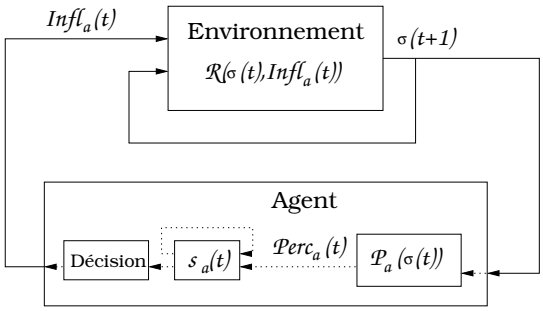


FIG. 1.3 – Description du modèle influence-réaction

1. Quelle architecture donner à un agent : quelles sont les capacités et les ressources dont doit disposer un agent pour résoudre un problème ?
2. Combien d'agents faut-il pour résoudre collectivement un problème ?
3. Comment définir l'espace des états internes S_a d'un agent a, sa dynamique F_a et sa fonction de décision $Infl_a$ de sorte qu'il maximise ses performances vis-à-vis des objectifs qui lui sont alloués ?
4. Comment représenter l'environnement E et sa dynamique R ?
5. Quel signal de renforcement RL, local ou global, interne ou externe faut-il susciter pour créer un système capable d'évoluer vers la structure spacio-temporelle recherchée ?

Nous apporterons quelques éléments de réponse à ces questions au cours de notre développement de thèse. Notre démarche prend d'ores et déjà le parti de respecter les contraintes et avantages qu'apportent la localité aux niveaux des perceptions, des prises de décision et des actions des agents.

1.2.4 Conclusion

En conclusion sur les définitions des systèmes multi-agents, on fera référence au travail de Demazeau [Demazeau, 1995], qui décrit les différentes dimensions des systèmes multi-agents autour des voyelles "AEIO" : A pour agents, E pour environnement, I pour interaction, et O pour organisation. Ainsi, on peut distinguer les systèmes où :

- les agents sont autonomes : ils agissent (A) et perçoivent l'environnement (E) ;
- les agents interagissent : ils sont autonomes (A+E) et interagissent (I) avec les autres agents du système ;
- les agents sont sociaux : agents interagissant (A+E+I) capables de gérer et d'entretenir des relations (O) avec d'autres agents.

De ces définitions, deux courants antagonistes sur l'approche décisionnelle ont émergé, conduisant à distinguer agents réactifs et agents cognitifs. Dans la suite de ce chapitre, nous allons étudier la distinction usuelle entre les systèmes multi-agents cognitifs et réactifs, puis nous présenterons les concepts d'interaction et d'organisation.

1.3 Systèmes à agents : réactifs *vs* cognitifs

Dans la littérature, deux visions s'opposent, d'un coté les systèmes multi-agents dits cognitifs, de l'autre les systèmes multi-agents réactifs. Les uns s'appuient sur l'intelligence des agents de

manière individuelle, les autres sur l'intelligence collective. Tandis que les systèmes d'agents cognitifs sont constitués d'un petit nombre d'agents capables d'effectuer un grand nombre de traitements élaborés au cours de leurs interactions, la conception de systèmes à agents réactifs s'appuie sur une conception émergente de l'intelligence : un grand nombre d'agents doués de faibles connaissances individuelles, adopte un comportement global cohérent et efficace.

1.3.1 Systèmes à agents cognitifs

Un système multi-agents cognitif est composé d'un petit nombre d'agents "intelligents". On dit aussi que les agents sont intentionnels, c'est-à-dire possèdent des buts et des plans explicites leur permettant d'accomplir leurs tâches. Les principales caractéristiques sont les suivantes :

- Les agents ont une représentation explicite de l'environnement et éventuellement une mémoire du passé.
- Les agents ont des capacités de planification et de prise d'engagement.
- Les relations entre les agents sont basées sur un mode d'organisation sociale.
- Enfin, les systèmes multi-agents cognitifs comprennent peu d'agents (entre dix et vingt au maximum).

Les premiers systèmes ont été conçus sur la base de communications de haut niveau. Les processus de négociation sont aussi utilisés pour réaliser une coopération ou une résolution de conflits. Ces dialogues sont souvent nombreux et complexes et nécessitent des capacités évoluées (langage, représentation de l'univers, mémoire [Ferber, 1995]). Les agents BDI[7] sont un exemple d'agents cognitifs [Rao et Georgeff, 1995][Wooldridge et Jennings, 1995].

Avantages et inconvénients

Cette approche a des inconvénients dûs à l'inhérente complexité de ces agents et des processus de coopération :

- complexité des protocoles de communication et durée des négociations qui croissent fortement lorsque le nombre d'agents en interaction augmente,
- temps important de réalisation des tâches (traitements sur des représentations symboliques de l'univers, planification des actions, dialogues, etc.),
- faibles performances pour des actions en temps réel (type navigation),
- agents inadaptés aux environnements dynamiques et/ou inconnus (peu adaptatifs).

En revanche, les systèmes cognitifs permettent de garder une approche descendante : connaissant les données d'un problème, il est souvent possible de concevoir le système multi-agents adéquat pour le résoudre.

1.3.2 Systèmes à agents réactifs

Les approches réactives sont une solution aux problèmes et limites rencontrés par les approches cognitives, tant au niveau de la complexité des agents cognitifs en terme de difficulté de conception qu'au niveau des performances et de l'adaptabilité des systèmes dans un environnement en évolution. Les principales caractéristiques sont les suivantes :

- Les agents n'ont pas de représentation, ni de mémoire du passé.
- La prise de décision d'un agent se fait sous la forme "stimuli-réponse".

[7] Acronyme de l'anglais : "Belief Desire Intention" dont la traduction française pourrait être "Croyance Désir Motivation"

- Le mode d'organisation des agents est souvent d'inspiration biologique (stratégie d'exploration en groupe).
- Le système compte beaucoup d'agents (>100).

On peut citer les premiers travaux en robotique mobile de l'architecture de subsomption de Brooks [Brooks, 1986] et le travail de Steels sur les robots fourrageurs [Steels, 1989] qui s'appuient sur les comportements élémentaires réactifs. Dans ces travaux, les systèmes utilisent des agents autonomes et l'environnement. Plus récemment, dans sa thèse [Simonin, 2001], Simonin propose une architecture avec des agents à la fois réactifs et intentionnellement coopératifs. La résolution d'un problème distribué est alors obtenue par un ensemble de coopérations locales (directes) et une auto-organisation au niveau global (coopération indirecte).

Avantages et inconvénients

Les avantages liés à l'utilisation d'une approche réactive sont une meilleure fiabilité du système (la perte d'un agent ne remet pas en cause le processus général), la production d'une performance collective qualitativement supérieure à celle des unités, une plus grande fléxibilité face aux situations imprévues (tolérance aux pannes).

En contre-partie, l'approche collective ou réactive pose un certain nombre de problèmes lorsqu'elle est envisagée pour des applications réelles [Simonin, 2001] :

- difficultés pour anticiper la résolution d'un problème par une intelligence "émergente" (problème de formulation, de compréhension et de preuve de la résolution),
- nécessite un grand nombre d'agents, donc risques de conflits et de coût élevé,
- risques de comportements oscillatoires ou bloquants,
- pas de comportements volontairement coopératifs.

Le paradigme collectif connaît certaines limites lorsqu'il est question de réaliser des tâches complexes. Le grand nombre d'agents utilisés introduit des problèmes de chutes de performances à partir d'un seuil de surpopulation et d'un coût financier lorsqu'il s'agit d'applications réelles [Drogoul et Ferber, 1993].

La solution de ces problèmes vient souvent de la conception de systèmes à agents hybrides capables de coopérer en introduisant une couche délibérative au-dessus d'une couche réactive. Toutefois, les problèmes de tels systèmes restent les mêmes que ceux des agents cognitifs.

1.3.3 Systèmes à agents hybrides

Les approches réactive et cognitive se complètent : chacune d'elles répond à des besoins précis, mais manque de ce que l'autre peut apporter. Pallier les défauts de chacune de ces approches, c'est l'objet des systèmes à agents hybrides, on parle également d'architectures multi-niveaux. Elles se basent sur une hiérarchie de niveaux qui interagissent entre eux [Ferguson, 1992][Müller, 1997]. Dans ces architectures, il existe au moins deux modes de contrôle des échanges d'information entre les niveaux :

- contrôle horizontal : tous les modules (un par niveau) sont directement connectés aux capteurs externes et à la sortie qui déclenche des actions ; chaque module interne à l'agent se comporte comme un agent en proposant des actions à faire (figure 1.4) ;
- contrôle vertical : seul un module gère les entrées (capteurs) et un autre les sorties (actions à faire) (figure 1.5).

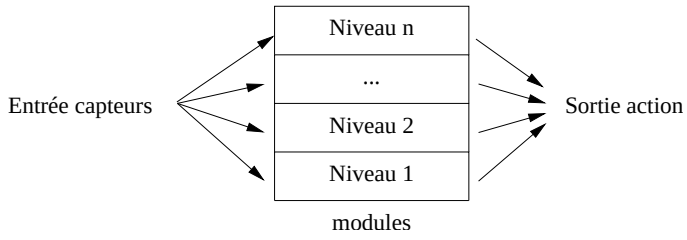


FIG. 1.4 – Agent à contrôle horizontal

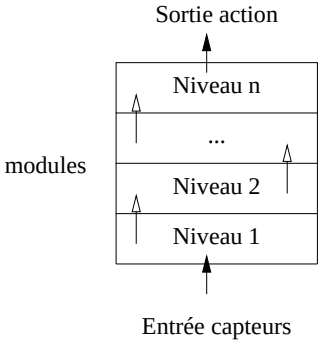


FIG. 1.5 – Agent à contrôle vertical (une passe)

Avantages et inconvénients

Chacune de ces approches a ses avantages et ses inconvénients. Le contrôle horizontal est plus facile à concevoir, mais il engendre un comportement parfois incohérent. Le contrôle vertical contrôle les informations en effectuant une ou deux passes dans les modules avant de prendre une décision. Le problème de ce type d'architecture est de devoir passer par chaque module avant de prendre une décision. De plus, en cas de panne d'un des modules, le comportement de l'agent sera perturbé.

La solution est de diminuer le nombre de modules, afin de simplifier l'architecture des agents et de garder la simplicité des agents réactifs en y intégrant le raisonnement des agents cognitifs.

Enfin, remarquons que le terme "agent hybride" ne se réduit pas à ce type d'architecture. Plus généralement, on parle d'agent hybride lorsque le fonctionnement d'un agent fait appel à certaines propriétés des agents cognitifs et réactifs.

1.4 Systèmes multi-agents et interactions

Une des caractéristiques essentielles qui différencie les systèmes multi-agents et la résolution de problèmes distribués est la notion d'interaction qui existe au sein du système. Dans [Ferber, 1995], Ferber propose une classification des types d'interaction possibles dans les systèmes multi-agents selon la nature des buts des agents (compatibles ou non), l'accès aux ressources et les compétences des agents par rapport aux tâches à résoudre. Il classe les types d'interaction de la manière suivante :

- Indifférence des agents. Dans ce cas, les agents prônent l'indépendance.
- Coopération des agents. Les interactions sont de type collaboration simple, encombrement, collaboration coordonnée.

– Antagonisme entre les agents. Les interactions prennent les formes de compétition individuelle pure, compétition collective pure, conflits individuels pour des ressources ou encore conflits collectifs pour des ressources.

Dans le contexte de notre étude, nous avons représenté sur la figure 1.6 une classification des différentes méthodes d'interactions selon le type de la situation (indépendance, compétition, coopération). Fort logiquement, nous avons choisi de détailler la branche des situations de coopération qui sont l'objet de notre étude. Cette classification n'est pas la seule possible, elle peut varier selon que l'on se place au niveau des méthodes ou au niveau des situations. Cette figure met en avant le grand nombre de méthodes de coopération et les différentes stratégies de coordination possibles. Dans cette thèse, nous nous intéressons aux situations où les agents coopèrent en utilisant comme méthode de coopération la planification d'actions.

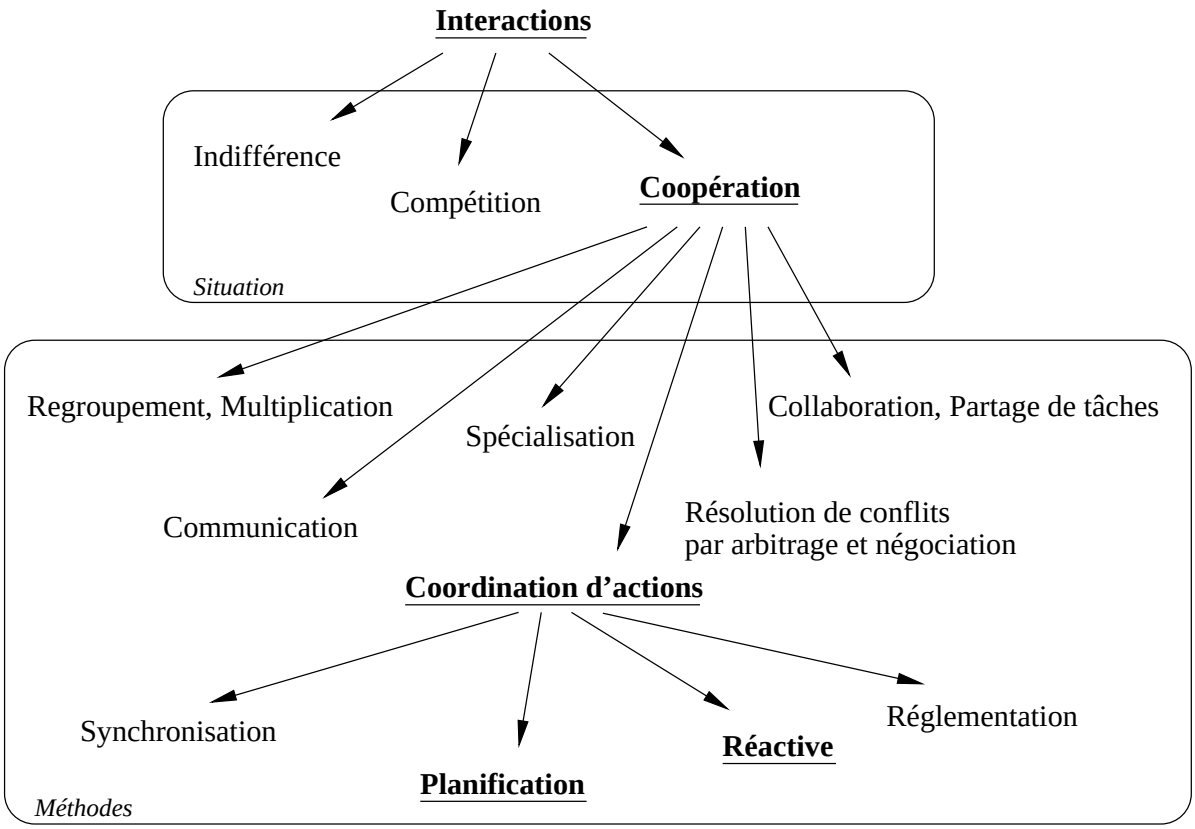


FIG. 1.6 – Positionnement de la planification : une méthode de coordination d'actions pour des agents coopérants.

1.4.1 Coopération

La coopération est une des caractéristiques les plus intéressantes des systèmes multi-agents. Ferber définit la coopération [Ferber, 1995] (page 81) :

Définition 5 (Coopération) :
"On dira que plusieurs agents coopèrent, ou encore qu'ils sont dans une situation de coopération, si l'une des deux conditions est vérifiée :

1. l'ajout d'un nouvel agent permet d'accroître différentiellement les performances du groupe,
2. l'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels." □

En effet, la coopération est parfois nécessaire à l'accomplissement de certaines tâches et peut être synonyme d'accroissement des performances [Simonin, 2001] :

- réalisation d'une tâche qui requiert la combinaison quantitative des efforts de plusieurs agents,
- réalisation d'une tâche qui requiert la combinaison de compétences diverses réparties chez divers agents spécialisés,
- entre-aide, répartition des tâches et partage des ressources pour l'amélioration des performances individuelles ou globales.

Parmi les méthodes utilisées pour mettre en œuvre cette coopération, Ferber distingue en plus de la coordination d'actions (détaillée dans le paragraphe suivant) et de la communication (directe ou indirecte grâce à l'environnement), quatres autres méthodes [Ferber, 1995] :

- Regroupement, Multiplication : la simple agrégation des agents est une forme de coopération. Le regroupement simplifie la navigation de nombreux agents. La multiplication des agents assure une grande fiabilité (ou robustesse) au groupe.
- Spécialisation : c'est un processus qui conduit un agent à progressivement se spécialiser dans certaines de ses tâches.
- Répartition des tâches, des informations et des ressources : il s'agit d'un processus collaboratif permettant aux agents de se répartir les tâches, les informations et les ressources dans le but de réaliser un objectif commun. Cette répartition peut se faire au sein de systèmes délibératifs par des mécanismes d'offre et de demande. Dans des systèmes réactifs, cette répartition se fait par le biais de l'environnement, et conduit à la spécialisation des agents et à leur répartition géographique.
- Arbitrage et Négociation : sont deux moyens de gérer les conflits entre agents. L'arbitrage établit des règles sur le comportement des agents qui ont pour conséquence au niveau global de limiter les conflits. La négociation intervient lorsque les agents interagissent pour prendre des décisions communes, alors qu'ils poursuivent des buts différents. L'objectif de la négociation est de résoudre des conflits qui pourraient mettre en péril des comportements coopératifs. La négociation nécessite un système de communication de haut niveau afin de parvenir à une solution ou à un compromis.

1.4.2 Coordination d'actions

Dans un environnement dynamique et du fait de leur nombre, les situations où les agents doivent résoudre les conflits sont coûteuses et impliquent une complexité de protocoles tant au niveau de la négociation que dans la conception des agents. La coordination permet aux agents de s'adapter à la dynamicité de l'environnement tout en évitant la complexité de cette phase de conflits.

De manière générale, Malone décrit la coordination comme :

Définition 6 (Coordination) :
"L'ensemble des activités supplémentaires qu'il est nécessaire d'accomplir dans un environnement multi-agents et qu'un seul agent poursuivant les mêmes buts n'accomplirait

pas."[Malone, 1987] □

Nous retiendrons la définition de coordination d'actions dans un SMA coopérant que propose Ferber :

Définition 7 (Coordination d'actions) :
"La coordination des actions, dans le cadre de la coopération, peut donc être définie comme l'articulation des actions individuelles accomplies par chacun des agents de manière à ce que l'ensemble aboutisse à un tout cohérent et performant."[Ferber, 1995] □

Comme nous l'avons vu précédemment, les situations de coopération n'entraînent pas obligatoirement l'utilisation de méthodes de coordination. Lorsqu'on choisit de la mettre en œuvre, la coordination peut se dérouler avec ou sans communication. On distingue quatre formes de coordination :

- Coordination par synchronisation. Il s'agit de définir la façon dont s'enchaînent les actions, tant au niveau des mouvements qu'en terme d'accès à une ressource (systèmes automatiques industriels, systèmes d'exploitation répartis).
- Coordination par réglementation. Il s'agit de suivre des règles de comportement afin d'éviter les conflits.
- Coordination par planification. C'est le type de coordination qui nous intéresse dans cette thèse. On parle de planification centralisée pour agents multiples, de coordination centralisée pour plans partiels, ou encore de planification distribuée.
- Coordination réactive. Il n'est question d'aucune planification, les agents réactifs s'auto-organisent au travers de leurs interactions avec l'environnement.

La coordination par planification est généralement associée aux techniques traditionnelles de planification mono-agent. Par exemple, dans [Durfee et Lesser, 1991], la Planification Partielle Globale est une approche flexible (au sens de Wooldridge) qui permet aux divers agents d'un système de se coordonner dynamiquement. Les agents interagissent en se communiquant leurs plans et leurs buts selon un niveau d'abstraction approprié. Ces communications permettent à chacun d'anticiper les actions futures d'un ou de plusieurs autres agents, augmentant ainsi la cohérence de l'ensemble des agents. Comme les agents coopèrent, le receveur d'un message peut utiliser les informations reçues afin d'ajuster sa propre planification.

A contrario, dans cette thèse, nous cherchons à éviter les contraintes de complexité de la conception par planification traditionnelle en combinant la coordination par planification et la coordination réactive. Il s'agira donc de calculer, sous certaines conditions, des plans individuels réactifs qui rendent compte d'une forme de coordination des agents.

1.5 Systèmes multi-agents et auto-organisation

Hormis l'importance des interactions, il existe parfois une autre composante qui intervient dans la conception d'un SMA capable de résoudre une tâche collective : l'organisation. Cette notion d'organisation est alors peu formalisée lorsqu'elle est émergente (auto-organisation).

D'une manière générale, l'organisation est un modèle permettant aux agents de coordonner leurs actions au cours de la résolution d'une ou plusieurs tâches. Elle définit d'une part une structure comprenant un ensemble de rôles qui doivent être attribués aux agents et un ensemble de chemins de communication entre ces rôles. Elle définit d'autre part un régime de contrôle qui dicte le comportement social des agents. Enfin, elle définit des processus de coordination qui

déterminent la décomposition des tâches en sous-tâches, l'allocation des sous-tâches aux agents et la réalisation des tâches dépendantes de façon cohérente [Malville, 1999].

1.5.1 Exemples d'organisation

Sans rentrer dans les détails, nous présentons ici des exemples d'organisation reconnus et souvent appliqués.

Hiérarchie

Dans une hiérarchie [Fox, 1981], chaque niveau intermédiaire est constitué de décideurs qui coordonnent les efforts des agents du niveau inférieur. L'inconvénient de la hiérarchie est que chaque niveau introduit un délai supplémentaire dans le traitement des tâches. Ce délai dû à la distribution de tâches peut être compensé de manière globale en exploitant le parallélisme. Cependant, dans le cas de tâches simples, les délais de distribution deviennent plus importants que les temps de traitement [So et Durfee, 1993].

Réseau contractuel

Le réseau contractuel (*"contract net protocol"*)[Smith, 1980] est historiquement le premier protocole utilisant un processus de négociation. Dans ce type d'approche, les actions sont initiées après le succès d'une négociation d'un contrat. Ainsi, lorsqu'un agent a une tâche à réaliser, il diffuse un appel d'offres à tous les agents. Chaque agent concerné soumet alors une offre. Si l'appel d'offres reçoit une réponse satisfaisante, l'agent initiateur contractualise l'offre la plus acceptable. Il peut également renouveler son appel d'offres ou attendre d'autres réponses.

Notons qu'il n'y a ici aucune structure de contrôle sur les agents, ce qui fait de cette organisation un système très robuste. Elle est également capable de s'adapter aux contraintes de temps. Cette organisation est donc appréciée pour des systèmes fortement dynamiques. Cependant, il faut veiller à la surcharge du réseau dûe à la diffusion des appels d'offres envoyés à tous les agents [Ferber, 1995].

Marché centralisé

Dans un marché centralisé, un agent courtier sert d'intermédiaire entre les acheteurs et les fournisseurs. De manière générale, le courtage (trading) consiste à centraliser les offres et les demandes. Le courtier recherche parmi les offres qu'il a reçues la ou les plus appropriées aux besoins des clients [Wolisz et Tschammer, 1993] [Malville, 1999]. Le courtier permet de réduire les inconvénients de surcharge du réseau contractuel. Les appels d'offres sont uniquement envoyés aux fournisseurs compétents. Les inconvénients de cette approche sont les mêmes que ceux de la centralisation : la vulnérabilité du système augmente. Il est possible d'augmenter le nombre de courtiers, ce qui entraîne une nouvelle gestion du processus entre les courtiers, les fournisseurs et les acheteurs.

1.5.2 Auto-organisation et émergence

Par opposition à l'organisation statique que nous avons présentée précédemment, l'auto-organisation permet à un système multi-agents de s'adapter de manière autonome à la dynamique des situations imprévues. Un système auto-organisateur est un système qui change sa structure de base en fonction de son expérience et de son environnement [Ünsal, 1993]. D'un point de vue

comportemental, le groupe MARCIA a défini l'auto-organisation comme le fait de passer d'un état stable du système à un autre de façon autonome [MARCIA, 1996].

Quant à l'évaluation de l'auto-organisation, le problème a été soulevé et reste un sujet ouvert. Puisque la réorganisation consiste à s'adapter face à des situations imprévisibles, comment peut-on évaluer le comportement d'un système auto-organisateur [MARCIA, 1996][Malville, 1999] ?

Mécanismes d'auto-organisation et émergence

Il est possible de classer les mécanismes d'auto-organisation de différentes manières. Youngpa So et Durfee [So et Durfee, 1993] distinguent les approches globales descendantes : un agent impose la restructuration de l'organisation ; et les approches locales ascendantes : chaque agent participe à la réorganisation sans avoir une vision globale du système. Chaque agent décide d'appliquer un changement local, qui peut pousser les autres agents à réaliser des changements locaux, et donc engendrer des modifications en cascade, ainsi la réorganisation globale émerge des comportements individuels.

Dans [Malville, 1999][Ishida *et al.*, 1992], les auteurs font un inventaire des mécanismes d'auto-organisation selon leur nature.

- Modification de l'attribution des rôles. Le réseau contractuel de Smith [Smith, 1980] est un mécanisme d'auto-organisation dans la mesure où il permet aux agents de s'organiser eux-mêmes pour résoudre des problèmes particuliers (un agent peut jouer un rôle différent selon qu'il propose un contrat, ou une offre).
- Instanciation de structures organisationnelles. Ces travaux portent sur l'idée de laisser à disposition des agents des structures organisationnelles prédéfinies dans une bibliothèque. Le groupe d'agents choisit parmi les modèles de structures, celui qui convient le mieux à la situation présente (recherche à travers l'espace des organisations possibles) [So et Durfee, 1993] [So et Durfee, 1996] [Grislin-Le Strugeon *et al.*, 1993].
- Modification de la topologie (modification des liens entre agents). Grâce aux interactions, les agents peuvent acquérir de nouvelles connaissances et découvrir l'existence d'autres agents (acte d'apprentissage collectif) [Camps et Gleizes, 1995] [Camps et Gleizes, 1996].
- Modification des liens préférentiels [Foisel, 1998].
- Réorganisation physique des agents. Bourdon propose un modèle descriptif des interactions observables entre les composants du système, et de leurs évolutions dans le temps [Bourdon, 1997]. Ünsal étudie l'auto-organisation dans le domaine de la robotique mobile et propose un mécanisme d'attraction/répulsion basé sur la valeur du champ gravitationnel d'une charge que les robots doivent encercler [Ünsal et Bay, 1994].
- Répartition de la connaissance (évaluation de la situation par chaque agent, choix d'une primitive de réorganisation appropriée selon des règles fixées [Ishida *et al.*, 1992]).
- Auto-organisation mixte (physique et logique) [Guichard, 1996].

Ce paragraphe nous a permis de comprendre le fonctionnement de certains SMA qui s'articulent autour de l'efficacité de leur mode d'organisation. En utilisant des plans réactifs, une forme d'auto-organisation peut apparaître. Il nous est toutefois difficile d'évaluer comment elle se produit. Dans la suite de ce manuscrit, nous ne ferons plus référence à l'organisation.

1.6 Systèmes multi-agents et intelligence collective

Dans cette thèse, nous cherchons à formaliser la conception d'un système multi-agents, ayant à notre disposition les caractéristiques du problème à résoudre : il s'agit d'une approche descen-

dante. Les agents de notre système coopèrent et suivent un comportement réactif afin de réaliser la tâche qu'on leur a confié dans un environnement incertain. A titre de comparaison sur une conception ascendante de systèmes multi-agents, nous présentons dans cette section le principe des systèmes multi-agents réactifs d'inspiration biologique.

1.6.1 Introduction

Les travaux des éthologues sur l'étude des comportements collectifs chez les insectes sociaux ont permis la mise au point d'algorithmes pour l'optimisation et le contrôle. Les algorithmes de fourmis ont été présentés pour la première fois comme une approche multi-agents pour résoudre des problèmes d'optimisation par Dorigo et ses collègues [Dorigo *et al.*, 1991]. Ces algorithmes de fourmis s'inspirent de l'observation de l'organisation des colonies réelles. A ce jour, les algorithmes d'intelligence collective s'utilisent aussi dans des applications réelles, les algorithmes ACO (Ant Colony Optimization) et ACR (Ant Colony-Routing) connaissent un succès grandissant [Bonabeau *et al.*, 1999].

Dans les paragraphes suivants, nous présentons les principes de base des algorithmes inspirés par le comportement des fourmis : ces agents minimalistes qui permettent de résoudre des problèmes de grande complexité. Nous verrons que la conception de tels systèmes multi-agents reste une affaire d'expérimentation empirique.

1.6.2 Recherche de plus court chemin

Les fourmis sont des insectes sociaux, elles vivent en colonies et leur comportement est dirigé par l'instinct de survie de la colonie dans son ensemble et non par une seule composante individuelle. Un des comportements les plus intéressants des fourmis est celui de l'activité de recherche de nourriture, et en particulier comment les fourmis trouvent le plus court chemin entre des sources de nourriture et leur nid.

Lorsque les fourmis marchent des sources de nourritures au nid et vice-versa, elles déposent sur le sol une substance chimique appelée phéromone, et forment ainsi une piste. Les fourmis, capables de sentir la phéromone, choisissent le chemin marqué par les plus fortes concentrations de phéromone. Cette marque leur permet de trouver le chemin pour, de la source de nourriture, revenir à leur nid. Elle permet également aux autres fourmis de trouver la source de nourriture en suivant la piste chimique indiquée par la concentration de phéromone.

Il a été montré expérimentalement que ce comportement lié à la piste de phéromone peut donner lieu à l'émergence des plus courts chemins. Aussi, lorsque plusieurs chemins sont possibles du nid à une source de nourriture, une colonie de fourmis doit être capable d'exploiter les pistes de phéromone laissées par les individus pour découvrir le plus court chemin du nid à la source de nourriture et vice versa.

Pour étudier le comportement des fourmis lors de la recherche de nourriture, Deneubourg et al. [Deneubourg *et al.*, 1990][Dorigo et Di Caro, 1999] ont mis au point une expérience. Il s'agit de séparer la fourmilière et la source de nourriture par un double pont où chaque branche a la même longueur (figure 1.7). Laissées à leur propre destin, elles se déplacent librement entre la source de nourriture et le nid. Le pourcentage de fourmis qui choisit l'une ou l'autre des branches est observé au court du temps . Le résultat est qu'après une phase initiale transitoire d'oscillations, les fourmis convergent vers le même chemin.

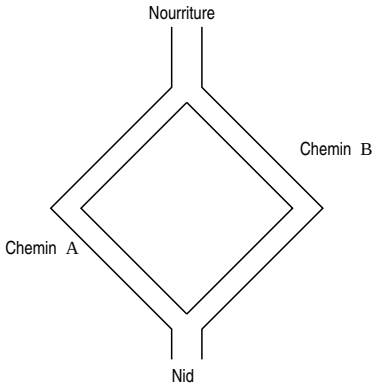


FIG. 1.7 – L'expérience de Deneubourg.

Le modèle probabiliste décrivant le phénomène observé est le suivant [Goss *et al.*, 1989][Dorigo et Di Caro, 1999]. Soit U_m et L_m les nombres de fourmis qui ont emprunté respectivement la branche gauche ou la branche droite après que m fourmis aient traversé le pont. On a $U_m + L_m = m$. La probabilité $P_U(m)$ avec laquelle la $(m+1)^{\text{ème}}$ fourmi choisit la branche gauche est :

$$P_U(m) = \frac{(U_m + k)^h}{(U_m + k)^h + (L_m + k)^h} \tag{1.3}$$

tandis que la probabilité $P_L(m)$ qu'elle choisisse la branche la droite est :

$$P_L(m) = 1 - P_U(m)$$

Les paramètres h et k permettent d'adapter le modèle théorique aux données expérimentales.

La dynamique de choix de la fourmi suit l'équation :

$$si\ \psi \leq P_U \quad U_{m+1} = U_m + 1$$

$$et \quad U_{m+1} = Um \quad sinon$$

où ψ est une variable aléatoire d'une distribution de probabilité uniforme sur l'intervalle $[0, 1]$.

Lorsque les branches sont de différentes longueurs, il est facile d'étendre le modèle de l'équation (1.3) de manière à ce qu'il décrive une nouvelle situation [Goss *et al.*, 1989][Dorigo et Di Caro, 1999]. Dans ce cas, la branche la plus courte est la plus souvent empruntée : les premières fourmis qui arrivent à la source de nourriture sont celles qui prennent le chemin le plus court, ainsi, lorsque ces fourmis retournent au nid, il y a plus de phéromone sur le chemin qu'elles viennent d'emprunter, ce qui les incite à emprunter le plus court chemin.

Le processus décrit est bien celui d'un mécanisme d'optimisation distribué dans lequel chaque agent fourmi donne une toute petite contribution. Ainsi, bien qu'une fourmi esseulée est capable de trouver une solution, c'est bien l'ensemble des fourmis qui trouve la solution optimale : le plus court chemin. On parle de propriété émergente du comportement de la colonie de fourmis. Cette forme de communication indirecte s'appelle la "stigmergie" [Grassé, 1959]. Elle correspond à un mode de coordination par le biais de l'environnement (coordination réactive). Son effet principal est de modifier la façon dont l'environnement (mémoire des actions passées de la collectivité) est localement perçu par les fourmis.

1.6.3 Algorithmes ACO

Les algorithmes ACO reposent sur la métaheuristique ACO (Ant Colony Optimization) dont le principe est d'utiliser la coopération des fourmis afin de résoudre des problèmes d'optimisation discrets à travers la stigmergie. Les bonnes solutions obtenues sont une propriété émergente des interactions coopératives des fourmis. Ces fourmis artificielles ont en commun un certain nombre de similarités avec leurs homologues réelles :

- Colonies d'individus coopérants. Les colonies artificielles sont composées d'une population d'entités globalement coopérantes. Les fourmis coopèrent à travers l'information qu'elles écrivent ou lisent de façon concurrente dans chaque état visité du problème considéré.
- Piste de phéromones et stigmergie. Tout comme les fourmis réelles, les entités artificielles modifient certaines caractéristiques de leur environnement : cela se traduit par le changement d'une valeur numérique d'un état visité. Par analogie, ce phénomène est appelé piste de phéromone artificielle. Cette forme de communication stigmergique joue un rôle majeur dans l'utilisation de la connaissance collective. L'évaporation de la phéromone au cours du temps est également retranscrite dans les algorithmes ACO. Elle permet à la colonie de fourmis d'oublier lentement ses décisions passées et permet ainsi l'exploration de nouveaux états dans de nouvelles directions.
- Recherche de chemin le plus court et déplacements locaux. Les fourmis artificielles et réelles partagent la même tâche : trouver le plus court chemin (optimisation de coût) entre un état de départ (nid) et des états d'arrivées (nourritures). Les déplacements se font entre états adjacents du problème.
- Politique de prise de décision stochastique et locale. La prise de décision des fourmis dans un état donné suit une politique décisionnelle probabiliste. Elles se déplacent ainsi vers des états adjacents. Cette politique considère uniquement de l'information locale dont les fourmis disposent, sans prendre en compte de quelconques projections futures. De ce fait, la prise de décision de chaque fourmi est complètement locale dans l'espace [8] et le temps.

Les fourmis artificielles diffèrent de leurs homologues réelles à différents niveaux qui rendent exploitables les algorithmes ACO :

- Les fourmis artificielles vivent dans un monde discret, leurs déplacements consistent en une transition d'un ensemble d'états discrets dans un ensemble d'états discrets.
- La fourmi artificielle a un état interne, qui contient la mémoire de ses actions passées.
- Elle dépose une quantité de phéromone qui est une fonction de la qualité de la solution trouvée.
- L'instant où le dépôt de phéromone se produit dépend de l'application. Par exemple, une fourmi pourra attendre de trouver une solution avant de mettre à jour la qualité de la piste chimique.
- Enfin, pour améliorer l'efficacité des algorithmes fourmis, il est parfois intéressant de les enrichir avec de nouvelles capacités comme par exemple l'optimisation locale ou le backtracking.

Ainsi, des fourmis artificielles peuvent être utilisées dans un algorithme pour résoudre des problèmes d'optimisation discrète (ACO) [Dorigo et Di Caro, 1999]. Soulignons la nécessité de déterminer les paramètres qui permettent le déplacement de la fourmi. C'est-à-dire son état interne et sa mémoire pour les informations individuelles ; la quantité et le moment de dépôt en ce qui concerne la phéromone. Ces paramètres dépendent de l'application.

[8] intersidéral

1.6.4 Applications

Voyageur de commerce, ordonnancement de tâches, routage dans les réseaux de communication, coloration de graphe, autant d'applications de grandes complexités auxquelles les algorithmes basés sur les fourmis ont permis de trouver des solutions de bonne qualité. En revanche, les algorithmes ACO ne seront pas adaptés aux problèmes dans lesquels les états possèdent un trop grand nombre de voisins accessibles. En effet, une fourmi qui se trouve dans un état et qui se déplace sur un état voisin aura le choix entre un grand nombre de déplacements possibles. Ainsi, la probabilité qu'un nombre de fourmis suffisamment important visitent le même état sera très faible. Par conséquent, il y aura peu de différence entre l'utilisation ou non de pistes de phéromone. Le lecteur intéressé pourra se reporter à [Dorigo et Di Caro, 1999].

1.6.5 Conclusion

Bien que les algorithmes d'intelligence collective donnent des résultats de qualité surprenante pour certains problèmes, il faut noter les contraintes de cette approche de conception ascendante. En effet, il est à ce jour impossible de savoir si les fourmis seront adaptées à un problème avant de l'avoir expérimenté et adapté au fur et à mesure en réglant un certain nombre de paramètres outils. On ne peut prouver la convergence vers une bonne solution. Les études sur le sujet se multiplient, et tentent de comparer et de comprendre l'approche intelligence collective avec des approches plus classiques [Bonabeau *et al.*, 1999]. Nous retiendrons toutefois les avantages de travailler avec des agents réactifs simples, comme la robustesse à l'évolution de l'environnement (adaptation aux pistes tronquées, défaillance de certains agents) qui n'entrave en rien le travail du système.

1.7 Systèmes multi-agents et incertitude

Les situations auxquelles sont confrontés les systèmes multi-agents sont de plus en plus souvent réelles. De ce fait, l'environnement dans lequel évoluent les agents retranscrit lui aussi la complexité de la réalité. C'est dans ce contexte de recherche que nous faisons évoluer nos agents. Ainsi, dans ce paragraphe, nous nous intéressons aux effets de l'incertitude dans une application multi-agents : la poursuite de proie [Benda *et al.*, 1986].

En 1992, Korf propose une solution très simple pour résoudre un problème académique "La poursuite de proie" [Korf, 1992]. Cette application met en jeu des prédateurs au nombre de quatre se déplaçant sur un échiquier et qui doivent encercler une proie afin de la capturer. Le travail de Korf mêle fonction d'utilité d'un agent et coopération implicite. Il reste à ce jour une référence dans le domaine. Nous avons étudié les effets de l'incertitude dans le déplacement des prédateurs afin de les rendre plus proches de la réalité des situations étudiées en robotique mobile [Laroche, 2000][Dutech, 1999].

1.7.1 Poursuite de proie : Algorithme glouton amélioré de Korf

Dans cette application, Korf remet en question la nécessité ou l'utilité d'une coopération explicite entre agents. Pour cela, il propose un algorithme glouton amélioré (the greedy algorithm) basé sur la maximisation d'une fonction d'évaluation qui dépend uniquement de la mesure d'utilité locale de chaque agent. Cette utilité est l'expression :

- d'une mesure de distance (selon une norme) entre un agent prédateur et la proie, et entre chaque position voisine libre : l'attraction vers la proie, et

– d'une force de répulsion entre les prédateurs à laquelle on attribue plus ou moins de poids : le facteur de répulsion k.

Si la position courante de l'agent est la plus proche possible, l'agent ne bouge pas. Si ce n'est pas le cas, l'agent se déplace sur la position qui maximise cette fonction d'utilité locale.

1.7.2 Etude des performances de Korf sans incertitude

Le tableau 1.1 montre l'influence du facteur de répulsion (k) dans l'application du modèle de Korf dans des conditions classiques de simulation : la proie se déplace aléatoirement sur un échiquier de taille 21×21 ; avec une vitesse inférieure à celle des prédateurs d'un coefficient $0,9$, c'est-à-dire que la proie se déplace aléatoirement sur une case libre de l'échiquier neuf fois sur dix, la dixième fois elle n'effectue aucun déplacement ; et les perceptions des agents sont complètes.

En accord avec les résultats de Korf, pour 100 simulations, les meilleures performances sont obtenues lorsque le coefficient de répulsion est égal à $0,5$. En effet, si k est trop petit ($0,1$ ou $0,3$) les prédateurs poursuivent la proie sans parvenir à l'entourer, tandis que pour des valeurs de k trop grandes (0,7 et 0,9), les prédateurs parviennent difficilement à se rapprocher de la proie.

Facteur de répulsion	$0,1$	$0,3$	$0,5$	$0,7$	$0,9$
Moyenne	$549,3$	415	**36,6**	$212,7$	$306,2$
Écart type	$708,1$	$530,1$	**49,8**	$274,9$	$317,4$

TAB. 1.1 – Performances en nombre de déplacements simultanés avant capture sur 100 simulations - Proie Folle

Si la proie suit un déplacement aléatoire (proie folle), ce modèle est très efficace pour un facteur de répulsion égal à $0,5$, en particulier dans le cas d'un jeu diagonal (les agents peuvent se déplacer aussi bien diagonalement qu'horizontalement). Cependant lorsque les agents sont confrontés à une proie réactive, ce modèle n'est plus adapté.

Facteur de répulsion	$0,1$	$0,3$	$0,5$	$0,7$
Moyenne	$2235,7$	$1968,1$	$30,2$	$53,4$
Échecs	9	8	40	66

TAB. 1.2 – Performances en nombre de déplacements simultanés avant capture sur 100 simulations - Proie Réactive

En effet, dans les mêmes conditions que précédemment, nous nous sommes attachés à faire varier le coefficient de répulsion k de la fonction d'utilité locale de chaque agent en les confrontant à une proie réactive. Comme le montre le tableau 1.2, le modèle de Korf est en situation d'échec lorsque la proie provoque des situations d'équilibre entre les agents. De la même manière que pour une proie folle, si k est trop petit les prédateurs ont de grosses difficultés à entourer la proie, et si k est trop grand des situations d'équilibre se créent et les prédateurs ne parviennent pas à se rapprocher de la proie. Contrairement à la proie folle, la proie réactive ne bouge plus lorsque les prédateurs sont bloqués, ce qui provoque des situations d'échecs (figure 1.8 A.) de plus en plus nombreuses lorsque k augmente.

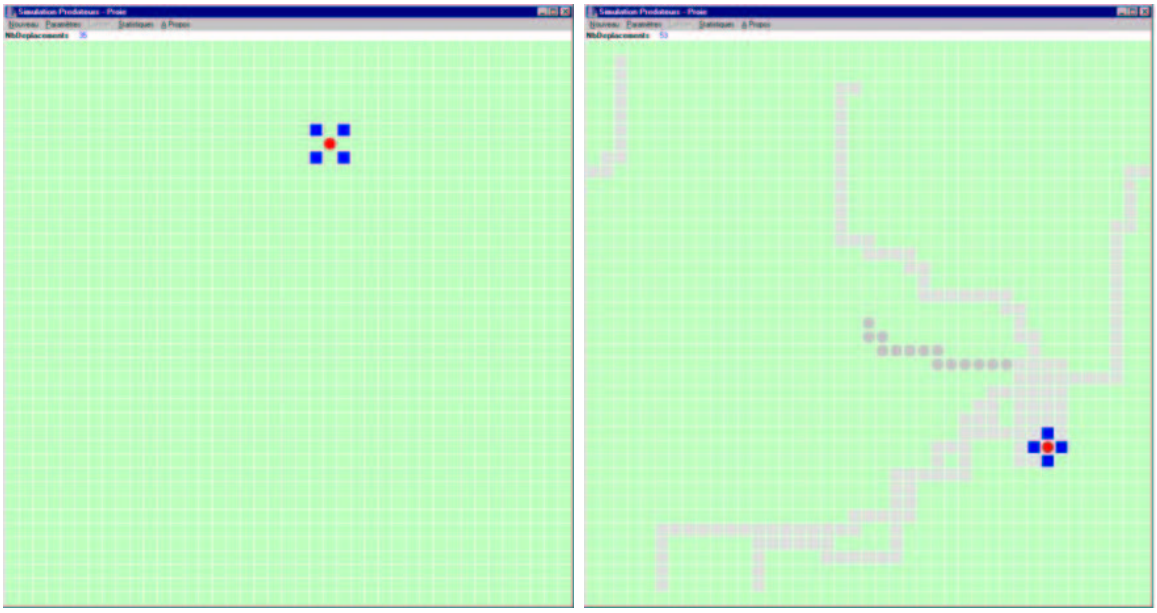


FIG. 1.8 – A - Les prédateurs sont en situation d'echec. B - Les prédateurs capturent la proie.

Le modèle de Korf est clairement dédié au problème de poursuite de proie folle. Nous venons de montrer les effets du coefficient de répulsion, intéressons-nous aux conséquences de l'introduction de l'incertitude dans le comportement des agents prédateurs. Dans la suite de nos simulations, nous choisirons la valeur du coefficient de répulsion la plus performante : $k = 0,5$.

1.7.3 Etude des effets de l'incertitude

Nous avons réalisé cinq séries de mille simulations chacune en faisant varier l'incertitude de déplacement des prédateurs de manière croissante afin d'en observer les conséquences sur les performances de l'algorithme de Korf. Nous appelons incertitude la probabilité qu'un prédateur effectue une action au hasard lors de la simulation. Ce paramètre varie de 0 (mouvement déterministe de l'agent) à $0,4$ (deux fois sur cinq l'agent suivra une action aléatoire). Le déplacement des prédateurs se fait de manière simultanée. Ils perçoivent parfaitement l'environnement. L'échiquier torique sur lequel évoluent les agents est de taille 21×21. Nous rappellons que la proie est folle et évolue à coefficient de vitesse $0,9$ par rapport aux prédateurs.

1.7.4 Résultats observés et analyse

Incertitude	0	0,1	0,2	0,3	0,4
Moyenne	37,8	30,5	38,6	54,8	92,4
Écart type	46,8	17	21,9	35,8	65

TAB. 1.3 – Performances sur 1000 simulations en nombre de déplacements simultanés - Proie Folle.

La figure 1.9 synthétise les résultats présentés tableau 1.3. *Sans incertitude*, la proie est capturée en moyenne en moins de quarante déplacements simultanés des prédateurs. L'écart type

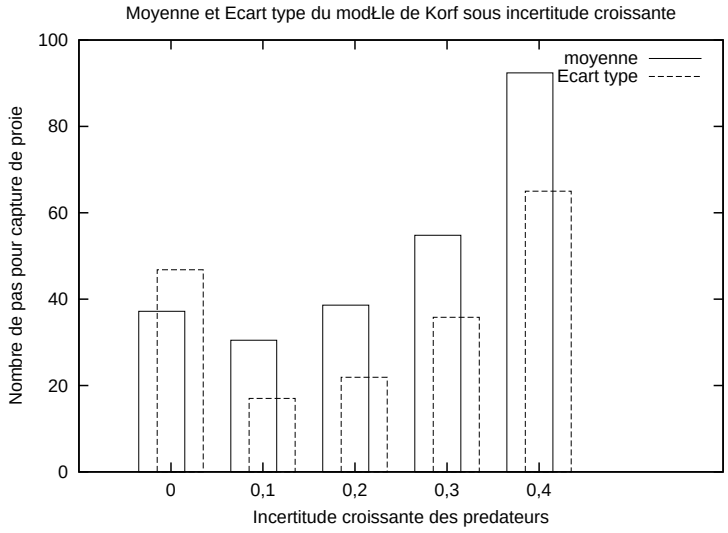


FIG. 1.9 – Effets de l'incertitude sur le modèle de Korf- Proie Folle

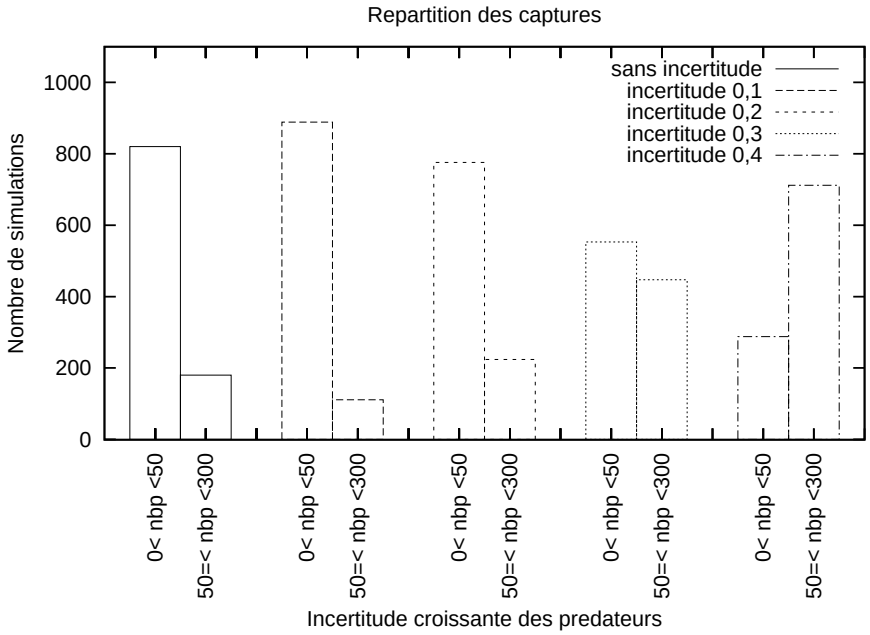


FIG. 1.10 – Histogramme de la répartition des classes des résultats obtenus

Facteur d'incertitude	0	0,1	0,2	0,3	0,4
$0 < nbp < 50$	820	889	776	553	288
$50 \leqslant nbp < 100$	105	102	204	347	357
$100 \leqslant nbp < 200$	70	9	20	100	331
$200 \leqslant nbp < 300$	5	0	0	0	24

TAB. 1.4 – Détails des classes de performances. *nbp* correspond au nombre de pas (déplacements) avant captures sur 1000 simulations.

reflète les grands écarts obtenus sur les 1000 simulations : $46,8$. La moyenne n'est alors pas un outil d'analyse appropriée. L'irrégularité des résultats apparaît plus en détails sur la table 1.4 et sur le graphique 1.10 : tandis que la capture de la proie se fait dans plus de 80% des cas en moins de 50 itérations, 75 cas sur 1000 nécessitent plus de 100 itérations. Ces chiffres s'expliquent par les situations d'équilibre provoquées par la force de répulsion entre prédateurs : les agents ont tendance à préserver une configuration de type carré, ce qui laisse apparaître lors des simulations une suite de déplacements où les agents restent autour de la proie sans la capturer. Les forces de chaque prédateur neutralisent les actions de l'ensemble du système. C'est la proie, dotée d'un comportement aléatoire, qui, en se déplaçant sur une case libre, débloque la situation et rompt l'équilibre.

Par contre, insérer de l'*incertitude* dans le comportement des prédateurs corrige le biais généré par les forces de répulsion et d'attraction. Les résultats sont meilleurs tant au niveau du nombre moyen de pas nécessaires qu'au niveau de la régularité des performances : écart type de 17 et $21,9$ pour des facteurs d'incertitude respectifs de $0,1$ et $0,2$. En rentrant dans les détails, on ne compte alors plus que 9 et 20 cas sur 1000 à plus de 100 déplacements.

Pour des facteurs d'incertitude plus importants ($0,3$ et $0,4$) les performances se dégradent : $54,8$ et $92,4$ pour la moyenne du nombre de pas nécessaires pour capturer la proie. Notons que l'écart type du facteur d'incertitude $0,3$ reste meilleur que le modèle de Korf sans incertitude.

1.7.5 Conclusion

Ces simulations ont révélé les effets de l'incertitude sur les performances d'un système multi-agents réactif qui s'appuie sur un principe simple d'interactions : l'attraction de chaque agent vers la cible et la répulsion des agents entre eux.

A faible intensité, l'incertitude améliore les performances du système en rompant les équilibres créés par les forces d'attraction et de répulsion. Ces performances sont significativement meilleures en terme de moyenne et de régularité (écart type). En revanche, dès que l'incertitude est trop importante, son influence diminue les performances du système de manière non négligeable.

A la lumière de ces simulations, nous pouvons conclure que peu d'incertitude permet de remédier aux situations d'équilibre bloquantes d'un système. Il est toutefois nécessaire d'en prévoir les conséquences si l'on désire conserver de bonnes performances.

1.8 Conclusions

La grande diversité des systèmes multi-agents élaborés depuis les années 80 en font un outil particulièrement adapté pour la résolution de problèmes distribués. Cette diversité des modèles a aussi pour conséquence la multiplication des redéfinitions de termes, de concepts majeurs et l'utilisation d'un vocabulaire parfois hermétique. Dans ce contexte très général, la modélisation formelle de systèmes multi-agents reste exceptionnelle.

Dans ce chapitre, nous avons exposé les fondements théoriques des systèmes multi-agents, tout d'abord du point de vue de l'agent, ensuite en considérant l'ensemble des systèmes.

Du point de vue de la planification d'actions dans les systèmes multi-agents cognitifs en environnement complexe, les modèles existants restent difficiles à mettre en œuvre dans le cadre

de problèmes nécessitant un grand nombre d'agents. L'adaptation au phénomène d'évolution de l'environnement passe souvent par une reconstruction partielle des agents. De plus, les formalismes théoriques proposés ne sont pas toujours adaptés à la dynamicité de l'environnement. La conception des systèmes multi-agents est maîtrisée mais elle met en jeu des mécanismes compliqués qui empêchent leur utilisation pour un grand nombre d'agents.

Les systèmes multi-agents réactifs sont, quant à eux, capables de résoudre des applications complexes en utilisant des agents étonnamment simples. Mais leur méthode de conception ascendante rend leur étude et leur utilisation particulièrement difficile. Il est à ce jour impossible de déterminer par avance les paramètres du système et des agents qui permettront de trouver des solutions de bonne qualité à un problème donné.

A la lumière de cet état de l'art sur les systèmes multi-agents, nous sommes maintenant en mesure de poursuivre notre écriture. Notre attention se concentrera à présent sur la recherche et l'évaluation d'outils de conception de plans réactifs pour des agents coopérants évoluant dans un environnement complexe. Nous garderons en mémoire les propriétés de perception partielle et d'interaction qui font des systèmes multi-agents une solution de choix pour la résolution de problèmes distribués, ainsi que la nécessité de prévoir l'incertitude. C'est vers les modèles décisionnels de Markov que notre attention s'est portée. De part leurs caractéristiques et leurs propriétés mathématiques, ils semblent s'accorder avec notre ambition de concevoir des plans réactifs individuels.

Chapitre 2

Modèles décisionnels de Markov

Dans cette thèse, nous nous intéressons aux problèmes qui mettent en jeu des agents autonomes situés évoluant dans un environnement complexe, pour résoudre une tâche collective selon un critère de performance précis. Ces entités doivent atteindre un but qui nécessite la coopération de tous. Cette classe de problèmes concerne la prise de décision avec incertitude mais également la recherche d'un comportement optimal : les agents perçoivent une configuration du système et décident de la modifier en effectuant une action individuelle, en échange ils reçoivent une récompense. Du point de vue de l'agent, sa satisfaction personnelle ne se limite pas aux effets de ses actions propres, elle dépend également des effets des actions de ses compères.

La théorie de la décision offre de nombreux outils mathématiques qui permettent de prendre en compte les incertitudes liées à la fois à la complexité de l'environnement, mais aussi aux propriétés des agents (perception, bruits inhérents aux actions). Les modèles décisionnels de Markov en font partie. Nous avons choisi de les utiliser pour concevoir de manière descendante et distribuée les plans de nos agents réactifs.

Dans ce chapitre, nous allons comprendre comment sont utilisés les principaux modèles décisionnels de Markov existants. Malgré une complexité souvent importante des algorithmes de résolution, les propriétés mathématiques qu'apportent les modèles décisionnels de Markov en font des outils de prise de décision de plus en plus étudiés dans la littérature, tant dans des situations d'apprentissage que dans des situations de planification.

Organisation du chapitre

En guise d'introduction, nous présentons les fondements de la théorie de la décision afin de situer les principes sur lesquels reposent les modèles décisionnels de Markov. Dans une deuxième section, nous nous attachons à définir et explorer l'univers des modèles décisionnels de Markov. Puis nous présenterons successivement les processus décisionnels de Markov (MDP) et les processus décisionnels de Markov partiellement observables (POMDP), ainsi que leurs méthodes de résolution. Nous mettrons en valeur les possibilités et les limites de ces deux principaux modèles. Enfin, nous terminerons cet état de l'art en étudiant plus précisément les modèles décisionnels de Markov qui mettent en jeu plusieurs agents. Tout au long de ce chapitre, nous tiendrons compte de la complexité des modèles étudiés, composante essentielle dans notre travail de recherche sur la conception d'un système multi-agents.

2.1 Introduction aux modèles décisionnels de Markov

2.1.1 Inévitable incertitude

Les systèmes qui raisonnent sur les problèmes réels ne peuvent représenter qu'une partie de la réalité. Il est évident qu'une quelconque représentation "informatique" est une simplification dramatique des objets et des relations qui peuvent être pertinents pour un problème de prise de décision. Cette inévitable incomplétude des représentations implique d'inévitables incertitudes sur les états du monde et sur les conséquences des actions. En pratique, l'incertitude est particulièrement importante avec des acteurs nombreux, des préférences complexes, des enjeux importants, et des conséquences à long terme. C'est précisément dans ce contexte que nous situons notre étude de conception de systèmes multi-agents.

Les robots sont l'exemple de référence. Ils se déplacent dans un monde réel qu'ils ne peuvent jamais représenter parfaitement du fait des erreurs de modélisation. Quand un robot suit une trajectoire, il est soumis à des déviations dans ses mouvements qui proviennent de ses imperfections mécaniques, de son système de contrôle, voire de l'environnement (glissements, frottements, ...) : on parle d'erreurs de commande. Les capteurs destinés à corriger ces erreurs sont eux-mêmes imparfaits : on parle d'erreurs de mesure.

Calculer le comportement d'un agent, en compensant cette incertitude, peut s'effectuer dans le cadre de la théorie de la décision. Afin de parfaire notre connaissance, nous présentons les principes sur lesquels repose la théorie de la décision.

2.1.2 Principes de la théorie de la décision

Les probabilités fournissent un langage pour rendre compte de l'incertitude. Elles permettent de rendre explicites les notions de croyance partielle et d'information incomplète. Une probabilité de 1 correspond à croire en la véracité d'une proposition. La théorie de la décision, quant à elle, étend ce langage pour nous permettre de faire état d'alternatives qui existent, et de l'évaluation de ces alternatives les unes par rapport aux autres. La théorie des probabilités et, plus encore, la théorie de la décision fournissent les principes pour l'inférence rationnelle et la prise de décision dans l'incertain [Horvitz *et al.*, 1988].

La théorie de la décision s'appuie sur les axiomes de probabilité et d'utilité. En effet, la théorie des probabilités fournit un cadre de travail pour donner une valeur à des croyances sur des informations incomplètes, tandis que la théorie de l'utilité introduit, pour sa part, un ensemble de principes pour rendre cohérentes les préférences et les décisions. Une décision est, ici, une allocation irrévocable des ressources sous contrôle de l'agent également appelé preneur de décision. Les préférences décrivent, quant à elles, les évaluations relatives de l'agent pour des états possibles du monde ou des résultats [Horvitz *et al.*, 1988].

Théorie de l'utilité

La théorie de l'utilité est fondée sur un ensemble d'axiomes simples et de règles concernant le choix dans l'incertitude. Le lecteur intéressé pourra se reporter à [Horvitz *et al.*, 1988] et [Russel et Norvig, 1995]. Retenons simplement que :

- Le premier ensemble d'axiomes évoque la préférence de résultats dans l'incertain. Ces axiomes assurent un ordre de préférence faible de tous les résultats. Ceci implique l'existence

d'une fonction de valeur $V(x)$, qui fait correspondre à tous les résultats une valeur scalaire telle qu'un agent préférera toujours le résultat à la plus grande valeur.

– Le second ensemble implique la notion de résultats multiples dans une situation incertaine.

L'acceptation de ces axiomes implique l'existence d'une fonction scalaire d'utilité $U(x,d)$ qui associe une valeur scalaire à chaque couple résultat (x) et décision (d). Cette valeur indique la relative désirabilité de ce couple. Par conséquent, lorsqu'il existe une incertitude sur le résultat x, les décisions préférées d sont celles qui maximisent l'espérance d'utilité $E[U(x,d)|\xi]$ selon la probabilité de distribution de x et étant données les informations disponibles (ξ).

Critère de cohérence

Une fois défini le principe d'utilité, il faut maintenant être capable de l'exploiter : on parle de critère de cohérence. De manière classique, le critère de cohérence peut être décrit de la façon suivante :

Soit un ensemble de préférences exprimées par une fonction d'utilité, des croyances exprimées par des distributions de probabilité, et un ensemble de décisions possibles. Un agent doit choisir la conduite qui maximise l'espérance d'utilité : c'est le *principe d'utilité maximale* [Horvitz *et al.*, 1988].

L'importance de ce principe est qu'il permet de calculer des préférences pour des combinaisons de résultats complexes et incertains (avec des caractéristiques multiples), à partir de préférences exprimées pour des composants simples. Ainsi, il peut être utilisé comme un outil pour aider à réfléchir sur des choix complexes en les décomposant en choix plus simples.

2.1.3 Rappel sur l'étude de la complexité

Un des inconvénients majeurs des modèles décisionnels de Markov est la complexité des méthodes de résolution qu'ils mettent en jeu. Naturellement, nous y ferons référence tout au long de ce chapitre. Nous prenons le temps dans ce paragraphe de faire le point sur cette notion de complexité.

La difficulté d'un problème peut être mesurée par le temps d'exécution d'un algorithme qui le résout, ou encore la quantité de mémoire requise lors de la mise en œuvre de cet algorithme. Cependant, les notions de temps de traitement sont dépendantes de la machine physique utilisée pour implémenter l'algorithme[9]. Les complexités temporelles et spatiales au pire cas sont les deux valeurs de complexité retenues pour spécifier un problème. On distingue ainsi deux grandes classes de problèmes :

– Les problèmes faciles pouvant être résolus par une machine de Turing de complexité au pire cas bornée par un polynôme en la taille des entrées. Ces problèmes seront aussi appelés *problèmes polynomiaux*, et les algorithmes permettant de les résoudre avec une complexité polynomiale seront appelés des *algorithmes efficaces*.

– Les problèmes difficiles. Ce sont les problèmes formalisables décidables qui ne sont pas faciles : la complexité dans le pire cas n'est pas bornée par un polynôme. Ces problèmes seront aussi appelés *problèmes non polynomiaux*, et les algorithmes permettant de les résoudre seront appelés des *algorithmes inefficaces*.

La distinction entre les problèmes faciles et les problèmes difficiles est d'une grande importance pratique :

[9] et de la qualité du programmeur ...

- Les problèmes polynomiaux correspondent à des problèmes que l'on a de bonnes chances de pouvoir résoudre (soit dès aujourd'hui, soit dans un proche avenir) avec une machine, pour toutes les tailles raisonnables des entrées.
- Les problèmes non polynomiaux correspondent à des problèmes que l'on a de bonnes chances de ne jamais pouvoir résoudre avec une machine, pour des tailles raisonnables des entrées.

En conséquence, si l'on sait qu'un problème est non polynomial, il ne sert à rien (dans la plupart des cas) d'essayer de le résoudre de façon exacte... les contraintes de temps ou d'espace mémoire que va imposer cet algorithme seront toujours prohibitives, et rendront la résolution irréaliste (des siècles de traitements ou des téra-octets d'espace mémoire), pour des tailles d'entrées suffisamment importantes. A ce titre, nous n'oublierons pas l'importance d'évaluer la complexité d'un problème.

Classe de complexité

Une classe de complexité est un ensemble de problèmes, où un problème est un ensemble infini d'instances de problèmes. Chacun de ces problèmes s'exprime sous la forme d'une question attendant une réponse binaire "oui" ou "non" : on les appelle les problèmes de décision (par opposition aux problèmes d'optimisation). La forme générale d'un problème de décision sera donc une description de l'instance du problème ; et l'expression d'une question oui/non portant sur cette instance.

Les modèles décisionnels de Markov que nous étudions dans cette thèse, sont des problèmes d'optimisation. Pour pouvoir discuter de leur complexité, il faut les convertir en problèmes de décision. Il s'agit souvent de fixer un seuil et de demander si la solution optimale rapporte une récompense qui n'est pas plus petite que ce seuil.

Nous avons vu qu'il existait des problèmes de complexité polynomiale : ils correspondent à la classe P ; et non-polynomiale : la classe NP. Il est clair que $P \subseteq NP$ car un algorithme déterministe n'est qu'un cas particulier d'algorithme indéterministe. Pour de nombreux théoriciens, il est raisonnable de penser que l'inclusion $P \subset NP$ est stricte, mais aucune démonstration n'a encore prouvé que $P \neq NP$ [Bernstein *et al.*, 2000]. Notons que nous savons résoudre tout problème de la classe NP par un algorithme déterministe grâce à la technique de retour-arrière pour la réalisation des choix. Mais cela conduit à des temps d'exécution exponentiels pour le nombre de choix considéré durant l'execution : n choix avec p alternatives conduisent à p^n étapes.

En haut de la hiérarchie des classes de complexité, nous avons la classe EXP (exponentielle) et NEXP (exponentielle non déterministe).

> **Définition 8 :**
> Un problème P est exponentiel si et seulement si il existe un entier $k > 0$ et une machine de Turing T résolvant P tels que la complexité au pire cas de T pour P, pour une entrée de taille n au sens d'un codage sympathique est en $O(2^{n^k})$. □

De même que pour les classes P et NP, il est connu que EXP $\subseteq$ NEXP, et il n'a pas été montré que EXP $\neq$ NEXP. Mais il a été montré que P et EXP sont deux classes distinctes (P $\neq$ EXP).

2.2 Modèles décisionnels de Markov

Les modèles décisionnels de Markov aident à résoudre les tâches de prise de décision séquentielles avec incertitude. A la différence du paradigme des systèmes multi-agents, la prise de décision d'un agent ou d'un ensemble d'agents est étudiée de manière centralisée. Cette prise de décision est en accord avec la fonction de récompense du système et elle se traduit par l'élaboration d'un plan.

Dans cette section, nous présentons la terminologie et les concepts de base nécessaires à la compréhension des modèles décisionnels de Markov.

2.2.1 Agent et état d'un système

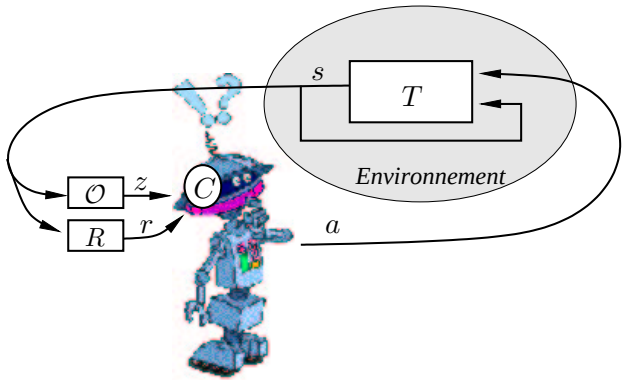


FIG. 2.1 – Comportement d'un agent.

Dans le premier chapitre, nous avons discuté du comportement d'un agent dans les systèmes multi-agents. Nous précisons maintenant le fonctionnement d'un agent dans les modèles décisionnels de Markov.

On définit l'*état d'un système* comme une description de tout ce qui peut changer d'un instant à un autre dans l'environnement. Considérons de nouveau le comportement d'un agent, mais cette fois-ci en utilisant le formalisme de la théorie de la décision. La figure 2.1 illustre un agent situé interagissant avec son environnement. L'agent est représenté par le robot, et l'environnement par l'ellipse. L'agent doit prendre une décision, donc choisir une action a, afin d'atteindre son but. Cette décision va influencer l'état de l'environnement. La fonction de transition T contrôle cette influence des actions sur l'environnement. La décision d'un agent est fonction de sa perception de l'environnement z via la fonction de perception de l'agent $\mathcal{O}$. $\mathcal{O}$ transforme l'état de l'environnement en perception. Dans de nombreux environnements, $\mathcal{O}$ est la fonction Identité, c'est-à-dire que l'agent a directement accès à l'état du système. La fonction qui fait la correspondance entre la perception et le choix de l'action est notée C et appelée "comportement". Enfin la composante R est la fonction de récompense de l'agent, et r la récompense de l'agent.

Un *état interne d'un agent* doit bien sûr refléter les sensations immédiates (perceptions), mais il peut contenir beaucoup plus d'informations. L'état informe l'agent sur tout ce qui peut être intéressant pour prendre sa décision dans les meilleures conditions. Toutefois, l'état ne doit pas informer l'agent de tout ce qui se passe dans l'environnement, il faut respecter la réalité de

la situation ainsi que les contraintes de l'environnement qui contient des informations cachées. Idéalement, il faudrait un état qui résume de manière compacte les perceptions passées telles que toutes les informations intéressantes y apparaissent.

2.2.2 Modéliser l'environnement

Dans les modèles décisionnels de Markov, un environnement est défini selon des caractéristiques précises qui déterminent la complexité et la nature du problème à traiter.

Qu'est-ce que l'environnement ?

L'agent interagit avec l'environnement à travers ses perceptions et ses actions. Les capteurs de l'agent perçoivent l'environnement, les caractéristiques de cette perception dépendent de la nature de l'agent et de l'environnement.

Dans un système multi-agents, l'environnement est l'espace commun aux agents du système. On différencie l'environnement du système multi-agents et l'environnement d'un agent qui est l'environnement du système multi-agents dans lequel sont aussi présents les autres agents.

Les caractéristiques d'un environnement s'expriment à différents niveaux. Russel et Norvig proposent dans [Russel et Norvig, 1995] la classification suivante des différents types d'environnements :

1. Accessibilité : l'agent peut obtenir des informations complètes et à jour sur la configuration de l'environnement.
2. Déterminisme : le degré de prédictibilité du comportement du système pour des données d'entrées identiques.
3. Épisodique : un épisode correspond à la période pendant laquelle l'agent perçoit et agit.
4. Statique/dynamique : est-ce que l'évolution de l'environnement est indépendante de l'activité des agents ? L'environnement dynamique possède des processus qui agissent sur son état indépendamment des actions de l'agent.
5. Discret/Continu : un environnement discret possède un nombre fini et fixé d'actions et de perceptions.

Pour Russel et Norvig, l'environnement le plus complexe est inaccessible, non-déterministe, non-épisodique, dynamique et continu. Dans ces conditions, il devient peu réaliste de calculer le comportement d'un agent.

Plus spécifiquement, dans les modèles décisionnels de Markov, on a l'habitude de définir un environnement par rapport aux caractéristiques suivantes :

Finitude

Un environnement est dit fini si le nombre de situations différentes que l'agent peut rencontrer est fini. Bien qu'il existe de nombreux problèmes réels avec des situations indénombrables, il existe également de nombreux problèmes où les configurations sont en nombre fini.

De plus, beaucoup de tâches avec environnement infini peuvent être modélisées en exhibant un environnement fini en choisissant par exemple un niveau d'abstraction approprié. Il existe également dans la littérature des méthodes d'approximation qui permettent de généraliser le passage d'un environnement infini à fini [Singh *et al.*, 1994][Munos, 2000].

Stationnarité

Un environnement est dit stationnaire si son évolution est indépendante du temps, c'est-à-dire si le résultat de l'exécution d'une action dans une situation particulière n'est pas fonction du temps. Un niveau d'abstraction approprié permet de considérer des tâches non stationnaires comme étant stationnaires. Un environnement stationnaire est bien sûr plus simple à étudier.

Observabilité

L'environnement peut être complètement ou partiellement observable. Dans le cas où les agents ont une vision incomplète de l'environnement, on dit que l'environnement est partiellement observable : il existe une incertitude sur la situation (l'état) dans lequel le système se trouve.

Dans la suite

Dans la suite de ce manuscrit, l'environnement sera fini et stationnaire. Nous ferons varier l'observabilité au cours de notre étude.

2.2.3 Propriété de Markov

Les propriétés mathématiques des modèles décisionnels de Markov reposent sur le respect ou non de la propriété de Markov.

De manière générale, si l'on considère l'évolution de l'environnement au temps $t+1$ après avoir effectué l'action a_t, la probabilité d'être dans l'état s' avec une récompense r dépend de l'ensemble des couples (état, action) passés. Quels que soient s', r et toutes les valeurs possibles des évenements passés $s_t, a_t, r_t, s_{t-1}, a_{t-1}, ..., r_1, s_0, a_0$, l'évolution de l'environnement s'exprime de la façon suivante :

$$P(s_{t+1} = s', r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, a_{t-1}, ..., r_1, s_0, a_0) \tag{2.1}$$

En revanche, si nous sommes dans un cadre vérifiant la **propriété de Markov**, alors l'évolution de l'environnement au temps $t+1$ ne dépend que de l'état et de l'action à l'étape t. Ce qui s'exprime formellement comme suit :

$\forall\ (s', r, s_t, a_t)$:

$$P(s_{t+1} = s', r_{t+1} = r | s_t, a_t) \tag{2.2}$$

Autrement dit, les états d'un système sont dits markoviens, si et seulement si (2.1) est égale à (2.2).

L'énoncé de cette propriété suggère que seuls les systèmes évoluant dans des environnements markoviens peuvent s'exprimer de manière complète sous la forme de l'équation (2.2). Toutefois comme nous l'avons vu dans le paragraphe (2.2.1), il suffit juste de réussir à exprimer l'état du système de manière à ce qu'il vérifie l'équation (2.2).

2.3 Processus Décisionnels de Markov

Parmi l'ensemble des modèles décisionnels de Markov, l'utilisation des processus décisionnels de Markov (MDP[10]) est appropriée à la résolution d'un problème de décision séquentielle et d'op-

[10] Markov Decision Processes

timalité dans un environnement markovien complètement observable. Dans le cas d'un environnement partiellement observable, les processus de Markov partiellement observables (POMDP[11]) prévalent.

2.3.1 Définition

On appelle communément MDP les processus qui satisfont la propriété de Markov. Si les espaces d'actions et d'états sont finis, comme ce sera le cas ici, on parle de Processus Décisionnels de Markov finis.

Définition 9 :
Formellement, un MDP est défini par $\langle S,A,T,R\rangle$:

- $S = \{s\}$: un ensemble d'états fini ;
- $A = \{a\}$: un ensemble d'actions fini ;
- $T : S \times A \to \mathcal{P}(S)$: une fonction de transition d'états. $T(s,a,s') = P(s_{t+1} = s' | s_t = s, a_t = a)\ \forall\ s,a,s'$ fait correspondre à chaque s', s, a une probabilité de transition ;
- $R : S \times A \times S \to \mathcal{P}(\mathbb{R})$: une fonction de récompense qui à une transition associe une distribution de probabilité. $R(s,a,s') = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\}$ est la valeur de la récompense attendue. □

Souvent on se limitera à une fonction de récompense déterministe $R : S \times A \times S \to \mathbb{R}$. Les quantités $T(s,a,s')$ et $R(s,a,s')$ reflètent les aspects les plus importants de la dynamique d'un problème modélisé par un MDP. La fonction T retranscrit les incertitudes liées aux erreurs de commandes ou de modélisation du problème à traiter. Les motivations d'un agent sont matérialisées par la fonction de récompense R.

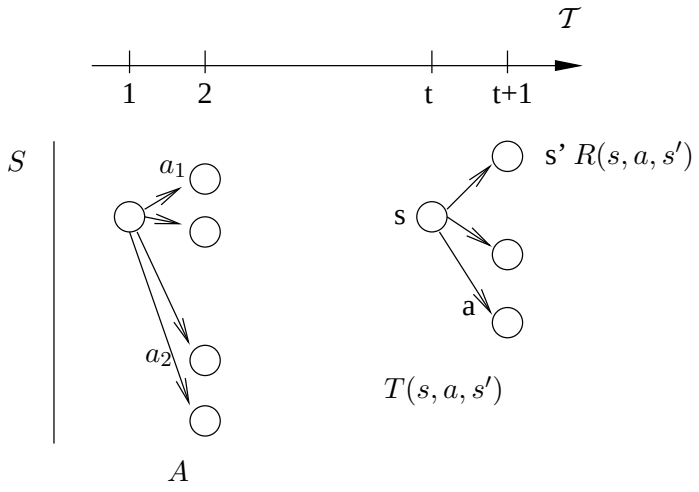


FIG. 2.2 – Vue générale d'un MDP

La figure 2.2 illustre le principe général d'un MDP. A chaque instant t de $\mathcal{T}$, l'agent perçoit l'état courant $s \in S$, applique sur le système une action $a \in A$ qui suivant T conduit le système

[11]Partially Observable Markov Decision Processes

dans un nouvel état s', et reçoit une récompense $R(s,a,s') \in \mathbb{R}$. Le domaine $\mathcal{T}$ des étapes de décision est dans le cas le plus général un sous-ensemble de la demi-droite $\mathbb{R}^+$. Pour $\mathcal{T}$ discret, l'ensemble des étapes de décision peut être fini ou infini, on parle d'horizon fini ou d'horizon infini. Les distributions T vérifient la propriété de Markov (2.2.3).

Conformément à l'hypothèse de stationnarité des systèmes que nous étudions, les processus décisionnels de Markov stationnaires sont un cas particulier de MDP. Cela se traduit par l'indépendance des probabilités de transition et de la fonction de récompense par rapport au temps :

$$\forall t \in \mathcal{T}, \quad T_t() = T(),\ R_t() = R()$$

Par la suite, nous supposerons vérifiée cette hypothèse de stationnarité.

2.3.2 Politique

L'agent doit accomplir un objectif ; ses actions doivent le mener vers son but. Une manière d'y parvenir est de faire correspondre à chacune de ses actions une récompense. L'ensemble des récompenses définit le but à atteindre, elle est immédiate. Le but de l'agent peut alors devenir : "maximiser les récompenses".

Définition

La politique détermine pour un agent la manière de se comporter dans un état donné. Elle peut être vue comme une association stimuli/réponse. Dans le cas d'un comportement déterministe, elle fait correspondre à chaque état s dans S une action a dans A :

$$\pi_t : s \in S \rightarrow \pi_t(s) \in A$$

La décision a ne dépend que de l'état actuel s, ce qui est suffisant dans un cadre markovien. Elle est dans ce cas déterministe, mais pourrait être stochastique comme nous le verrons dans l'étude des POMDPs.

Tout comme pour l'environnement dans lequel évolue un agent, notre étude discute de la recherche de politiques stationnaires :

$$\forall t \in T \ \ ,\ \pi_t(s) = \pi(s)$$

Se poser un problème décisionnel de Markov, c'est rechercher parmi une famille de politiques Π celles qui optimisent un critère de performance donné pour le processus décisionnel de Markov considéré. Au sein des MDP, ce critère a pour ambition de caractériser les politiques qui permettront de générer les séquences de récompense les plus importantes possibles.

2.3.3 Fonction de valeur ou utilité d'une politique

L'utilité d'une politique ou la valeur d'un état pour une politique donnée dépend du critère d'optimalité retenu. Afin de comparer différentes politiques selon ces critères, on définit pour chacun d'eux une fonction de valeur V, qui, pour une politique π fixée, associe à tout état initial $s \in S$ la valeur espérée du critère considéré en suivant π à partir de s :

$$\forall \pi, \quad V^\pi : S \rightarrow \mathbb{R}$$

On note $\mathcal{V}$ l'espace des fonctions de S dans $\mathbb{R}$.

Autrement dit : tandis que les récompenses déterminent la désirabilité immédiate et intrinsèque des états de l'environnement, les valeurs indiquent la désirabilité des états au long-terme après avoir pris en compte les états qui sont vraisemblablement susceptibles de suivre, ainsi que les récompenses disponibles dans ces états.

L'objectif d'un problème décisionnel de Markov est de rechercher et de caractériser, si elles existent, les politiques optimales π^* telles que :

$$\forall \pi \in \Pi,\ \forall s \in S,\quad V^{\pi}(s) \leq V^{\pi^*}(s)$$

2.3.4 Critères d'optimalité

Avant de penser aux algorithmes pour calculer des comportements optimaux, il faut déterminer quel critère d'optimalité (de performance) utiliser. Il est clair que l'agent doit prendre en compte les récompenses futures, la question est : comment ? On distingue trois modèles principaux : l'horizon fini, l'horizon infini γ-pondéré, et le critère de la récompense moyenne.

Horizon fini : Espérance de gain

A un instant donné $t = 0$, dans un état s_0, l'agent doit optimiser son espérance de gain pour les n prochaines étapes :

$$\forall s \in S,\quad V_n^{\pi}(s) = E(\sum_{t=0}^{n-1} r_t | s_0 = s) \tag{2.3}$$

On ne s'inquiète pas de savoir ce qui se passera après ces n étapes. r_t représente la valeur scalaire de la récompense au temps t dans le futur à partir de l'état s_0, son expression dépend de la fonction de récompense choisie. On peut l'écrire $R(s_t, \pi(s_t))$.

Le modèle d'optimalité à horizon fini n'est pas souvent utilisé car il nécessite une connaissance précise et au préalable du temps de vie de l'agent.

Horizon infini : Espérance de gain γ-pondéré

Le modèle à horizon infini γ-pondéré prend en compte toutes les récompenses. Néanmoins, les récompenses reçues dans le futur sont atténuées géométriquement par un facteur γ avec $0 \leq \gamma < 1$:

$$\forall s \in S,\quad V_{\gamma}^{\pi}(s) = E(\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s) \tag{2.4}$$

On peut interpréter γ de différentes façons. Il peut être vu comme un taux d'intérêt, ou encore un stratagème mathématique pour éviter la divergence de la somme infinie. Ce modèle défavorise les gains à long terme. Pour un γ proche de 0, l'agent tiendra compte principalement des récompenses immédiates. Lorsque γ tend vers 1, le comportement obtenu sera plus efficace sur une longue période.

Ce critère d'optimalité est le plus adapté et le plus utilisé dans la littérature, grâce notamment aux algorithmes qu'il permet de mettre en œuvre. C'est ce critère que nous utiliserons dans la suite de cette thèse.

Horizon infini : Espérance de gain moyen

La deuxième possibilité pour évaluer une politique sur un horizon temporel infini est de calculer la moyenne des gains espérés :

$$\forall s \in S, \quad V^{\pi}(s) = \lim_{n \to \infty} E(\frac{1}{n} \sum_{t=0}^{n-1} r_t | s_0 = s) \tag{2.5}$$

Ce critère a l'avantage de ne pas utiliser de facteur d'atténuation. Toutefois il ne permet pas de distinguer entre deux politiques laquelle obtient des gains importants dans les premiers instants. Les gains obtenus dans les phases initiales sont cachés par les récompenses moyennes au long terme. Il est possible de généraliser ce modèle afin qu'il reflète à la fois les récompenses moyennes espérées au long terme et les récompenses initiales : *bias optimal model* [Mahadevan, 1996].

Exemple : critères d'optimalité

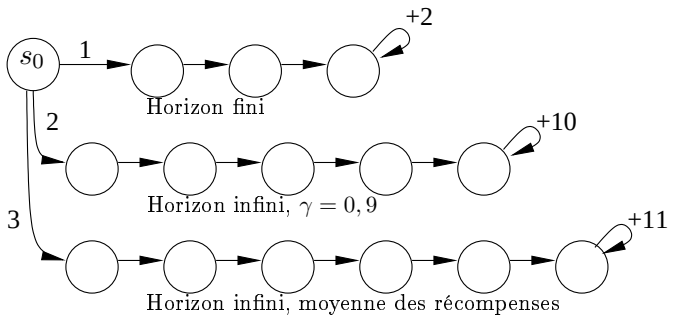


FIG. 2.3 – Comparaisons des critères d'optimalité.

La figure 2.3 met en évidence les différentes politiques optimales obtenues selon les critères d'optimalité considérés. Les états correspondent aux cercles et les actions aux flèches. En s_0, l'agent a le choix entre trois actions dont les effets sont ici déterministes. Puis, quelle que soit l'action effectuée, l'agent obtient une récompense $(+2, +10, +11)$ après un nombre variable d'étapes effectuées.

Avec un critère d'optimalité à horizon fini où $h = 5$, les trois actions entraînent des récompenses respectives de 4, 0, et 0 : la politique optimale est de choisir l'action 1. A horizon infini, et avec un critère pondéré ($\gamma = 0,9$), les trois actions entraînent des récompenses de $13,122$; $53,1441$ et $52,612659$: l'action 2 en s_0 conduit aux meilleurs gains. Dans le cas du critère d'optimalité à espérance de gains moyens c'est clairement la troisième action qui identifie la politique optimale, on obtient respectivement $2, 10$ et 11.

2.3.5 Politique optimale

Résoudre un problème décisionnel de Markov revient à calculer la politique optimale d'un agent selon un critère d'optimalité. Nous rappelons que nous avons choisi de considérer le critère de performance γ-pondéré dans la suite de cette thèse.

Définition π^*

Pour un MDP fini, une politique π est meilleure ou équivalente à une politique π' si sa récompense espérée est plus grande ou égale à celle de π'. Plus formellement $\pi \geq \pi'$ si et seulement si $V^{\pi}(s) \geq V^{\pi'}(s)$ pour tout $s \in S$. Il existe toujours au moins une politique meilleure ou égale à toutes les autres politiques : c'est une politique optimale, on la note π^*. De plus, pour un MDP, l'une d'elles est déterministe.

Fonction de valeurs optimale V^*

Bien qu'il puisse y en avoir plus d'une, nous notons toutes les politiques optimales par π^*. Elles partagent la même fonction de valeur appelée fonction de valeur optimale et notée V^* définie par :

$$V^*(s) = \max_{\pi \in \Pi} V^{\pi}(s) \ , \ \forall \ s \in S \tag{2.6}$$

Fonction d'action-valeur optimale Q^*

Une autre fonction utile est la fonction Q^{π} d'action-valeur qui mesure l'espérance des gains escomptés à l'état s en effectuant l'action a, puis en suivant la politique π.

Les politiques optimales partagent également la même fonction optimale d'action-valeur, Q^*, définie par :

$$Q^*(s,a) = \max_{\pi \in \Pi} Q^{\pi}(s,a) \ , \ \forall s \in S \ et \ a \in A(s) \tag{2.7}$$

Pour les couples état-action, cette fonction donne la récompense espérée après avoir choisi l'action a dans l'état s et en suivant de plus une politique optimale. Ainsi, Q^* peut s'écrire en fonction de V^* :

$$Q^*(s,a) = E\{r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a)\} \tag{2.8}$$

Comment calculer π^*

On distingue 2 catégories de problèmes : celle où le modèle du monde dans lequel évolue l'agent est connu dans ce cas les problèmes à résoudre sont du type planification ; et celle où la connaissance du monde est insuffisante, l'agent doit alors faire de l'apprentissage (par renforcement).

Pour calculer la politique optimale, nous faisons le lien entre π, V et Q :

$$\forall s, \ \pi^*(s) = \arg\max_a \{Q^*(s,a)\}$$

2.4 Algorithmes de résolution d'un MDP et complexité

Il existe plusieurs algorithmes de planification qui permettent de résoudre un MDP. La programmation dynamique est une méthode de résolution pour les problèmes qui satisfont au principe d'optimalité de Bellman [Bellman, 1957] :

> "Une sous-trajectoire d'une trajectoire optimale est elle-même optimale pour la fonction d'objectif restreinte aux trajectoires ayant pour origine celle de cette sous-trajectoire".

Ce principe permet une méthode de résolution ascendante, qui détermine une solution optimale d'un problème à partir des solutions de tous les sous-problèmes.

Nous présentons dans cette section, les algorithmes de résolution d'un MDP les plus utilisés dans la littérature : *Value Iteration* et *Policy Iteration*, lorsque le modèle est connu, et le *Q-learning* pour un environnement inconnu. Dans un premier temps, nous montrons comment évaluer une politique.

2.4.1 Évaluer π, évaluer les états

Les algorithmes pour résoudre les MDPs attribuent des valeurs aux états et manipulent ces valeurs de façon à trouver la politique optimale.

Equation de Bellman

A partir de l'expression de la fonction de valeur V d'une politique π :

$$V^{\pi}(s) = E_{\pi}(\sum_{t=0}^{\infty} \gamma^t r_{t+1}|s) \tag{2.9}$$

ou encore :

$$V^{\pi}(s) = R(s, \pi(s)) + \sum_{t=1}^{\infty} \gamma^t R(s_t, \pi(s_t)) \tag{2.10}$$

avec s_t une variable aléatoire qui représente un état futur, on peut montrer que V s'exprime sous la forme d'une équation de Bellman [Bellman, 1957] :

$$V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V^{\pi}(s') \tag{2.11}$$

Cette équation établit une relation entre la valeur $V^{\pi}(s)$ et $V^{\pi}(s')$ de tous les états s' du modèle. La valeur d'un état s sous la politique π est égale à la récompense immédiate reçue dans l'état s plus la valeur des gains espérés des états suivants.

Equation d'optimalité de Bellman

L'expression de la fonction de valeur optimale V^* devient alors :

$$V^*(s) = R(s, \pi^*(s)) + \gamma \sum_{s'} T(s, \pi^*(s), s') V^*(s') \tag{2.12}$$

soit :

$$V^*(s) = \max_{a \in A(s)} [R(s, a) + \gamma \sum_{s'} T(s, a, s') V^*(s')] \tag{2.13}$$

Tandis que la fonction de valeur état-action correspondante devient :

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a' \in A(s)} Q^*(s', a') \tag{2.14}$$

Algorithmes d'évaluation

On peut se servir de la formule de Bellman (2.11) pour évaluer une politique. Comme le fait l'algorithme (2.1), il suffit de tirer profit de la relation de récurrence mise en évidence :

$$V_{t+1}^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, a, s') V_t^{\pi}(s') \text{ et } a \in A(s) \tag{2.15}$$

Algorithme 2.1 Évaluer une politique par récurrence

Entrée: π, ϵ

1: **Pour tout** $s \in \mathcal{S}$ **Faire**
2: Initialiser arbitrairement $V_0(s) \leftarrow$ une valeur dans R
3: **Fin Pour**
4: **Répéter**
5: $t \in t + 1$
6: **Pour tout** $s \in \mathcal{S}$ **Faire**
7: $V_t(s) \leftarrow \sum_{s' \in S}[R(s, \pi(s), s') + \gamma V_{t-1}(s')]T(s, \pi(s), s')$
8: **Fin Pour**
9: **Jusqu'à** $\max_s |V_t(s) - V_{t-1}(s)| < \epsilon$

Sortie: V_π une évaluation de π

Il est également possible, pour une politique donnée, de résoudre un système linéaire d'équations et de déterminer sa fonction de valeur associée (algorithme (2.2)).

Algorithme 2.2 Évaluer une politique par la résolution d'un système linéaire

Entrée: π

1: Trouver les $V(s)$
2: Sachant que :

$$V(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V(s')$$

Sortie: V_π une évaluation de π

2.4.2 *Value Iteration*

Nous avons soulevé la question d'attribuer une valeur aux couples état, action dans le paragraphe précédent. Nous allons maintenant présenter les algorithmes qui utilisent cette mesure afin d'en déterminer une politique optimale dans le cas de la résolution d'un MDP.

Le *Value Iteration* est à la fois le plus simple à mettre en œuvre et aussi le plus utilisé. Il s'agit de calculer la valeur de chaque état jusqu'à une certaine précision grâce à l'utilisation de l'équation de Bellman (2.11).

L'algorithme calcule les valeurs successives de V_t en utilisant la fonction annexe : $Q_t(s, a)$. Il se termine lorsque la différence maximale entre deux fonctions de valeurs successives est inférieure

Algorithme 2.3 *Value Iteration*

Entrée: $\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, \epsilon$

$t \leftarrow 0$
Pour tout $s \in \mathcal{S}$ **Faire**
  $V_0(s) \leftarrow 0$
Fin Pour
Répéter
  $t \leftarrow t + 1$
  Pour tout $s \in \mathcal{S}$ **Faire**
    Pour tout $a \in \mathcal{A}$ **Faire**
      $Q_t(s,a) \leftarrow R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V_{t-1}(s')$
    Fin Pour
    $\pi_t(s) \leftarrow \arg\max_a Q_t(s,a)$
    $V_t(s) \leftarrow Q_t(s, \pi_t(s))$
  Fin Pour
Jusqu'à $\max_s |V_t(s) - V_{t-1}(s)| < \epsilon$

Sortie: π_n une politique

à une valeur prédéterminée ϵ. La valeur V_n de la politique obtenue ne diffère que très peu de V^*. Cette différence est inférieure à $2\epsilon\gamma/(1-\gamma)$ quel que soit l'état [Bellman, 1957][Lovejoy, 1991][Littman, 1996] :

$$\max_{s \in S} |V_n(s) - V^*(s)| < 2\epsilon \frac{\gamma}{1-\gamma} \tag{2.16}$$

Complexité

L'efficacité de l'algorithme dépend de deux facteurs : la complexité d'une itération, et le nombre d'itérations nécessaire pour converger. Dans le cas général, chaque itération prend $|A||S|^2$ pas. Les études de complexité portent sur le nombre d'itérations nécessaires pour atteindre la politique optimale.

Dans [Littman, 1996], Littman montre que déterminer la politique optimale dans le cas d'un horizon infini par l'algorithme du *Value Iteration* prend un nombre d'itérations proportionnel à $1/(1-\gamma)\log(1/(1-\gamma))$ dans le pire cas.

Dans le cas général, [Littman *et al.*, 1995] montrent que l'algorithme est polynomial selon $|S|, |A|, \gamma$ et B, où B est le nombre de bits nécessaires à la représentation des données du problème (principalement R et T).

2.4.3 *Policy Iteration*

Tandis que le *Value Iteration* calcule à chaque itération une valeur approchée de la fonction de valeur de la politique optimale, le *Policy Iteration*, lui, améliore itérativement une politique jusqu'à ce que l'algorithme converge vers une politique optimale à partir de la résolution d'un

système d'équations [Howard, 1960]. La première phase de l'algorithme (2.4) consiste à évaluer la valeur de la politique courante π en utilisant de nouveau la formule de Bellman (2.11).

Algorithme 2.4 *Policy Iteration*

Entrée: $\mathcal{M} = \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma$

// initialisation avec une politique quelconque π_i

$\pi \leftarrow \pi_i$

// calcul itératif jusqu'à obtenir 2 politiques identiques

Répéter

  $\pi \leftarrow \pi'$

  // phase d'évaluation de la politique courante

  Pour tout $s \in \mathcal{S}$ **Faire**

    Calculer $V_\pi(s)$ en résolvant les $|S|$ équations à $|S|$ inconnues (algorithmes (2.1) ou (2.2))

  Fin Pour

  // phase d'amélioration

  Pour tout $s \in \mathcal{S}$ **Faire**

    Si il existe une action $a \in A$ telle que : $R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V_\pi(s') > V_\pi(s)$ **Alors**

      $\pi'(s) \leftarrow a$

    Sinon

      $\pi'(s) \leftarrow \pi(s)$

    Fin Si

  Fin Pour

Jusqu'à $\pi = \pi'$

Sortie: π

Complexité

Chaque itération de l'algorithme consiste en deux opérations : la résolution du système d'équations, qui nécessite un peu moins de $|S|^3$ opérations, et la phase d'amélioration, qui est effectuée en $|A||S|^2$ opérations. Le nombre d'itérations est difficile à déterminer, [Littman *et al.*, 1995] donne le même résultat de complexité que pour *Value Iteration*.

2.4.4 Comparaisons

[Littman, 1996] donne différentes références, chacune affirmant la supériorité d'un algorithme par rapport à un autre [Laroche, 2000]. Puterman [Puterman, 1994] a montré que la suite de fonctions de valeur calculées par le *Policy Iteration* ne converge pas plus lentement vers V^* que la fonction de valeur calculée par le *Value Iteration*. Le *Policy Iteration* peut converger en un plus petit nombre d'itérations que le *Value Iteration*, toutefois sa vitesse de convergence peut être ralentie par l'importance des calculs à chaque itération.

Le critère d'arrêt est une autre différence entre ces deux algorithmes. Tandis que le *Value Iteration* peut converger vers la politique optimale de manière régulière, le *Policy Iteration* procède par sauts [Puterman, 1994][Littman, 1996].

2.4.5 Apprentissage par renforcement dans les MDPs

Dans les paragraphes précédents, nous avons vu comment trouver une politique optimale pour un MDP connaissant sa description complète : les états, les actions, les récompenses, les transitions. Cette situation idéale n'est pas toujours présente dans les problèmes que nous sommes amenés à étudier. Dans ces conditions, il est alors nécessaire de recourir à des techniques d'apprentissage afin de doter un agent d'une politique optimale. Nous présentons un algorithme d'apprentissage par renforcement : le *Q-learning*.

L'apprentissage par renforcement est une technique qui a pour objectif d'apprendre à partir d'expériences quoi faire en chaque situation, de manière à maximiser une récompense numérique au cours du temps. Il est important de faire le point sur les conditions d'application de l'apprentissage par renforcement :

- l'agent ne sait pas quelles actions faire ou prendre, mais il sait dans quel état il se trouve,
- il s'agit donc de procéder par essais/erreurs,
- la récompense n'est pas forcément immédiate dans le sens où des gains futurs peuvent passer par une absence de gains immédiats,
- enfin, il faut trouver un équilibre entre le besoin d'explorer l'environnement et son exploitation.

Algorithme du *Q-learning*

Introduit par [Watkins, 1989], le *Q-learning* est une méthode pour estimer la fonction optimale état-action Q^* pour un MDP inconnu. L'agent utilise l'expérience qu'il reçoit pour améliorer son estimation. Il mélange la nouvelle information perçue à sa précédente expérience en utilisant un coefficient d'apprentissage α, avec $0 < \alpha < 1$.

Nous avons vu dans les paragraphes précédents que lorsque le modèle était connu (T et R), il suffit d'obtenir une des fonctions Q, V, ou π pour calculer toutes les autres. Si T et R ne sont pas accessibles, nous pouvons utiliser la fonction Q pour reconstruire les deux autres : $V(s) = \max_a Q(s,a)$ et $\pi(s) = \arg\max_a Q(s,a)$. De plus Q n'est pas difficile à estimer à partir d'expériences réalisées dans le monde réel.

Ces expériences permettent de réaliser des séquences $\langle s, a, r, s' \rangle$ à partir desquelles il est possible d'évaluer $Q(s,a)$: l'agent est dans l'état s, il effectue l'action a, reçoit une récompense r et arrive dans l'état s'. Étant donnée une expérience $\langle s, a, r, s' \rangle$, la règle d'apprentissage de l'agent est :

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')] \tag{2.17}$$

Cette règle crée une nouvelle estimation de $Q^*(s,a)$ en ajoutant la récompense immédiate à l'estimation actuelle pondérée de $V(s')$. Ainsi, en choisissant une manière d'explorer les r et s', la valeur moyenne de cette estimation est exactement :

$$V(s) = R(s,a) + \gamma \sum_{s'} T(s,a,s')V(s')$$

Algorithme 2.5 Q-learning

1: // Initialisation
2: $Q(s,a) \in \mathbb{R}$ arbitrairement initialisé pour tout $(s,a) \in S \times A$
3: **Pour tout** épisode **Faire**
4: Initialiser s
5: **Pour tout** pas de l'épisode jusqu'à s terminal **Faire**
6: Choisir a pour s en utilisant une politique dérivée de Q
7: Effectuer l'action a ; observer r, s'
8: $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'}[Q(s',a')] - Q(s,a)]$
9: $s \leftarrow s'$
10: **Fin Pour**
11: **Fin Pour**

L'algorithme (2.5) procède par itération. Un épisode est une phase d'expérimentation qui se termine par l'accession à un état terminal. Un état terminal est défini comme un état de fin de processus. Notons qu'il n'existe pas toujours d'états terminaux dans un MDP. Il est immédiat d'observer que l'algorithme *Q-learning* est une formulation stochastique de l'algorithme (2.3) *Value Iteration*.

Bien qu'il a été montré que l'algorithme du *Q-learning* (2.5) convergeait vers Q^* en théorie, il souffre rapidement d'une explosion combinatoire quand l'espace d'états augmente. Une solution possible est parfois de ne plus travailler sur les états du système, mais sur les observations de ces dits états, dans ce cas on parlera d'apprentissage dans les POMDP [Dutech, 1999][Buffet *et al.*, 2002].

Dans la suite de ce manuscrit, nous nous intéresserons essentiellement au processus de planification qui est une forme d'apprentissage avec un modèle connu. Pour plus de détails, l'ouvrage très complet de Sutton et Barto [Sutton et Barto, 1998] développe les techniques classiques liées à l'apprentissage dans les modèles décisionnels de Markov.

2.5 Processus Décisionnels de Markov Partiellement Observés

L'observabilité de l'environnement par l'agent n'est pas toujours totale. Les POMDPs tiennent compte de cette propriété [Monahan, 1982][Lovejoy, 1991]. C'est également le cas dans les problèmes étudiés avec les systèmes multi-agents, en effet la propriété de localité des agents entraîne immanquablement une incomplétude dans les perceptions du système.

Dans cette section, nous allons nous intéresser au problème de choisir des actions optimales dans un environnement partiellement observable et stochastique. L'objectif de ces paragraphes est de comprendre l'intérêt des POMDP et d'en cerner leur possible application dans notre étude.

2.5.1 Définition

Définition 10 :

Un POMDP $M =< S, A, T, R, Z, \mathcal{O} >$ est défini en partie par un modèle de MDP :

- S l'ensemble fini des états de l'environnement ;
- A l'ensemble fini des actions de l'agent ;
- T une fonction de transition $T : S \times A \to \mathcal{P}(S)$,
- et R une fonction de récompense $R : S \times A \times S \to \mathcal{P}(\mathbb{R})$.

De plus, il inclut :

- un ensemble fini d'observations $Z = \{z\}$,
- et une fonction d'observation $\mathcal{O} : S \times A \to \mathcal{P}(Z)$. $\mathcal{O}(s, a, z)$ correspond à la probabilité d'observer $z \in Z$ dans l'état s après avoir effectué a. □

Un POMDP est donc un MDP dans lequel un agent est incapable de percevoir l'état actuel du système. A défaut, il a une observation incomplète fondée sur l'action qu'il a faite et l'état résultat.

Le but de l'agent est d'obtenir une politique π qui fait correspondre une action a à un historique d'observation (une trajectoire) $h_{1:t} = ((z_1, a_1, r_1), \ldots, (z_{t-1}, a_{t-1}, r_{t-1}), (z_t, -, -))$ afin de maximiser la qualité ou la valeur de π. Dans le cas d'une politique stochastique, $\pi(h_{1:t})$ retourne une distribution de probabilité sur les actions. Toutefois, l'agent reste confronté à sa capacité de mémoire limitée. De ce fait, il doit compresser les données contenues dans l'historique et les représenter par un état interne x_t. L'agent doit apprendre une politique qui fait correspondre à ses états internes des actions.

Comme dans le cas du MDP, la valeur d'une politique peut être définie de différentes manières, selon le critère d'optimalité choisi.

Critères d'optimalité à horizon infini

L'espérance de gain moyen est souvent inadaptée aux POMDPs, car dans certains problèmes et dans certains cas, ne rien faire peut garantir à un agent un gain important dans l'avenir. La politique optimale devient alors de ne rien faire pour toujours [Platzman, 1977].

Tout comme dans le cas du MDP, le critère le plus utilisé reste l'espérance de gain γ-pondéré.

Problématiques

On peut classer les problèmes liés aux POMDPs selon les axes suivants :

1. Est-ce que l'agent connaît le modèle de l'environnement (T) et sa fonction de récompense (R) ?

   Si non, comment les apprendre ?

2. Quel genre d'état interne l'agent met à jour et, de ce fait, quel type de politique recherche-t-il ?

   Nous avons vu que l'agent ne peut pas avoir accès à toute la trajectoire de ses observations. Dans le paragraphe suivant, nous détaillons quelques unes de ces politiques.

3. Quel outil est utilisé pour représenter et apprendre une politique ?

   Il y a plusieurs réponses à cette question qui dépendent de l'approche de résolution choisie :
   - Utiliser la fonction des Q-valeurs pour l'apprentissage par renforcement de type *Q-learning* adapté.

- Si le POMDP est connu, on peut utiliser les techniques qui font appel à l'espace des états probables et se ramener à la résolution du MDP correspondant de manière exacte ou en utilisant des méthodes approchées. Dans ce cas, on utilise la fonction de valeur V.
- Enfin, rechercher directement dans l'espace des politiques un maximum local, en utilisant les dérivées partielles et un ensemble de paramètres θ pour représenter V_π.

Précisons d'ores et déjà qu'il n'existe pas d'algorithme pour résoudre les POMDPs à horizon infini à ce jour. Cependant, il existe des algorithmes capables de trouver des bonnes politiques selon les caractéristiques du POMDP étudié.

2.5.2 Calcul de politique réactive

Cette première approche fait état des situations où l'agent est réactif et n'a pas accès à une mémoire. Trouver une politique réactive revient à calculer une politique pour laquelle l'état interne de l'agent se réduit à l'observation courante. On distingue le calcul de politiques réactives déterministes, et le calcul de politiques réactives stochastiques.

Politique déterministe *vs* Politique stochastique

Dans un système où l'agent ne peut observer que partiellement l'environnement, calculer une politique déterministe sans recourir à une quelconque forme de mémorisation des observations et actions passées est hasardeux. La qualité de la politique ainsi calculée dépend de l'importance des informations cachées par les observations courantes. Le problème n'étant pas forcément markovien, il peut exister des états semblables dans l'environnement qui nécessitent une action différente pour atteindre le but. Nous rappelons que s'il existe une politique déterministe optimale dans le cas des MDPs, Singh et *al.* dans [Singh *et al.*, 1994] ont montré que ce n'était pas le cas dans les POMDPs. De ce fait, une politique calculée de manière déterministe peut avoir des performances décevantes.

Des améliorations sont possibles en s'intéressant à la recherche de politiques stochastiques réactives :

$$\pi : Z \rightarrow \mathcal{P}(A)$$

S'il y a du hasard dans les actions de l'agent, il ne restera pas dans un état absorbant. La figure 2.4 donne un exemple de POMDP à 2 états. Le caractère stochastique de la politique permet à l'agent de choisir parfois des actions différentes dans des situations différentes qui ont pourtant la même apparence pour l'agent. Trouver une politique stochastique réactive optimale est NP-difficile [Papadimitriou et Tsitsiklis, 1987][Littman, 1994b].

Algorithmes

Dans le cas où le modèle n'est pas connu, utiliser une politique réactive stochastique est indispensable afin notamment d'explorer tout l'espace des politiques. Les algorithmes d'apprentissage par renforcement du type *Q-learning* adapté utilisent des politiques stochastiques. C'est également le cas de *SARSA(λ)* [Sutton et Barto, 1998], et de la méthode de montée de gradient [Jaakkola *et al.*, 1995].

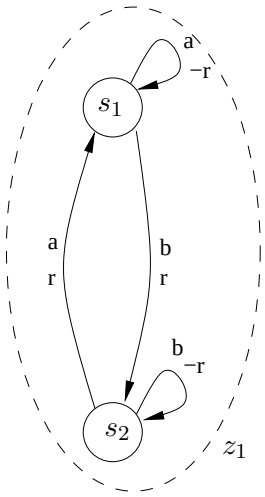


FIG. 2.4 – Exemple de POMDP pour lequel la politique optimale stationnaire est stochastique [Singh *et al.*, 1994].

2.5.3 Calcul de politique avec mémoire

Une manière de se comporter efficacement dans un grand nombre d'environnements est de garder en mémoire les précédentes actions et observations afin de lever les ambiguïtés sur l'identification de l'état courant. C'est le cas par exemple de la politique avec fenêtre d'historique. Il s'agit de définir la mémoire d'état de l'agent par une liste des k plus récentes actions et observations [Cassandra *et al.*, 1994].

Algorithmes

Dans un contexte d'apprentissage par renforcement, Andrew McCallum utilise un arbre de profondeur variable dans lequel est classée l'historique des observations et où chaque branche se termine par l'action que l'agent devrait suivre [McCallum, 1995]. Alain Dutech propose d'utiliser une mémoire contextuelle [Dutech, 1999].

2.5.4 Calcul de politique avec états probables

Dans certains cas, il n'existe pas de mémoire finie suffisante pour définir un comportement optimal. La solution est de choisir une représentation des états qui permette de condenser l'information passée : on parle d'états probables (*information states ou belief-states*)[Smallwood et Sondik, 1973].

Un état probable, $b \in \mathcal{P}(S)$, est une représentation de l'état interne courant de l'agent étant données ses actions et observations passées :

$$b : S \rightarrow [0, 1]$$

$b(s)$ est la probabilité d'être dans l'état de l'environnement s.

Cet ensemble doit être mis à jour après chaque action en fonction de l'état probable précédent b, de la dernière action a et de la dernière observation faite o, à l'aide d'un estimateur d'état SE :

$$b'(s') = SE_{s'}(b,a,o) = P(s'|o,a,b) \quad (2.18)$$

$$= \frac{P(o|s',a,b)P(s'|a,b)}{P(o|a,b)} \quad (2.19)$$

$$= \frac{O(s',a,o)\sum_{s\in S} b(s)T(s,a,s')}{P(o|a,b)} \quad (2.20)$$

où $P(o|a,b)$ permet de normaliser la somme :

$$P(o|a,b) = \sum_{s'\in S} O(s',a,o) \sum_{s\in S} b(s)T(s,a,s')$$

La question soulevée par l'utilisation d'états probables est : comment associer une action à une distribution de probabilité ?

L'état de l'environnement courant n'est pas connu. On calcule la valeur d'une distribution de probabilité b :

$$V^a(b) = \rho(b,a) + \gamma \sum_{b'\in B} \tau(b,a,b')V(b') \quad (2.21)$$

où ρ est la nouvelle fonction de gain prenant en compte tous les états probables :

$$\rho(b,a) = \sum_{s\in S} b(s)R(s,a)$$

et τ est la fonction de transition entre distributions de probabilités :

$$\tau(b,a,b') = \sum_{o\in O|SE(b,a,o)=b'} P(o|a,b)$$

La politique optimale est obtenue en calculant pour chaque distribution de probabilité l'action optimale, c'est-à-dire l'action qui maximise l'équation (2.21).

Ainsi, n'importe quel POMDP discret induit un MDP dont les états sont les états probables. Notons que l'ensemble B de tous les ensembles d'états probables est infini : l'espace à considérer pour construire la politique devient continu.

Algorithmes

Dans le cas où le modèle de l'environnement est complètement connu, Cassandra et al. ont travaillé sur l'adaptation du *Value Iteration* [Cassandra *et al.*, 1994]. Dans la suite de ce chapitre, nous détaillerons les principes de l'algorithme *Witness* [Littman, 1996], qui reste le plus performant à ce jour. Toutefois, son utilisation se restreint à des situations où le nombre d'états ne dépasse pas la centaine et où le nombre d'observations est inférieur à 20.

2.6 Algorithmes de résolution d'un POMDP : avec et sans modèle

Nous développons dans cette section les principes généraux qui permettent de calculer une politique optimale, de manière exacte pour *Witness*, et de manière approchée pour la montée du gradient. Le lecteur intéressé pourra se rapporter à [Littman, 1996] [Cassandra, 1998][Baxter et Bartlett, 2001].

2.6.1 États probables : *Value Iteration*

La méthode du *Value Iteration* [Bellman, 1957] est une manière de calculer la valeur approchée de la fonction de valeur optimale V^*. Une adaptation de l'algorithme *Value Iteration* permet de calculer la valeur de chaque distribution de probabilité. Mais le problème de la continuité de l'espace des distributions de probabilité reste entier.

Or une fonction de valeur à horizon fini est une fonction linéaire et convexe par morceaux [Sondik, 1971]. Pour le cas infini, la fonction de valeur peut être approchée arbitrairement par une fonction linéaire et convexe par morceaux. L'espace d'états peut alors être partitionné à l'aide de vecteurs à $|S|$ dimensions, un vecteur donnant la valeur de chaque état selon sa probabilité. A chaque vecteur correspond une action optimale, et le vecteur choisi pour représenter une région de l'espace d'états est celui qui maximise la valeur cumulée des états. Ainsi, la valeur d'une distribution de probabilité sur les états peut être exprimée de la façon suivante :

$$V_t(b) = \max_{\alpha \in \mathcal{V}_t} \sum_s b(s)\alpha(s) \tag{2.22}$$

où $\mathcal{V}_t$ est un ensemble fini de vecteurs à $|S|$ dimensions. Le problème est maintenant de trouver les vecteurs permettant de calculer la fonction de valeur optimale.

On peut représenter l'ensemble des politiques sous forme d'arbres de décision. La racine d'un arbre correspond à un état initial possible, à chaque nœud correspond une action et chaque branche est fonction d'une observation. Les arbres sont construits de manière incrémentale : à l'étape t, on crée un ou plusieurs arbres de hauteur t dont la racine est une des actions possibles et les fils sont des arbres créés à l'étape $t-1$.

Chaque arbre a son utilité selon la distribution de probabilité initiale. La résolution du problème consiste alors à trouver les arbres optimaux.

Dans l'algorithme du *Value Iteration*, à chaque itération on doit trouver l'ensemble $\mathcal{V}_t$ qui représente V_t^* connaissant l'ensemble $\mathcal{V}_{t-1}$, et trouver une meilleure estimation. La complexité de cette itération rend inutilisable l'algorithme du *Value Iteration*.

2.6.2 États probables : *Witness*

L'adaptation du *Value Iteration* a donné lieu à de nombreux travaux. Toutes les méthodes cherchaient à réduire à chaque étape la construction de l'ensemble des politiques possibles $\mathcal{V}$ en procédant à des éliminations des politiques les plus éloignées de la politique optimale [Smallwood et Sondik, 1973][Cheng, 1988]. Parmi les méthodes les plus intéressantes, *Witness* de [Littman, 1994c] et *Incremental Pruning* de [Cassandra *et al.*, 1997] ont donné des résultats très supérieurs. Toutefois leur complexité demeure trop importante pour espérer travailler avec des modèles de taille moyenne.

Sans rentrer dans les détails, l'intérêt de *Witness* est de construire incrémentalement les arbres, en ne construisant que les arbres nécessaires. L'originalité de l'algorithme consiste en une procédure qui détermine que l'ensemble courant d'arbres est suffisant pour représenter la politique optimale.

États probables : limites et conclusions

Bien qu'*Incremental Pruning* soit plus performant que *Witness* [Cassandra *et al.*, 1997], les algorithmes basés sur la résolution de MDP dans l'espace des états probables sont beaucoup trop complexes pour les utiliser dans des environnements réels et encore moins dans le cadre des systèmes multi-agents. L'utilisation d'états probables n'est donc pas réaliste dans le cas où nos agents ont une connaissance partielle de l'environnement.

2.6.3 Apprentissage par renforcement : montée de gradient

Nous avons vu dans la section (2.4.5) une technique d'apprentissage basée sur l'utilisation des Q-valeurs. Nous proposons dans ce paragraphe de comprendre le principe d'un algorithme de recherche directe dans l'espace des politiques. Les algorithmes de *"montée du gradient"* en sont un exemple.

[Baxter et Bartlett, 2001] et [Baxter *et al.*, 2001] exposent un algorithme de simulation pour générer une estimation biaisée du gradient de la récompense moyenne dans un POMDP. Les auteurs proposent de se placer dans l'espace des politiques stochastiques paramétrées par $\theta \in \mathbb{R}^k$ afin de calculer le gradient de la récompense moyenne, puis d'améliorer la politique en ajustant les paramètres dans la direction du gradient.

Rappel : Calcul du gradient

On rappelle que la dérivée partielle première d'une fonction représente la pente de la tangente au point étudié et selon une dimension à la fois. Le gradient d'une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ se définit comme suit :

$$\overrightarrow{grad(f)} = \overrightarrow{\nabla} \cdot f(x_1, .., x_i, .., x_n) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_i} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} = \sum_i \frac{\partial f}{\partial x_i} \overrightarrow{k_i}$$

où k_i est le i^{eme} vecteur de la base de l'espace étudié.

Le gradient est le vecteur représentant les variations spatiales d'une fonction scalaire f. Le vecteur obtenu aura les quatre propriétés suivantes :

- Ses composantes représentent la variation (pente) de f selon les axes $\overrightarrow{k_i}$.
- Sa norme est la variation maximum de f en fonction de la distance.
- Sa direction est, selon la variation maximum de f, fonction de la distance.
- Le sens indique les valeurs où f augmente.

De ces propriétés, [Baxter et Bartlett, 2001] profitent du fait que le gradient indique la direction où la dérivée de f est la plus élevée pour trouver directement une politique dans l'espace des politiques stochastiques.

Principe de la montée de gradient

La montée du gradient tente de trouver un point $(x_k)_{k \in \mathbb{N}}$ où la fonction de mesure des performances est maximale. Pour cela, il faut d'abord estimer la valeur du gradient en un point,

c'est l'objet de l'algorithme GPOMDP [Baxter et Bartlett, 2001], et répondre à un certain nombre de questions :

1. Quelle fonction de mesure de performance V choisir (critère d'optimalité) ? Quels paramètres utiliser ?

   Les auteurs suggèrent d'utiliser la formule suivante pour mesurer la performance d'une politique :

   $$V(\theta) = \sum_{s \in S} P_\theta(s) r(s)$$

   où $P_\theta(s)$ est la probabilité d'être dans l'état s une fois le système à l'état stationnaire. Précisons que la récompense r ne dépend que de l'état s.

   On remarquera qu'il n'est nullement question de faire appel au principe de Bellman dans cette nouvelle expression de V.

2. Comment calculer le gradient de manière expérimentale ?

   En effet, le modèle de l'environnement n'est pas connu. L'évaluation du gradient est faite par :

   $$\overrightarrow{z_{t+1}} = \beta \overrightarrow{z_t} + \frac{\overrightarrow{\nabla} \mu_{at}(\theta, o_t)}{\mu_{at}(\theta, o_t)} \tag{2.23}$$

   $$\overrightarrow{\nabla_{t+1}} = \overrightarrow{\nabla_t} + r_{t+1} \overrightarrow{z_{t+1}} \tag{2.24}$$

   où o_t est l'observation à l'instant t, a_t l'action choisie et r_{t+1} la récompense reçu. Bien évidemment, μ est la politique stochastique paramétrée par θ et souvent définie comme suit :

   $$\mu_a(\theta, o) = \frac{e^{\theta_{a,o}}}{\sum_{b \in A} e^{\theta_{a,o}}}$$

   Remarquons que sa forme facilite le calcul des dérivées partielles.

3. Sous quelles conditions peut-on utiliser l'algorithme ? Pour information (car nous n'entrerons pas dans la théorie menant à cet algorithme), les hypothèses suivantes doivent être satisfaites pour l'utilisation de GPOMDP :

   (a) Les dérivées

   $$\left[\frac{\partial \mu_a(\theta, o)}{\partial \theta_k} \right]_{k=1 \ldots K}$$

   existent pour tout $a \in \mathbb{A}$, $o \in \Omega$ et $\theta \in \mathbb{R}^K$.

   (b) Les rapports

   $$\left[\frac{|\frac{\partial \mu_a(\theta, o)}{\partial \theta_k}|}{\mu_a(\theta, o)} \right]_{k=1 \ldots K}$$

   sont uniformément bornés par $B < \infty$, pour tout $a \in \mathbb{A}$, $o \in \Omega$ et $\theta \in \mathbb{R}^K$.

   (c) Les magnitudes des récompenses, $|r(s)|$, sont uniformément bornées par $R < \infty$ pour tous les états s.

   (d) Chaque $P(\theta), \theta \in \mathbb{R}$, a une unique distribution stationnaire, $\pi(\theta)$.

Algorithme 2.6 OLPOMDP$(\beta, T, \theta_0) \rightarrow \mathbb{R}^K$

Entrée:

- $\beta \in [0, 1)$.
- $T > 0$.
- Des valeurs initiales du paramètre $\theta_0 \in \mathbb{R}^K$.
- Des politiques paramétrées randomisées $\{\mu(\theta, .) : \theta \in \mathbb{R}^K\}$ vérifiant les hypothèses a et b (ci-dessus).
- Un POMDP dont les récompenses vérifient l'hypothèse c, et qui, quand il est contrôlé par $\mu(\theta, .)$ génère des matrices stochastiques $P(\theta)$ satisfaisant l'hypothèse d.
- Des tailles de pas $\alpha_t, t = 0, 1, \ldots$ satisfaisant $\sum \alpha_t = \infty$ et $\sum \alpha_t^2 < \infty$.
- Un état de départ arbitraire (inconnu) s_0.

1: Soit $z_0 = 0$ ($z_0 \in \mathbb{R}^K$).

2: **Pour** $t = 0$ à $T - 1$ **Faire**

3: Observer o_t (généré d'après $v(s_t)$).

4: Générer une commande a_t d'après $\mu(\theta, o_t)$

5: Observer $r(s_{t+1})$ (où le prochain état s_{t+1} est généré d'après $T(s_t, a_t, s_{t+1})$)

6: $z_{t+1} \leftarrow \beta z_t + \frac{\nabla \mu_{a_t}(\theta, o_t)}{\mu_{a_t}(\theta, o_t)}$

7: $\theta_{t+1} \leftarrow \theta_t + \alpha_t r(s_{t+1}) z_{t+1}$

8: **Fin Pour**

Sortie: θ_T

Algorithme de la montée du gradient

A titre indicatif, nous présentons succinctement l'algorithme (2.6) OLPOMDP (On-Line POMDP) qui permet à la fois d'estimer le gradient (principe de l'algorithme GPOMDP) et d'améliorer la politique en conséquence.

Nous invitons les lecteurs intéressés à se reporter aux articles [Baxter et Bartlett, 2001] et [Baxter *et al.*, 2001].

Travaux connexes

Parmi les travaux sur la recherche directe dans l'espace des politiques, on distingue les algorithmes qui apprennent une politique, soit :

- en estimant $V(\pi)$ avec un modèle [Kearns *et al.*, 2000],
- en estimant $\frac{\partial V(\pi)}{\partial \theta_\pi}$ avec un modèle [Ng *et al.*, 1999] [Baxter *et al.*, 2001], ou
- en estimant $\frac{\partial V(\pi)}{\partial \theta_\pi}$ sans modèle [Williams, 1992] [Kimura *et al.*, 1997] [Baxter et Bartlett, 2001] [Baxter *et al.*, 2001].

Apprentissage : limites et conclusions

A ce jour, aucune technique d'apprentissage semble mieux adaptée qu'une autre, toutefois la recherche directe de politiques a le grand avantage de ne nécessiter que très peu de place mémoire.

Notre travail se place dans le cas où le système étudié est connu, cependant les techniques

d'apprentissage par renforcement sont très proches de certains principes de planification. Dans notre contexte, où nos agents ne perçoivent que partiellement l'environnement, l'apprentissage comble le manque d'informations nécessaires au calcul d'une politique. Les méthodes d'apprentissage resteront une source d'inspiration lors de notre recherche d'algorithmes de conception de systèmes multi-agents.

2.6.4 Complexité

Jusqu'à présent nous avons vu comment résoudre un POMDP dans différents cas de figures : méthode approchée, méthode exacte, modèle connu ou inconnu. Nous résumons dans ce paragraphe les résultats de complexité connus.

Horizon fini

Comme nous l'avons vu, à horizon fini, l'étude des POMDPs peut se transformer en l'étude d'un MDP complètement observable en redéfinissant l'état s_t à l'étape t à partir de son historique d'observation h_t. La cardinalité de l'espace des états redéfinis (et de ce fait les algorithmes de résolution) augmente exponentiellement avec l'horizon $\mathcal{T}$.

L'autre solution est de reformuler un POMPD en un MDP complètement observable en utilisant cette fois-ci des états probables. Le MDP résultant dispose d'un espace d'état continu, et d'une fonction de probabilité de transition continue. Si la préimage[12] de chaque observation possible a une cardinalité limitée par une constante k, et si $k \geq 3$, le problème demeure NP-difficile (le cas $k = 2$ reste ouvert). Cependant, l'espace des états est essentiellement de dimension k, le problème peut être résolu avec une précision ϵ-optimale en un nombre d'opérations arithmétiques polynomial en : n le nombre d'états, m le nombre d'actions, $1/\epsilon$ l'inverse de la précision et $\mathcal{T}$ l'horizon [Burago *et al.*, 1996].

Horizon infini

A horizon infini, les problèmes POMDPs ne peuvent pas être résolus en les réduisant à des MDPs parfaitement observables (nombre d'états infinis). Dans [Lovejoy, 1991], les auteurs ont montré que résoudre un POMDP à horizon infini n'était pas calculable. [Madani *et al.*, 1999] montrent que sous les critères d'optimisation différents (espérance de gain totale, pondéré, ou moyen) le problème est non décidable.

2.6.5 Conclusion

Dans cette section, nous avons fait le tour des techniques de résolution des POMDPs de manière exhaustive. Nous savons qu'à horizon infini, il n'existe pas d'algorithme de résolution du problème du POMDP. En revanche, certains algorithmes proposent des méthodes de résolution approchées qui tentent de contourner les difficultés de l'entreprise.

Nous savons maintenant que bien que nos agents se rapprochent des caractéristiques des POMDPs, il n'est pas raisonnable d'en utiliser les algorithmes pour concevoir des politiques puisque leurs complexités les rendent inutilisables dans notre contexte de recherche multi-agents.

[12] La préimage correspond à l'image réciproque d'une observation par la fonction d'observation $\mathcal{O}$.

2.7 Modèles décisionnels de Markov et systèmes multi-agents

La caractéristique multi-agents induit une prise de décision de chaque agent. Certains modèles décisionnels rendent compte de cette composante complexe. C'est le cas des Jeux de Markov, des MMDP (MultiAgent Markov Decision Processes), et des DEC-POMDP (Decentralized Partially Observable Markov Decision Processes). Nous proposons d'identifier les différences et similitudes de ces modèles théoriques.

2.7.1 Jeux de Markov

Les jeux de Markov, également appelé jeux stochastiques, sont des modèles de décision séquentiels qui généralisent les processus décisionnels de Markov [Owen, 1982]. Ils modélisent des agents (ou joueurs) décidant d'une action simultanément ou alternativement (Jeux de Markov alternatifs), ainsi les récompenses et transitions sont déterminées par les actions simultanées des joueurs. Les jeux de Markov ont tout d'abord été étudiés par Shapley [Shapley, 1953]. Le contexte n'est plus ici de maximiser une récompense face à un environnement, mais de maximiser une récompense en face d'un adversaire optimal, les récompenses des agents sont de ce fait individuelles. Néanmoins, il y a de grandes similarités entre le problème de trouver une politique optimal pour un MDP et celle de trouver une politique optimale pour un jeu [Littman, 1996].

Définition

Définissons formellement un jeu de Markov.

Définition 11 (Jeu de Markov) :
Un jeu de Markov est un t-uple $G = \langle S, N, A_1, \ldots, A_n, P, u_1, \ldots, u_n \rangle$ où
- S est un ensemble fini d'états,
- N est un ensemble fini de n joueurs,
- A_i est un ensemble fini d'actions disponibles pour le joueur i. On pose $A = A_1 \times \cdots \times A_n$ l'ensemble des actions jointes.
- $T : S \times A \to \mathcal{P}(S)$ est la distribution de probabilité. On pose T(s,a,s') la probabilité d'aller de l'état s à l'état s' après avoir effectué l'action jointe a.
- $u_i : S \times A \to \mathbb{R}$ est la fonction d'utilité réelle pour le joueur i (*i.e.* la fonction de récompense). □

Ainsi chaque agent (ou joueur) i tente de maximiser son espérance de gain de manière individuelle. Le critère de performance γ-pondéré se définit comme suit :

$$E(\sum_{j=0}^{\infty} \gamma^j u_{i,t+j})$$

avec $u_{i,t+j}$ la récompense reçu j pas dans le futur par l'agent i. Notons qu'il est possible de définir un critère non pondéré, cependant dans ce cas tous les jeux de Markov n'ont pas de stratégie optimale [Owen, 1982].

Stratégie

Définition 12 (Vecteur de stratégies) :
Dans un jeu de Markov, une stratégie ou une politique pour un joueur i est une fonction $\pi_i : S \to (P)A_i$. Un ensemble de stratégies pour tous les joueurs $\pi = \pi_1 \times \cdots \times \pi_n$ est appelé un vecteur de stratégies ou vecteur de politiques. □

Dans un jeu de Markov, le nombre de politiques est fini, mais croît exponentiellement avec le nombre d'agents. Pour avoir une notion de la qualité d'une politique, comme dans les processus décisionnels de Markov, il faut être capable de l'évaluer. Dans le cas d'un système à horizon infini, la fonction de valeur d'un vecteur de politiques π commençant dans l'état s pour le joueur i est défini par l'espérance des gains pondérés :

$$V_i^{\pi}(s) = E(\sum_{j=0}^{\infty} \gamma^j u_i(s_j, \pi(s_j)) | s_0 = s)$$

où l'espérance est sur les trajectoires $s_0, s_1, \ldots$ dans l'espace des états S, et γ est le facteur de pondération. Évidement, calculer la valeur d'un vecteur de stratégies n'est pas particulièrement réalisable en utilisant l'équation précédente. On lui préférera la forme habituelle de l'équation de récurrence (2.15).

Équilibre de Nash

L'équilibre de Nash, initialement défini pour la théorie des jeux classique, a été étendu aux jeux de Markov. On l'appelle également équilibre parfait de Markov (MPE[13]).

> **Définition 13 (Équilibre parfait de Markov) :**
> Un vecteur de stratégies ou vecteur de politiques π, est appelé équilibre parfait de Markov (ou simplement équilibre de Nash) si et seulement si pour chaque joueur i :
>
> $$V_i^{\pi}(s) \geq V_i^{\pi_{-i} \times \pi_i'}(s), \ \forall s \in S, \ \forall \pi_i'$$

En d'autres termes, un vecteur de stratégies est un équilibre parfait de Markov si des déviations unilatérales sont dommageables. Tout jeu de Markov a au moins un équilibre de Nash, son calcul est en général très compliqué.

Méthodes de résolution pour les jeux de Markov

Les jeux de Markov peuvent être résolus à l'aide des algorithmes de *Value Iteration* ou *Policy Iteration* [Littman, 1996]. Des algorithmes d'apprentissage de politiques ont également été explorés. On distingue les algorithmes mettant en jeux 2 joueurs [Littman, 1994a] et ceux qui s'attaquent à une population comprenant n agents [Hu et Wellman, 1998a] [Hu et Wellman, 1998b].

Conclusion

La difficulté majeure des jeux de Markov reste, une fois de plus, la complexité de résolution des techniques employées. Notre thèse s'interesse à l'étude de comportements coopératifs. De ce fait, nous ne nous situons pas dans les approches traditionnellement étudiées dans la théorie des jeux de Markov pour laquelle les agents ont souvent des approches indépendantes, voire concurrentes. Nous chercherons à tirer profit de cette composante coopérative dans la conduite de notre travail.

Enfin, remarquons qu'une caractéristique essentielle des jeux de Markov, est l'expression individuelle de la fonction d'utilité ou fonction de récompense. Celle-ci dépend de l'état du système et des actions de tous les agents. A ce titre, le formalisme des jeux de Markov permet l'expression de buts locaux individuels qui ne sont pas nécessairement coopératifs.

[13] Acronyme de la version anglaise "Markov Perfect Equilibrium"

2.7.2 MMDP : le travail de Boutilier

Dans [Boutilier, 1999], Craig Boutilier propose un modèle pour résoudre la planification dans les systèmes multi-agents fondé sur le calcul de politiques jointes à partir d'actions jointes. Nous développons ici le modèle proposé par l'auteur et mettons en avant les caractéristiques intéressantes pour la suite de notre travail.

Formalisme

Définition 14 (MMDP) :
Un Multi-agent MDP (MMDP) M est defini par $\langle \alpha, \{A_i\}_{i \in \alpha}, S, Pr, R \rangle$:

- α est le nombre d'agents du système.
- A_i est l'ensemble fini des actions individuelles de l'agent i.
- A est l'ensemble des actions jointes. $A = \times A_i$ représente l'exécution des actions a_i par chaque agent i.
- S, Pr, R sont les mêmes que dans un MDP, à l'exception de Pr la fonction de probabilité de transitions qui utilise les actions jointes $\langle a_1, ..., a_n \rangle$ □

Ce modèle peut être appréhendé comme un MDP avec de grands espaces d'états et d'actions : l'espace des actions devient celui des actions jointes. Sa résolution en utilisant des algorithmes classiques comme le *Value Iteration* permet la détermination d'une politique jointe $\Pi = \langle \pi_1, \ldots, \pi_n \rangle$.

L'utilisation de la politique jointe peut se faire de deux façons : soit en faisant appel à un contrôleur central qui décide pour chaque agent l'action qu'il doit effectuer en regard de Π, soit en individualisant la politique jointe optimale en politiques individuelles π_i. Dans le contexte système multi-agents qui nous intéresse, les agents sont des entités autonomes situées dans un environnement et capables d'interagir. De ce fait, la deuxième utilisation a notre préférence.

Problématique soulevée

La problématique soulevée par Craig Boutilier concerne également cette deuxième utilisation : les agents perçoivent une situation complètement observable et doivent individuellement décider d'une action optimale à produire en suivant la politique jointe calculée. Comme nous l'avons vu dans la section (2.3), bien qu'il n'existe qu'une seule fonction de valeur optimale V^*, plusieurs politiques optimales π^* peuvent lui correspondre. C'est également le cas dans le MMDP. Un agent peut être amené à choisir entre deux actions optimales. Si parfois choisir une action optimale parmi celles disponibles n'a pas de conséquences importantes, il existe des situations où un mauvais choix peut conduire à de fortes contre-performances du système dans son intégralité. La figure 2.5 illustre une situation où la coordination des deux agents A_1 et A_2 est nécessaire.

Dans l'état $s1$, l'action optimale jointe est "A_1 *décide a quelle que soit l'action de l'agent* A_2". En $s2$, les deux agents doivent effectuer la même action sous peine de se voir infliger une récompense négative. Ainsi, on distingue rapidement le problème : si les actions a et b sont optimales pour les agents A_1 et A_2, comment coordonner les agents pour qu'ils choisissent la même action ?

Value Iteration étendu

Craig Boutilier s'est intéressé à une méthode de résolution mettant en avant la coordination explicite des agents. Rappelons que les agents ont tous la même fonction d'utilité ou de récompense R, un agent perçoit complètement le système. L'auteur propose un algorithme du *Value*

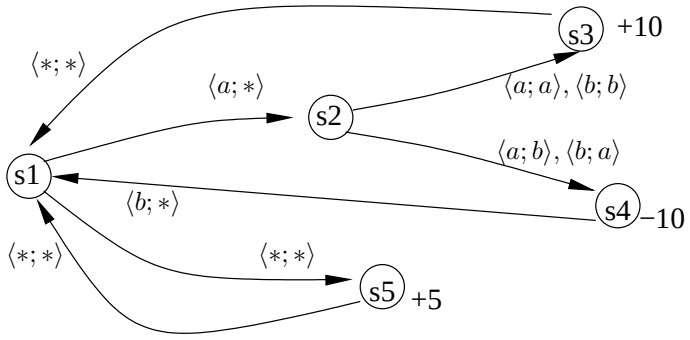


FIG. 2.5 – Exemple d'une situation de coordination nécessaire.

Iteration avec une extension d'état. L'idée repose sur l'identification des actions jointes qui induiraient des problèmes de coordination. Un problème de coordination (CP) se produit lorsque chaque agent décide d'une action individuelle optimale a_i parmi l'ensemble des actions individuelles potentiellement optimales (PIO) et que l'action jointe $\langle a_1, \ldots, a_n \rangle$ est sous-optimale. Ainsi, l'ensemble fini d'états S est étendu sous la forme $\langle S, C \rangle$. C correspond à l'état du mécanisme de coordination choisi.

Mécanisme de coordination explicite

Un mécanisme de coordination est un protocole par lequel les agents restreignent leur attention à un sous-ensemble de leurs actions-PIO. Un mécanisme possède un état qui résume les aspects intéressants de l'histoire de l'agent, et une règle de décision pour sélectionner les actions en fonction de l'état du mécanisme. Ces mécanismes peuvent garantir une coordination immédiate, une coordination éventuelle, ou ne fournir aucune de ces assurances. Boutilier utilise dans son article un mécanisme aléatoire ("randomization"), qui repose sur une sélection uniforme et aléatoire d'une action-PIO jusqu'à ce que la coordination soit terminée. L'agent fera alors l'action choisie pour toujours. Ainsi, les deux états les plus souvent utilisés pour constituer l'ensemble C sont c et u, respectivement pour signifier que l'état étendu est coordonné ou non.

Fonction de transition et fonction de valeur

Que devient la fonction de transition pour un état étendu ? Tout comme dans le MMDP, chaque action provoque une transition d'état du système, tandis que l'état de coordination change de u à c seulement si les agents se sont coordonnés.

De la même façon qu'il est nécessaire de connaître le modèle du monde pour utiliser un MMDP, il est également nécessaire de savoir calculer les probabilités de transition avec les états étendus.

Ainsi, la fonction de valeur optimale V^* ne dépend plus uniquement du système seul mais aussi de l'état du mécanisme de coordination. C'est, par conséquent, aussi le cas de la politique optimale calculée en utilisant l'algorithme du *Value Iteration* pour des états étendus.

Conclusion

Dans cet article, Boutilier a proposé un formalisme de MDP multi-agents, dans lequel plusieurs agents partagent la même fonction de récompense et la même fonction de valeur. Les agents perçoivent tout le système ainsi que les actions de chacun. Cette omniscience permet à l'auteur de mettre en place un protocole de coordination explicite fondé sur l'extension des états comme nous l'avons vu précédemment. Étendre l'espace des états implique un accroissement de la taille du problème à résoudre. Par exemple, pour un problème académique où deux agents doivent collecter des objets dans un environnement discrétisé sous forme de grille, 8 CP provoquent une augmentation d'un facteur 256 pour 900 états, ce qui nous amène à la gestion d'un total de 230400 états. Nous atteignons là les limites de ce formalisme.

Ce travail nous apprend qu'il n'est pas souhaitable d'enrichir l'espaces des états en y incluant des états de coordination explicites, si l'on souhaite utiliser un grand nombre d'agents. De plus, l'omniscience des agents tant au niveau de leur perception que de la connaissance des lois du monde, va à l'encontre des principes de localité et d'autonomie que nous avons fixé pour la conception de nos agents.

2.7.3 DEC-POMDP : le travail de Bernstein

Tandis que la majorité des travaux sur l'utilisation des modèles de Markov utilise un contrôle centralisé, Bernstein et al.[Bernstein *et al.*, 2000] s'intéressent à l'étude de la complexité lorsque le contrôle est décentralisé.

Ils se préoccupent, plus précisément, de la planification centralisée pour des agents distribués qui ont accès à des informations incomplètes. Les auteurs ont étendu le modèle du POMDP pour permettre à des agents de percevoir chacun leur observation et de fonder leur décision sur ces observations. La fonction de transition (T) ainsi que la fonction de récompense (R) restent jointes, c'est-à-dire qu'elles dépendent des actions de tous les agents. Ils ont appelé ce modèle : DEC-POMDP pour *DECentralized Partially Observable Markov Decision Process*. Ce modèle a déjà été utilisé dans la littérature [Peshkin *et al.*, 1996] [Bernstein *et al.*, 2000].

Formalisme

Le DEC-POMDP permet un contrôle décentralisé. Dans ce modèle à chaque pas de temps, chaque agent reçoit une observation locale et choisit une action. La fonction de transition ainsi que la fonction de récompense dépendent du vecteur des actions de tous les agents.

Définition 15 (DEC-POMDP) :
Un DEC-POMDP pour deux agents est défini par $\langle S, A_1, A_2, P, R, \Omega_1, \Omega_2, O, T, K\rangle$ où :

- S est un ensemble d'états fini.
- A_1 et A_2 sont des ensembles d'actions finis pour respectivement les agents 1 et 2.
- T est la table des probabilités de transition. $T(s, a_1, a_2, s')$ représente la probabilité de transition de l'état s à s' en effectuant les actions a_1, a_2. Avec $s, s' \in S$, $a_1 \in A_1$ et $a_2 \in A_2$.
- R est la fonction de récompense. $R(s, a_1, a_2, s')$ est un réel représentant la récompense obtenue en effectuant les actions a_1, a_2 dans l'état s en arrivant dans l'état s'. Avec $s, s' \in S^2$, $a_1 \in A_1$ et $a_2 \in A_2$.
- Ω_1 et Ω_2 sont les ensembles finis des observations.

- O est la table des probabilités d'observation. $O(s, a_1, a_2, s', o_1, o_2)$ représente la probabilité d'observer o_1, o_2 en effectuant les actions a_1, a_2 dans l'état s en arrivant en s'. Avec $s, s' \in S$, $a_1 \in A_1$, $a_2 \in A_2$, $o_1 \in \Omega_1$ et $o_2 \in \Omega_2$.
- $\mathcal{T}$ est un entier positif représentant l'horizon.
- K est un réel représentant le seuil de valeur. □

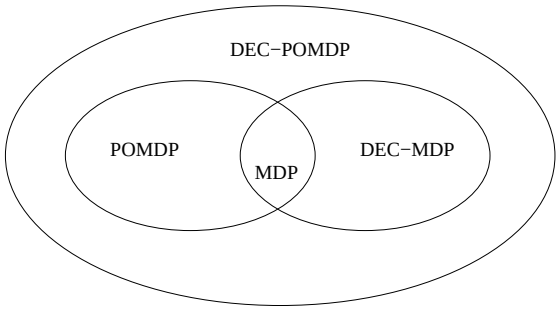


FIG. 2.6 – Relations entre les différents modèles.

Un DEC-POMDP généralise un POMDP en permettant un contrôle décentralisé par les agents qui ensemble ne parviennent pas à observer complètement le système. Dans le cas où les observations jointes des agents permettent une information complète le modèle devient un DEC-MDP.

Dans cet article, Bernstein et al. ont montré que le DEC-POMDP et le DEC-MDP à horizon fini sont NEXP-difficiles pour un nombre d'agents n $\geq$ 2. Lorsque le problème est limité à un horizon plus petit que le nombre d'états, les problèmes sont NEXP-complet. Tout comme pour le POMDP, dans le cas d'un horizon infini, le DEC-POMDP et le DEC-MDP sont non décidables pour certains critères d'optimalité [Madani *et al.*, 1999].

Conclusion

Le DEC-POMDP formalise le contrôle décentralisé des agents du système : la fonction d'observation est individuelle. En cela, il se rapproche des propriétés d'autonomie des agents qui constituent les SMA. En revanche, comme pour le MMDP, la fonction de récompense est globale.

Les résultats théoriques obtenus nous révèlent qu'il est "impossible" de résoudre un DEC-POMDP dès que le nombre d'agents est supérieur ou égal à 2. Résoudre un problème de type DEC-POMDP ne sera alors possible qu'en utilisant des méthodes approchées. La difficulté sera alors de qualifier la qualité des solutions obtenues.

2.8 Conclusions

Dans ce chapitre, nous avons entrepris la réalisation d'un état de l'art des modèles décisionnels de Markov susceptibles de nous intéresser dans notre recherche d'une méthode de conception descendante de systèmes multi-agents.

Le formalisme des MDPs a, à ce jour, des propriétés de convergence vers un plan optimal mono-agent et possède des méthodes de résolution de complexité des plus intéressantes. Son utilisation implique une connaissance parfaite de l'environnement. Dans un univers multi-agents, leur application est envisageable en dépit des nouvelles contraintes de complexité que cela entraînent. On parlera alors de MMDP et de DEC-MDP. Nous avons également discuté des jeux de Markov, et de la qualification d'équilibre de Nash de politiques lorsque les agents possèdent des récompenses individuelles. Cette notion d'équilibre apparaîtra de nouveau dans la suite de ce manuscrit.

Les POMDPs prennent en compte l'observabilité partielle et de ce fait, sous certaines conditions d'utilisation, les résoudre devient un problème non décidable. Dans un univers multi-agents, Bernstein et al. ont montré que la complexité dans des conditions favorables d'utilisation (horizon fini) du DEC-POMDP est NEXP. Ce résultat nous projette immédiatement vers la recherche de méthodes approchées pour le calcul de politiques de qualités intéressantes.

Enfin, les techniques d'apprentissage que nous avons présentées nous ont montré qu'il était possible d'obtenir des politiques de bonne qualité (qui peuvent être optimales) sans connaître l'évolution du système par avance. Bien que notre hypothèse de travail nous donne accès à ces informations, nous verrons que l'apprentissage sera une source d'inspiration dans notre étude, il nous permettra entre autres de pallier le manque d'informations disponibles sur les perceptions des agents.

Notre étude se dirige naturellement vers une réflexion sur des techniques de résolution mêlant la relative simplicité des algorithmes de résolution d'un MDP et les propriétés de perception partielles du POMDP.

Chapitre 3

Modélisation et simulation d'un phénomène réel

Après avoir parcouru et approfondi certains modèles décisionnels de Markov, nous présentons dans ce chapitre l'étude réalisée avec Samuel Venner, étudiant en thèse dans le laboratoire de biologie du comportement à l'Université Henri Poincaré, sur la modélisation et la simulation du comportement d'une araignée orbitèle. Il s'agit de reproduire l'activité de tissage d'une araignée, de son état initial (après sa mue) à son état final (la ponte), en utilisant un processus décisionnel de Markov (MDP), selon un critère d'optimalité.

L'utilisation d'un modèle de décision stochastique à horizon infini tel que le MDP permet de déterminer la politique optimale que devrait suivre l'araignée. Ici, l'hypothèse biologique formulée est que l'araignée devrait se comporter de manière à maximiser son espérance de reproduction, c'est-à-dire qu'elle devrait maximiser ses gains énergétiques afin de parvenir le plus rapidement possible à l'état de ponte, tout en restant en vie. A notre connaissance, les travaux sur la modélisation et la simulation de comportements biologiques n'ont, à ce jour, été étudiés qu'à travers l'utilisation de modèles théoriques de programmation dynamique à horizon fini [Mangel et Clark, 1988][Houston et McNamara, 1999][Clark et Mangel, 1988]. Or, nous sommes dans une configuration de problème où nous ne connaissons pas cet horizon, c'est-à-dire le nombre d'étapes nécessaire pour atteindre le but.

Eco-éthologie

L'éco-éthologie est définie comme l'étude des comportements des espèces animales et de leurs relations avec leur milieu. Aussi l'approche éco-éthologique du comportement a conduit à formuler l'hypothèse selon laquelle l'histoire évolutive aurait façonné l'animal, via la sélection naturelle, de manière à ce qu'il possède des règles de fonctionnement (règles de prises de décision) lui permettant de se comporter au mieux par rapport à la situation du moment (cette situation étant définie par l'état interne de l'individu et par le milieu dans lequel il est). Ceci lui permettrait à long terme de maximiser son succès reproducteur (fitness). C'est dans ce contexte éco-éthologique que Samuel Venner a étudié le comportement de construction des toiles chez *Zygiella x-notata*, une araignée orbitèle (à toile géométrique).

Modélisation, simulation : méthodologie

La simulation d'un phénomène biologique dans son ensemble est une tâche difficile et complexe. Pour simuler un phénomène, il faut être capable d'en écrire le modèle, de le modéliser. Cela implique de déterminer les paramètres qui reproduiront le phénomène que l'on veut étudier et les relations entre ces derniers. Nous nous intéressons, ici, à l'activité de tissage d'une araignée à des fins de reproduction. Bien sûr, de part leur quantité et la difficulté à les déterminer, il est impossible de prendre en compte tous les paramètres qui régissent l'évolution de l'individu. Dans ces conditions, la méthode de travail consiste à choisir judicieusement un nombre limité de paramètres afin d'en analyser les influences et les implications une fois le comportement simulé. Nous avons choisi de considérer le comportement optimal de construction de toiles successives que devrait suivre une araignée en faisant varier trois paramètres : la quantité de proies disponibles, le risque de prédation et la taille des toiles que peut tisser l'araignée.

Organisation du chapitre

Ce chapitre s'organise de la façon suivante. Dans un premier temps (section (3.1)), nous précisons la problématique biologique soulevée qui nous permet d'identifier le critère d'optimisation que devra respecter la résolution du MDP. Puis, nous décrivons les paramètres qui interviennent dans la conception du processus décisionnel de Markov (3.2). Il s'agit d'en préciser les états, les actions, la fonction de transition probabiliste ainsi que la fonction de récompense. Enfin, nous présentons les résultats des simulations effectuées (3.3), et mettons en évidence les différences observées expérimentalement par Samuel Venner (3.4).

3.1 Problématique

Chez de nombreuses espèces d'araignées, la prise alimentaire (gain énergétique) des femelles adultes influence la vitesse de leur croissance, le moment de la ponte, le nombre et/ou la taille des œufs produits, ce qui pourrait influencer à long terme leur fitness. Les prises de décision relatives au comportement alimentaire pourraient donc avoir des conséquences à court terme sur le gain énergétique des araignées et à long terme sur leur succès reproducteur. Dans ces conditions et selon les hypothèses de l'éco-éthologie, les araignées devraient se comporter selon des règles leur permettant, en particulier, de maximiser leur vitesse de gain net énergétique.

Les araignées orbitèles sont des prédateurs chassant à l'affût en utilisant leur toile (piège) pour capturer leurs proies. Ces araignées sont capables de moduler leur comportement de construction (moment de tissage dans la journée, investissement en temps et en énergie consacré à la construction d'une toile, structure du piège, fréquence de tissage des toiles successives), aussi bien en fonction de leur état interne (disponibilité en soie, état énergétique), qu'en fonction de facteurs environnementaux abiotiques (température, humidité) et biotiques (présence de proies, de congénères).

3.1.1 Gestion des coûts, risques et bénéfices

La construction d'un piège peut également être appréhendée en terme de coûts, de risques et de bénéfices :

- les coûts : la construction représente un investissement en énergie (production et mise en place de la soie) et en temps. L'investissement en énergie devrait être d'autant plus grand

que la quantité de fils de soie utilisée pour la mise en place du piège est importante. De plus, la masse de l'araignée croît durant toute sa période de croissance. La masse de l'araignée devrait également intervenir dans l'investissement nécessaire à la construction de la toile : plus l'araignée est lourde, plus elle doit dépenser de l'énergie pour se déplacer.

- les risques : durant la construction du piège, l'araignée se trouve exposée aux aléas de l'environnement (présence de parasites, de prédateurs...). Aussi, plus la durée de tissage est longue, plus les risques qui y sont associés pourraient être importants. D'autre part, ces risques pourraient varier au cours du temps (sur un cycle de 24h par exemple), le moment du tissage pourrait alors s'avérer important.
- les bénéfices : le piège permet la capture de proies et assure un retour énergétique. Le gain énergétique dépend en partie de l'efficacité du piège. Cette efficacité dépend de facteurs abiotiques (température, humidité), de l'orientation de la toile, de son âge (la glu perdant de son efficacité avec le temps, et la toile pouvant être partiellement détruite suite à des contacts avec les proies ou d'autres éléments du milieu), et dépend de sa structure (nombre de rayons, surface de capture, distance séparant deux tours de spire consécutifs).

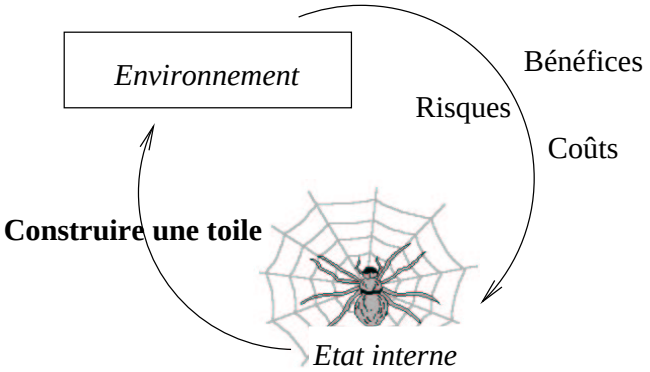


FIG. 3.1 – Les différents paramètres intervenant dans la gestion de la construction successive de toiles.

On peut alors considérer que l'araignée est confrontée à un problème de gestion de la construction de sa toile (figure 3.1) : en fonction des coûts, des risques et bénéfices associés aux différents types de toiles qu'elle peut réaliser et en fonction de la situation du moment, l'araignée devrait décider de tisser ou de ne pas tisser un piège, et si elle tisse, du moment de tissage, de l'investissement à y consacrer (durée du tissage et quantité de soie utilisée) et de la façon d'utiliser cet investissement (structure du piège).

3.1.2 Hypothèse de travail : que faut-il optimiser ?

Pour les araignées orbitèles, la construction d'un piège nécessite, en premier lieu, le choix d'un site adéquat dans l'environnement. Quitter un tel site pour s'installer ailleurs présente un coût élevé et les femelles adultes, restant assez longtemps en un même site reconstruisent pratiquement quotidiennement leur piège. Le problème de gestion précédemment évoqué devient donc un problème de gestion de constructions successives.

Ainsi, l'hypothèse biologique proposée est que, pour accéder au meilleur niveau de production de descendants, l'araignée devrait gérer la construction des toiles successives en limitant les risques de mourir (par prédation ou de faim) et en maximisant la vitesse de gain net énergé-

tique. Nous nous proposons de tester cette hypothèse chez *Zygiella x-notata* en modélisant la construction des toiles successives à l'aide d'un MDP sur un horizon infini, puis en confrontant les prédictions quantitatives obtenues au comportement de construction observé chez des araignées femelles adultes.

3.2 Modèle théorique : MDP

Comme l'illustre la figure 3.1, l'état interne de l'araignée influence les prises de décision relatives à la construction d'une toile qui elle-même modifie en retour son état interne (dépenses et gains énergétiques). Nous avons là un système dynamique. D'autre part, l'environnement dans lequel l'araignée construit ses toiles est complexe et variable dans le temps et dans l'espace, aussi, les conséquences de la réalisation d'une toile sont probabilistes (l'araignée peut être tuée par un prédateur au court du tissage, elle peut ne rien capturer ou capturer un nombre variable de différents types de proies). Nous considérerons ici les hypothèses biologiques suivantes :

- Les facteurs environnementaux (biotiques et abiotiques) sont constants.
- Les différents paramètres (état énergétique, efficacité du piège) varient de manière discrète par intervalle de 24 heures.
- L'araignée décide de la réalisation d'une action par jour.

Ainsi conformément à la définition d'un processus décisionnel de Markov fini, il nous faut préciser :

- un ensemble fini d'états (de configurations) ($\mathcal{S}$),
- un ensemble fini d'actions ($\mathcal{A}$),
- une fonction de transition probabiliste ($\mathcal{T}$),
- une fonction de gain ($\mathcal{R}$),
- un coefficient d'atténuation (γ).

3.2.1 Etats

L'ensemble fini d'états $\mathcal{S}$ de notre MDP représente les configurations possibles de l'évolution de notre araignée. La configuration $s \in \mathcal{S}$ dans laquelle l'araignée prend des décisions est définie par le poids de son corps (w) et par les caractéristiques de sa toile (c).

Nous détaillons à présent la signification de ces deux paramètres, puis nous précisons les caractéristiques des états buts et des états de mort.

Poids de l'araignée w

Le gain énergétique de l'araignée est proportionnel au poids de l'araignée. Ainsi, après sa dernière mue l'araignée a un poids initial (w_i) et doit atteindre un niveau énergétique final (w_f) favorable à la production d'une descendance[14].

Caractéristiques de la toile c

Le deuxième paramètre qui qualifie nos états est c, défini par le couple $\langle CTL, Age \rangle$. Noté $c_{CTL,Age}$, ce paramètre reflète l'efficacité de capture du piège avec :

[14] Nous en déduisons que l'état final s_f ne dépendra que du poids de l'araignée qui devra être égal à w_f quelles que soient les valeurs de c.

– CTL[15] : la longueur du fil de la toile tissée. Cette donnée fournit de l'information à la fois sur :

1. la taille de la toile, que l'on présume corrélée au succès de capture de proie [Eberhard, 1986][Sherman, 1994],
2. la durée de construction [Venner, 2002], qui peut être très liée au risque de prédation,
3. et l'investissement consacré à la construction d'une toile qui peut être estimé partiellement.

– *Age* : l'âge de la toile. Il nous renseigne sur la capacité de l'araignée à capturer de la nourriture. L'efficacité du piège décroît au cours du temps.

Etat de mort de l'araignée

L'état s_d marque la mort de l'araignée qui peut être atteinte de 2 manières :
– L'araignée peut être tuée par un prédateur ;
– L'araignée peut mourir de faim.
Le poids w_{d_s} dans lequel l'araignée meurt de faim sera fonction de l'état maximal qu'elle a déjà atteint par le passé : $w_{d_s} = f(w_{max\_atteint})$. En effet, nous considérons que l'araignée peut engager une partie des gains énergétiques dans la reproduction et qu'elle ne pourra plus utiliser cette part d'énergie pour assurer sa maintenance.

Indépendamment des caractéristiques de la toile, les états s_i, s_f et s_{d_s}, respectivement l'état initial, l'état final et l'état de mort par famine, sont identifiés par les poids, respectivement, w_i, w_f, w_{d_s}.

Exemple

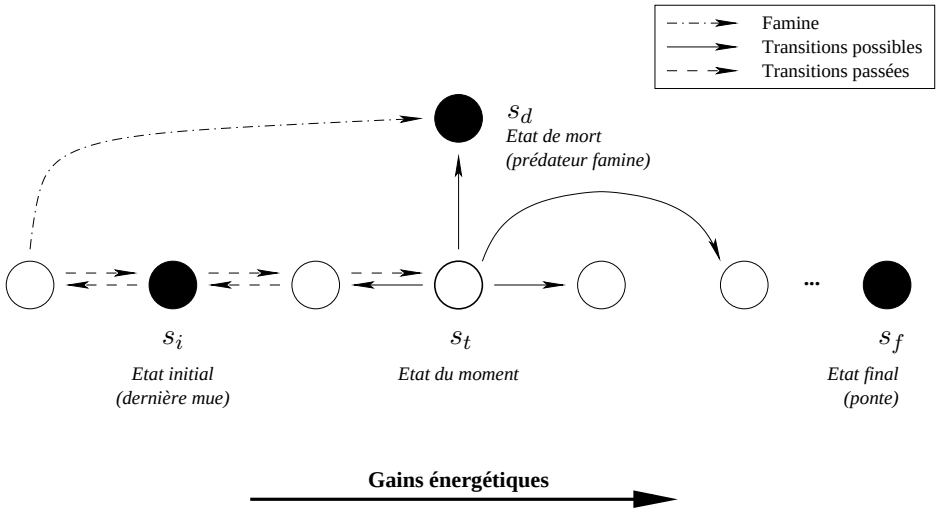


FIG. 3.2 – Evolution des gains énergétiques de l'araignée.

[15]CTL est l'acronyme anglais de "Capture Thread Length" que l'on peut traduire par "longueur du fil de la toile".

La figure 3.2 donne l'exemple d'une situation possible du système à l'état s_t. On distingue l'état initial s_i qui identifie la situation de l'araignée à la dernière mue. Si l'araignée perd trop d'énergie (*i.e.* de poids), elle peut arriver dans l'état de mort s_d pour cause de famine. A chaque instant elle peut également être victime d'un prédateur. Enfin, respectant l'hypothèse biologique formulée, l'araignée devrait réaliser l'action optimale dans chaque état, celle qui lui permettra de se rapprocher du but à atteindre, l'état s_f. Il nous faut maintenant définir l'ensemble des actions que l'araignée peut réaliser.

3.2.2 Actions

Dans les conditions naturelles, les araignées peuvent reconstruire quotidiennement leur toile. L'araignée réalisera donc une action par jour. L'action de construction ou de non construction d'une toile sera caractérisée par $A_{CTL=x,Age=a}$. Le choix d'action de l'araignée sera limité, il dépendra de l'état du moment $s_{w,c}$ et notamment des caractéristiques de la toile déjà existante $c_{x,a}$.

Chaque jour, l'araignée décide :

- soit de ne pas tisser, elle conservera alors la toile existante qui vieillit d'un jour et l'action correspondra à $A_{x,a+1}$.
- soit de tisser une nouvelle toile, elle choisira alors une toile de type $CTL = y$ parmi l'ensemble des types de toiles qu'elle peut réaliser. La toile aura 0 jour et n'aura pas encore permis à l'araignée de capturer de proies, l'action correspondra alors à $A_{y,0}$.

L'action $A_{CTL,a}$ relative à la construction ou non d'une toile aura des conséquences aléatoires. Il nous faut maintenant définir la fonction de transition stochastique entre les différentes configurations du système.

3.2.3 Fonction de transition probabiliste

Pour définir cette fonction de transition, nous devons déterminer pour toutes les actions $A_{CTL,Age}$ possibles, les dépenses et les gains énergétiques que nous exprimons sous la forme de pertes ou de gains de poids de l'araignée. Puis il nous faut prendre en considération les probabilités de capture et le risque de prédation dans le calcul de la matrice de transition. Ces deux probabilités dépendent des paramètres CTL et Age caractéristiques de la toile. On notera respectivement $\lambda_{CTL,Age}$ et $\beta_{CTL,Age}$ les probabilités de capture et de prédation auxquelles l'araignée est sujette en effectuant l'action $A_{CTL,Age}$.

Dépenses et gains énergétiques

Nous posons les hypothèses suivantes :

- Chaque jour, l'araignée doit couvrir des dépenses métaboliques lui permettant de rester en vie (α_m), indépendamment de son poids. Cependant, la construction d'une toile présente des dépenses énergétiques ($\alpha_{CTL,Age}$) qui dépendront à la fois de l'action réalisée $A_{CTL,Age}$ mais également du poids du corps de l'araignée w [Venner, 2002]. Ainsi, le coût énergétique associé à n'importe quelle action de construction de toile ($A_{CTL,0}$), noté $\alpha_{CTL,0}$, dépend de l'action choisie aussi bien que de l'état de l'araignée, c'est-à-dire principalement de son poids. Lorsqu'aucune action de tissage n'est décidée ($A_{CTL,Age\neq 0}$) aucune énergie supplémentaire n'est dépensée. Ce qui peut se traduire par l'équation suivante :

$$w_{t+1} = w_t - \alpha_m - \alpha_{CTL,Age} \text{ avec } \alpha_{CTL,Age} = f(CTL, Age, w_t) \tag{3.1}$$

– En contre-partie, la toile permet la capture de proies. Dans notre modèle, un seul type de proie de profitabilité constante N peut être capturé. La capture d'une proie entraîne des dégâts et donc une baisse d'efficacité du piège. D'autre part, l'ingestion d'une proie par l'araignée nécessite du temps. Aussi, le nombre de proies qu'une araignée peut capturer et ingérer en une journée est limité. Nous fixerons alors un nombre maximal n_{max} de proies capturables et ingérables par jour. L'équation (3.1) devient alors :

$$w_{t+1}^n = w_t - \alpha_m - \alpha_{CTL,Age} + nN \quad \text{avec } 0 \leq n \leq nmax \tag{3.2}$$

Nous venons de formuler le modèle économique des profits et coûts du comportement de l'araignée, il faut maintenant prendre en compte les probabilités de capture et de prédation.

Probabilité de transition

Dans chaque état s_t, l'araignée décide de réaliser l'action $A_{CTL,Age}$ avec des conséquences probabilistes. La figure 3.3 résume les cas possibles :

1. La prédation. L'araignée construit sa toile, elle peut être tuée par un prédateur. La probabilité d'atteindre l'état de mort s_{d_p} est $\beta_{CTL,Age}$. Toute araignée qui décide de garder sa toile n'est pas sujet au risque de prédation : $\beta_{CTL,Age\neq 0} = 0$.
2. La capture de proie. Le jour t, si l'araignée n'est pas tuée, elle pourra capturer et ingérer n proie(s) avec la probabilité $(1 - \beta_{CTL,Age})\lambda_{CTL,Age}^n$ avec $0 \leq n \leq n_{max}$. Le jour suivant, le poids de son corps sera :

$$w_{t+1}^n = w_t - \alpha_m - \alpha_{CTL,Age} + nN \tag{3.3}$$

$$\text{et} \quad C_{t+1} = C_{CTL,Age+n} \quad \text{avec } 0 \leq n \leq nmax$$

w peut alors prendre toutes les valeurs comprises entre les bornes $[w_{d_s}, w_f]$ incluses. L'équation (3.3) calcule des valeurs continues que nous avons discrétisées en utilisant un intervalle de poids donné Δ_w. Nous sommes ici confrontés au problème de la discrétisation d'un ensemble continu qui entraîne l'apparition d'artefacts [Houston et McNamara, 1999]. Pour résoudre ce problème, nous avons adapté la méthode proposée par Houston et McNamara. Elle consiste à déterminer la probabilité pour une valeur donnée w de tomber sur l'une des deux valeurs adjacentes appartenant à notre grille de poids discrétisée.

Ainsi,

(a) si le poids calculé w_{t+1}^n est plus bas que le poids de mort w_{d_s}, l'araignée meurt de faim : $\mathcal{T}(s_d, A_{CTL,Age}, s_d) = 1$. L'état de mort est absorbant.

(b) Si $w_{t+1}^n \geq w_f - \Delta_w$, l'araignée a atteint l'état but et $\mathcal{T}(s_f, A_{CTL,Age}, s_f) = 1$. Lorsque l'araignée est proche de l'état final, des effets de bord peuvent se produire. D'un point de vue théorique, la résolution du problème par des méthodes MDP fournit une politique optimale en respectant le critère d'optimalité : arriver à s_f (identifié par w_f), et ce rapidement, en maximisant sa fonction économique, tout en restant en vie. Cela engendre une variabilité de choix d'action à proximité de l'état but. Cependant, d'un point de vue biologique ce comportement instable n'a pas de sens. Une fois de plus, la discrétisation implique une simplification d'un modèle et introduit des biais de comportement. Identifier l'état but par une valeur de poids discrète w_f est une hypothèse trop simplificatrice. Les araignées ne pondent pas à l'instant précis où elles atteignent ce poids. Il est donc plus judicieux de considérer une zone floue autour

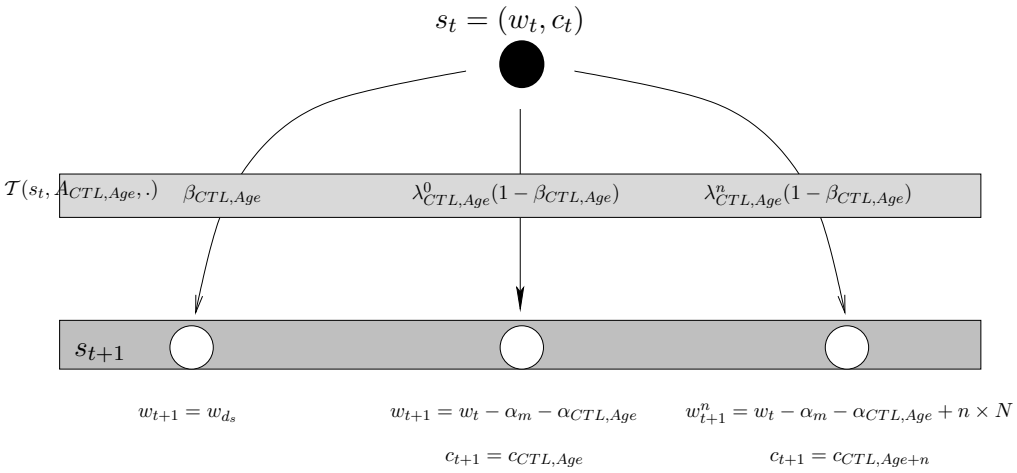


FIG. 3.3 – Fonction de transition probabiliste $\mathcal{T}(s_t, A_{CTL,Age}, .)$.

de cette valeur w_f. Ainsi, pour rendre le modèle plus proche de la réalité biologique, nous proposons de faire évoluer la probabilité d'atteindre l'état but s_f de manière croissante en fonction du poids calculé de l'araignée w^n_{t+1} dans l'intervalle défini par $[w_f - \Delta_w; w_{max}]$ où $w_{max} = w_f - w_i + n_{max} \times N$ [Venner, 2002].

(c) Enfin, dans tous les autres cas, lorsque w^n_{t+1} appartient à $]w_{d_s}; w_f - \Delta_w[$, l'araignée peut effectuer n'importe quelle action.

3.2.4 Fonction de gain

La fonction de gain permet d'attribuer une récompense $\mathcal{R} : s \to \mathbb{R}$, obtenue par l'araignée dans chaque état s_t. Elle permet de fixer le but à atteindre, c'est-à-dire de désigner l'état final dans lequel le système doit se trouver pour que le but soit satisfait. Dans notre situation, la fonction de gain ne dépendra que de l'état dans lequel se trouve l'araignée et notamment de sa masse corporelle (w).

$$\mathcal{R}(s_t) = \begin{cases} 1 & \text{si } w_t = w_f, \\ 0 & \text{si } w_t \neq w_f. \end{cases}$$

Afin de s'assurer que l'état final ait la valeur la plus forte possible, il est défini comme un état absorbant : une fois le but atteint il est impossible d'en sortir, quelle que soit l'action exécutée. C'est également le cas pour l'état de mort s_d.

La valeur d'un état absorbant s est la suivante :

$$V_\pi(s) = \mathcal{R}(s) + \gamma(1 \times V_\pi(s)) \Rightarrow V_\pi(s) = \frac{\mathcal{R}(s)}{1-\gamma}$$

Soit, pour l'état but $V_\pi(s_f) = \frac{1}{1-\gamma}$, tandis que, pour l'état de mort, la valeur sera nulle $(V_\pi(s_d) = 0)$.

3.2.5 Politique optimale

Comme nous l'avons vu dans le chapitre (2), la politique optimale π^* est calculée en associant à chaque état s l'action qui maximise la fonction de valeur calculée comme suit [Bellman, 1957] :

$$V_\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_\pi(s') \tag{3.4}$$

soit dans le cas de notre modèle :

$$V_\pi(s_t) = \mathcal{R}(s_t) + \gamma[\beta_{\pi(s_t)} \times V_\pi(s_d) + (1 - \beta_{\pi(s_t)}) \sum_{n=0}^{n_{max}} \lambda^n_{\pi(s_t)} \times V_\pi(s^n_{t+1})] \tag{3.5}$$

avec le coefficient d'atténuation $\gamma \in [0; 1[$. Cette fonction calcule la valeur reproductive associée à chaque état pour une politique π fixée. Nous avons utilisé l'algorithme *Value Iteration* (2.3) décrit dans le chapitre précédent pour calculer la politique optimale.

3.2.6 Conclusion

Tout au long de cette collaboration avec Samuel Venner, nous avons été confronté aux difficultés de modéliser un phénomène biologique réel, et de ce fait continu. Discrétiser l'ensemble des actions et des états de l'araignée constitue un compromis entre la volonté de reproduire fidèlement le comportement de l'araignée et la complexité grandissante du processus de résolution et de sa modélisation. Comme nous l'avons vu dans la section précédente, nous avons adapté le modèle théorique de départ qui, bien que correct, produisait des effets de bord dûs à la simplification du modèle expérimental et nuisibles à une interprétation biologique correcte. Nous nous sommes donc efforcés d'enrichir le modèle de simulation en y intégrant de nouvelles données biologiques, comme la gestion de l'intervalle dans l'équation de gain énergétique (3.3).

3.3 Réalisations et Résultats

Après avoir apprécié le succès des politiques optimales calculées sur des configurations types, nous avons choisi de tester l'influence de deux paramètres environnementaux (disponibilité de proie et risque de prédation), et de la contrainte des effets du poids de l'araignée. Pour cela, il a choisi d'étudier le comportement optimal que devrait suivre l'araignée dans deux types de situations : A. la situation où l'araignée doit choisir de construire, chaque jour, une grande ou petite toile ; et B. la situation où l'on s'intéresse à la fréquence de tissage de l'araignée qui ne peut construire que des toiles de taille moyenne[16]. Ainsi, dans chacune des situations, l'araignée doit choisir à chaque fois de réaliser une des deux actions possibles.

Pour chaque situation, une centaine de simulations a été réalisée. Une simulation du comportement de l'araignée se déroule de la façon suivante :

1. Au départ, l'araignée est dans l'état de fin de mue, il n'existe aucune toile.
2. Au cours de la simulation, l'araignée peut mourir de faim ou par prédation, ce qui termine la simulation.
3. La simulation se termine également lorsque l'araignée a atteint son poids de ponte.

[16] Nous ne rentrerons pas dans les détails des résultats biologiques obtenus, le lecteur intéressé se reportera à la thèse de Samuel Venner [Venner, 2002].

Paramètre	Description	Estimation
w_i	Poids initial de l'araignée	30(mg)
w_t	Poids de l'araignée à un instant t	–
w_f	Poids final de l'araignée	100(mg)
w_{d_s}	Poids de mort par famine	26(mg)
s_i	Etat initial	–
s_t	Etat de l'araignée à un instant t	–
s_f	Etat final (but)	–
s_d	Etat de mort	–
Δ_w	Intervalle minimal de discrétisation des poids	$0,5$
$A_{CTL,Age}$	CTL : longueur du fil de la toile	4 ; 8 ; 12 (m)
	Action de renouvellement de toile ($Age = 0$) ou	
	de conservation ($Age > 0$)	0 ; 1 (jour)
$\lambda^n_{CTL,Age}$	Probabilité de capturer n proies avec $A_{CTL,Age}$	variable
$\beta_{CTL,Age}$	Probabilité de prédation avec $A_{CTL,Age}$	variable
$\alpha_{CTL,Age}$	Dépenses énergétiques associées à $A_{CTL,Age}$.	$-0,149 \times CTL +$
	($\alpha_{CTL,Age} = 0$ si $Age > 0$)	$5.10^{-3} \times w_t \times CTL$
α_m	Dépenses énergétiques de base	$0,5$
N	Valeur énergétique d'une proie	5
n_{max}	Nombre maximal de proies ingérable par jour	1

TAB. 3.1 – Récapitulatif des paramètres utilisés dans le modèle.

Les valeurs comparées sont deux estimateurs du succès reproducteur : le taux de survie de l'araignée, et le délai entre la fin de la mue de départ et la ponte. Le taux de survie correspond au nombre d'araignées qui atteignent l'état de ponte. Expérimentalement, les mesures du taux de survie et des délais ont été réalisées sur une population de 1000 araignées, et ont ensuite été comparées [Venner, 2002].

3.3.1 Calcul de la politique optimale

Analyse de politique

La politique optimale fait correspondre à chaque état de l'araignée une action précise. Toutefois, pour certains états, les valeurs reproductives associées à certaines actions sont parfois très proches les unes des autres. D'un point de vue biologique, il est important d'avoir une estimation des actions qui ne seront pas trop coûteuses en terme de succès reproducteur. Ainsi, certaines déviations observées expérimentalement par rapport à la politique optimale ne remettront pas en cause l'hypothèse de travail formulée sur l'optimalité du comportement de l'araignée qui cherche à maximiser son succès reproducteur. McNamara et Houston ont été les premiers à prendre en compte l'importance des déviations [McNamara et Houston, 1986].

Afin de visualiser l'importance des déviations qui peuvent être observées expérimentalement, nous avons travaillé sur les valeurs associées à chaque couple état-action calculées sur la dernière itération de *Value Iteration*. La dernière itération est réalisée lorsque le test de fin de l'algorithme est vérifié, c'est-à-dire lorsque la différence des valeurs successives des deux dernières itérations est inférieure à ϵ. Cette dernière itération valide la convergence de l'algorithme vers la politique

optimale π^*.

Ainsi, nous avons utilisé un outil de la thermodynamique : l'équation de Boltzmann[17] [Sutton et Barto, 1998], qui fait correspondre à chaque couple (s, a) la valeur de Boltzmann :

$$BV(s,a) = \frac{e^{\frac{V_a(s)}{T}}}{\sum_{b \in A} e^{\frac{V_b(s)}{T}}} \tag{3.6}$$

Nous avons choisi de poser $T = 1$, afin de n'accroître, ni réduire les différences entre les valeurs. Ainsi, cette équation calcule le rapport entre l'exponentielle de la valeur reproductive de l'état s lorsque l'action a est choisie, et la somme des exponentielles des valeurs reproductives de l'état s pour toutes les actions b possibles. La probabilité calculée ainsi nous renseigne sur le coût d'éventuels biais par rapport à la politique optimale calculée. L'araignée ayant le choix entre deux actions, plus proche sera la probabilité d'une action sous-optimale de l'équilibre $0,5$, plus petit sera le coût et donc la perte de choisir cette action.

3.3.2 Influence du poids de l'araignée

La figure 3.4 représente les deux politiques optimales calculées par notre modèle théorique dans les situations A et B. La première courbe représente le cas où l'on prend en compte la masse de l'araignée dans la dépense énergétique :

$$\alpha_{CTL,Age} = -0,149 \times CTL + 5.10^{-3} \times w_t \times CTL$$

Pour la seconde politique optimale, on considère que le coût énergétique est indépendant du poids de l'araignée, Samuel Venner a montré que le modèle biologique le plus approprié était obtenu avec $\alpha_{CTL,Age} = 0,118 \times CTL$ [Venner, 2002].

Quelle que soit la situation A ou B, l'influence du poids de l'araignée dans la décision est de s'économiser lorsque qu'elle a atteint un certain poids. Dans la situation A, elle ne tissera alors que des petites toiles, tandis que dans la situation B, elle ne tissera qu'un jour sur deux. Lorsque le poids de l'araignée n'influence pas les décisions (cas 2), la politique optimale calculée dans les deux situations privilégie le tissage d'une grande toile, et de façon quotidienne. Samuel Venner a confronté ces résultats avec les expériences menées sur le terrain. Ces observations ont montré que les araignées se comportaient en diminuant leur activité en fin de cycle comme le prédit la politique optimale du premier cas ce qui conforte son hypothèse selon laquelle le poids de l'araignée interviendrait dans les dépenses énergétiques associées à ses activités de tissage. En d'autres termes, plus l'araignée est grosse, plus il lui faut dépenser de l'énergie pour construire sa toile.

3.3.3 Influence de la quantité de proie disponible

Comment devrait se comporter l'araignée lorsque la disponibilité en proies de son environnement est plus ou moins élevée ? La figure 3.5.A montre l'évolution de la politique optimale lorsque la richesse du milieu est importante, moyenne ou faible. Si l'on analyse quand a lieu le changement de décision (représenté ici par l'intersection avec la valeur $0,5$ de Boltzmann), on constate que plus l'environnement est faible, plus ce point se produit tôt. Effectivement, l'araignée choisit de construire une petite toile plus rapidement si la quantité de proies dans l'environnement est

[17] Nous précisons, qu'à l'instar des techniques utilisées en apprentissage par renforcement, ce calcul n'a qu'une valeur représentative, il ne s'agit en aucun cas d'y associer une politique stochastique optimale.

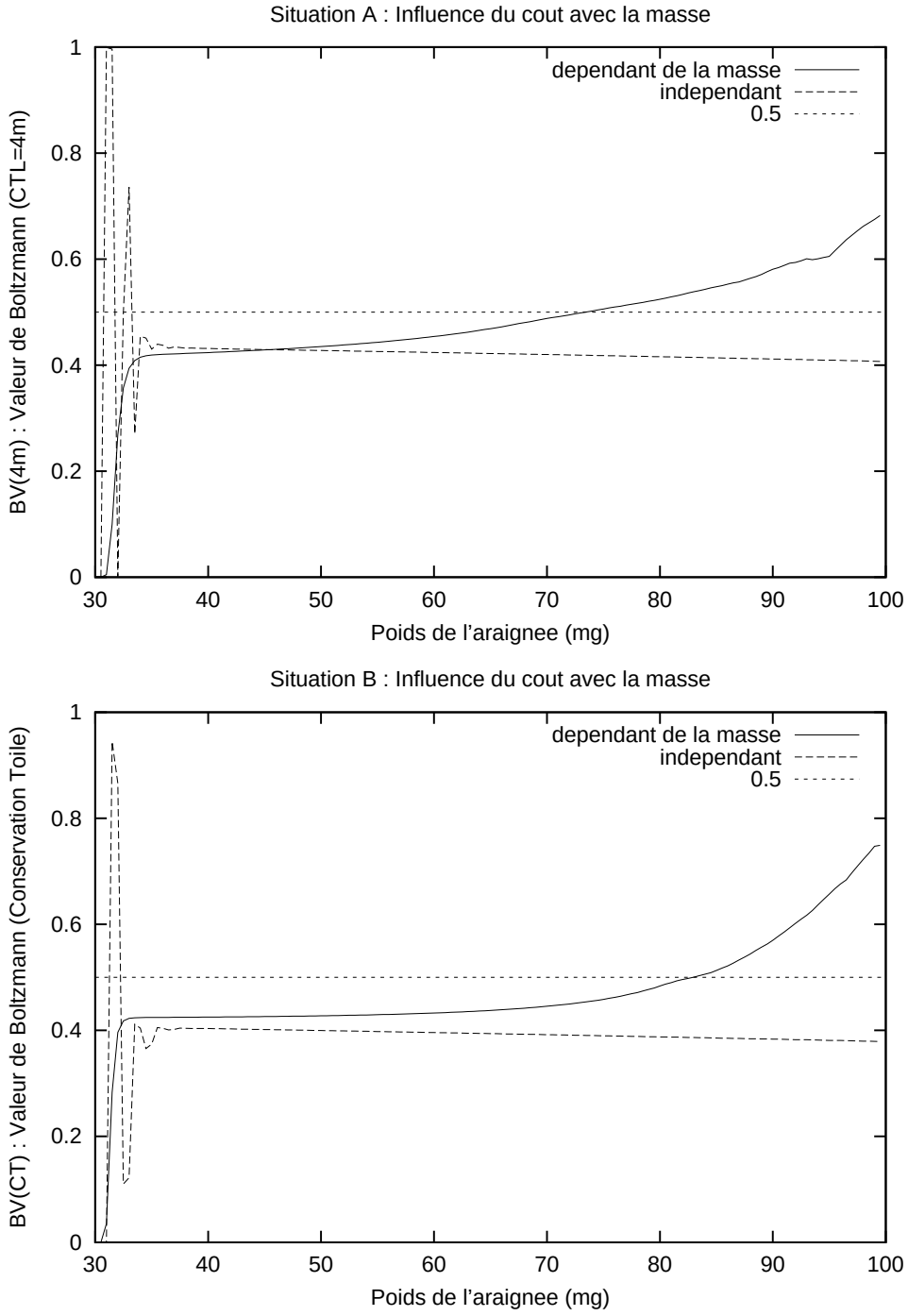


FIG. 3.4 – Situation A et B : politiques optimales calculées selon l'influence du poids de l'araignée.

faible, afin de minimiser ses dépenses énergétiques. C'est également ce que l'on observe sur la figure 3.5.B qui retrace la situation B : l'araignée choisit également de garder sa toile pendant 2 jours, alors qu'il lui était plus avantageux de la conserver avant ce poids de basculement.

3.3.4 Influence du risque de prédation

Le second paramètre dont nous étudions l'influence, est le risque de prédation. Tout comme c'était le cas précédemment, plus le risque de prédation est grand, plus le choix de tisser une petite toile est tardif (figure 3.6.A). Il en va de même en ce qui concerne la fréquence de tissage (figure 3.6.B). Enfin, notons que choisir l'une ou l'autre des actions peut ne pas avoir de grandes conséquences sur l'économie de l'araignée en début de construction des toiles. En revanche, ce n'est plus le cas en fin de processus : le coût de choisir une action alternative augmente significativement avec le poids de l'araignée dans les deux situations.

3.3.5 Simulation : comportement optimal *vs* comportement aléatoire

Après avoir analysé l'évolution des politiques optimales calculées par notre modèle en faisant varier les deux paramètres environnementaux que sont la quantité de proie disponible dans l'environnement et le risque de prédation, intéressons nous à présent aux simulations réalisées. Dans un premier temps nous comparons les performances d'araignées suivant la politique optimale à celles qui choisissent aléatoirement une action (tableaux 3.2 et 3.3).

Situation A et B : disponibilité en proies

Suivre la stratégie optimale réduit significativement les délais et variances entre la dernière mue et l'état de ponte. Cette réduction est d'autant plus importante lorsque la disponibilité de l'environnement en proies diminue : près de deux fois plus rapide pour le délai et 65% pour la variance dans le cas de la situation A.

Disponibilité en proies	Stratégie	Situation A		Situation B	
		Moyenne	Ecart type	Moyenne	Ecart type
Forte	Optimale	**32,88**	**7,31**	**35,38**	**7,94**
	Aléatoire	39,11	8,76	38,16	8,57
	Mixte	34,70	7,6	35,80	8,20
Moyenne	Optimale	**53,11**	**18,71**	**48,90**	**13,85**
	Aléatoire	68,34	26,97	54,71	17,88
	Mixte	55,5	20,4	50,00	15,10
Faible	Optimale	**101,97**	**52,63**	**83,46**	**32,31**
	Aléatoire	202,07	148,31	100,54	46,58
	Mixte	113,8	59,8	84,10	34,40

TAB. 3.2 – Disponibilité en proies : récapitulatif des simulations réalisées (en jours).

Situation A et B : risque de prédation

Comme on pouvait le prévoir, le taux de survie des araignées diminue lorsque le taux de prédation augmente et ce, quelle que soit la stratégie. On constate tout de même un meilleur taux de survie en faveur de la stratégie optimale. Cette différence de performance s'accroît avec l'augmentation du risque de prédation.

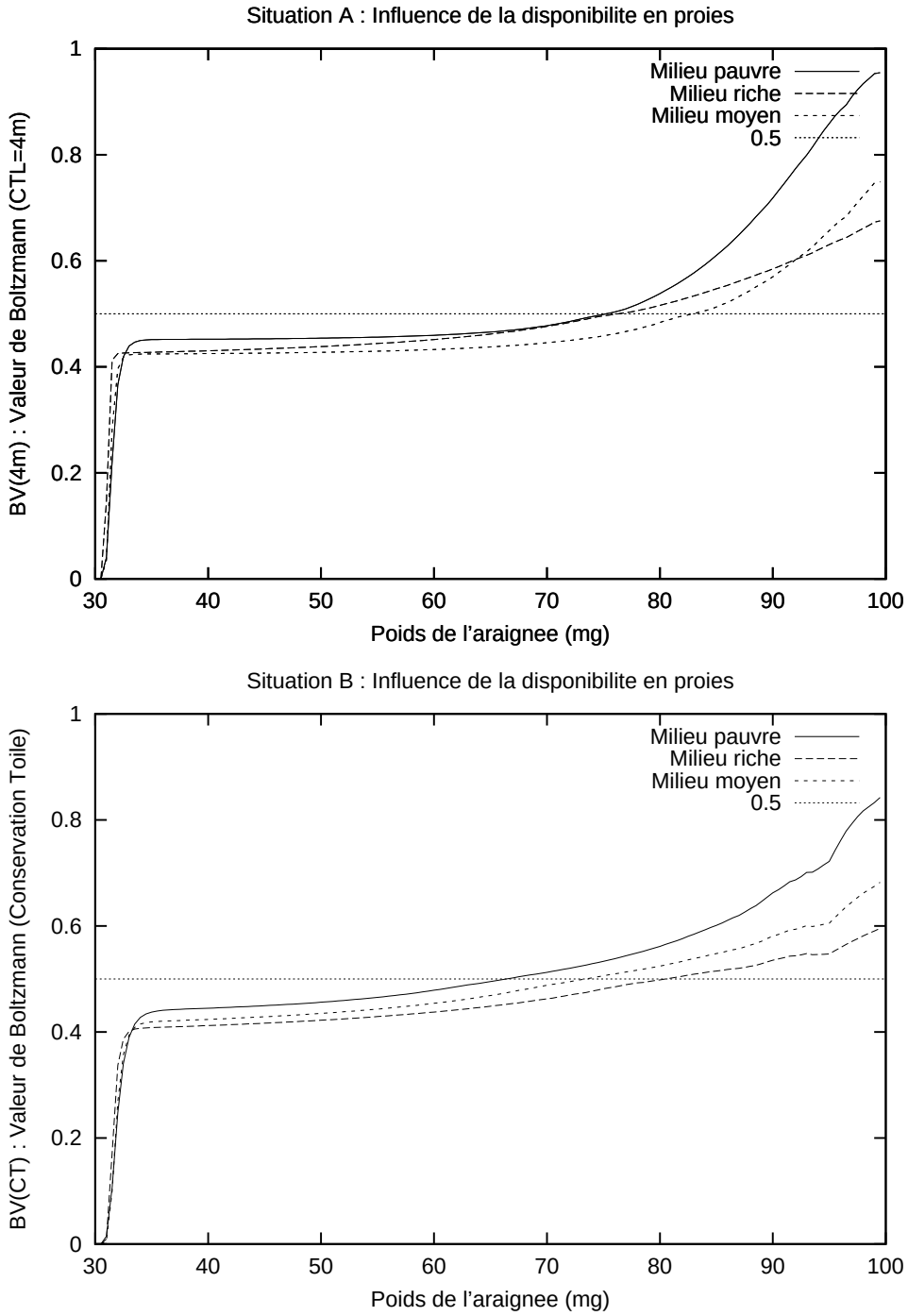


FIG. 3.5 – Situation A et B : politiques optimales calculées selon la disponibilité en proies de l'environnement.

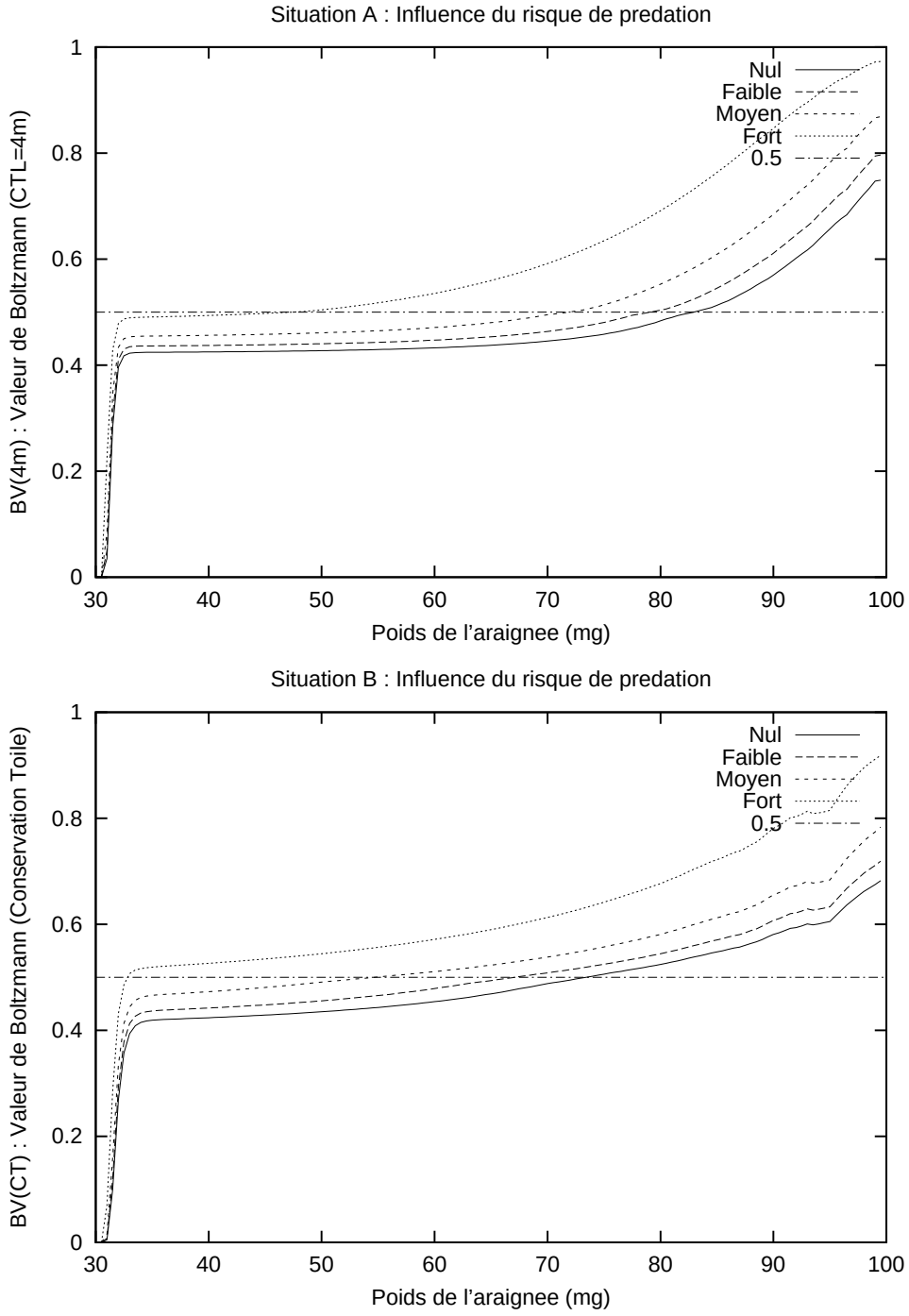


FIG. 3.6 – Situation A et B : politiques optimales calculées pour chaque risque de prédation.

Risque de prédation	Stratégie	Situation A		Situation B	
		Moyenne	Ecart type	Moyenne	Ecart type
Fort	Optimale	60,02	**19,14**	50,43	**13,47**
	Aléatoire	61,87	22,11	52,28	15,73
	Mixte	**57,9**	19,5	**49,98**	14,01
Moyen	Optimale	**55,65**	21,25	**49,88**	14,75
	Aléatoire	64,88	24,05	52,88	15,83
	Mixte	58,6	**20,4**	51,00	**14,40**
Faible	Optimale	**52,74**	**18,67**	**49,27**	**14,75**
	Aléatoire	65,86	23,04	54,05	15,52
	Mixte	56,1	21	51,41	15,40
Nul	Optimale	**53,11**	**18,71**	**48,90**	**13,85**
	Aléatoire	68,34	26,97	54,71	17,88
	Mixte	55,5	20,4	50,00	15,10

TAB. 3.3 – Risque de prédation : récapitulatif des simulations réalisées (en jour).

Situation A *vs* situation B

Les performances de la politique optimale comparativement à une politique aléatoire sont beaucoup plus significatives dans la situation A que dans la B. En effet, le problème posé en A exige de choisir entre deux actions de construction, tandis que l'araignée doit décider en B de construire une toile ou de la conserver un seul jour de plus. L'araignée n'a donc pas de choix une fois sur deux.

3.3.6 Simulation : comportement optimal vs comportement mixte

Le comportement mixte consiste à choisir aléatoirement, quelles que soient les situations (A ou B), entre les deux actions si les valeurs associées de Boltzmann sont comprises entre $0,45$ et $0,55$. Si ce n'est pas le cas, les araignées choisiront la stratégie optimale. Les résultats ont montré que, dans approximativement 43% des situations, l'araignée choisit aléatoirement. Evidement ce choix aléatoire ne se produit pas pour des états critiques (figures 3.5, 3.6). Le tableau 3.3 récapitule les résultats des simulations.

Situation A et B : disponibilité en proies

Le délai entre les deux stratégies comportementales n'est pas significatif dans la situation B. On constate, dans le meilleur des cas pour la situation A, que ce délai est légèrement plus court (10%).

Situation A et B : risque de prédation

Il n'y a pas de différence significative entre ces deux stratégies comportementales quel que soit le risque de prédation.

3.4 Conclusions

Dans ce chapitre, nous avons présenté une modélisation du comportement de constructions successives d'une araignée orbitèle à l'aide d'un processus décisionnel de Markov à horizon infini.

Le critère d'optimisation, choisi pour calculer la politique optimale, que devrait suivre l'individu, respecte l'hypothèse biologique selon laquelle l'araignée devrait optimiser sa fonction de gain énergétique afin de parvenir le plus rapidement possible dans son état de ponte (état final) tout en maximisant ses chances de survie, c'est-à-dire en tenant des caractéristiques environnementales comme le taux de prédation et la disponibilité en proies. En comparant les valeurs des actions sous-optimales, il a été possible d'estimer le surplus de dépense énergétique qu'engendreraient des actions sous-optimales.

Validation expérimentale

Confronté sur le terrain les simulations informatiques d'un comportement biologique n'est encore une fois pas chose aisée. Samuel Venner a confronté les résultats des simulations aux expériences réalisées *in vitro* afin de contrôler les paramètres, comme par exemple la disponibilité en proies de l'environnement. Les prédictions du modèle (variations de la fréquence de construction des toiles successives, de la quantité de soie utilisée pour la réalisation de chaque toile) ont pu être comparées aux comportements observés et ont révélé des différences analysées [Venner, 2002].

Ainsi, les écarts entre les prédictions du modèle et les observations pourraient être dûs à l'existence de contraintes limitantes non envisagées dans le modèle telles que la disponibilité en soie, une vitesse de développement des araignées (production, maturation des œufs, digestion de proie), évaluation imparfaite de la richesse du milieu ou l'âge de l'araignée. En revanche, l'adéquation entre les observations et les prédictions du modèle quant à la réduction des dépenses énergétiques en fin de période, a confirmé l'importance du poids de l'araignée dans l'influence de ces décisions.

Les résultats des analyses de cette première esquisse de modélisation du comportement de l'araignée peut à présent être enrichi selon les nouvelles hypothèses formulées. La difficulté reste de contrôler l'influence de nouveaux paramètres dans la prise de décision.

Raisonnement sous contraintes de ressources

Dans des travaux précédents, nous avons proposé une modélisation théorique des processus décisionnels de Markov dans un problème de gestion de ressources multi-agents étudié dans le contexte d'un problème d'ordonnancement de tâches [Chadès, 1998]. Hosam Hanna utilise lui aussi les modèles décisionnels de Markov afin de résoudre cette topologie de problèmes [Hanna et Mouaddib, 2002]. Notre modèle biologique est aussi un problème d'optimisation sous contraintes de ressources, et il s'agit d'une application réelle. L'araignée doit maximiser son énergie interne afin d'atteindre son état de ponte. Nous sommes donc dans une configuration de problème semblable à ceux rencontrés en robotique mobile (collective ou non). Les batteries fournissent de l'énergie au robot qu'il faut apprendre à gérer afin de maximiser un critère de satisfaction lié à la tâche que ce robot doit accomplir. L'état du modèle, qui reflète la situation de l'agent, ne prend plus en compte une position spatiale dans un environnement, il intègre une nouvelle donnée continue qu'il faut apprendre à discrétiser et dont dépend le succès de la tâche à accomplir.

Approximation d'un problème non Markovien

Enfin, ce travail illustre la possibilité d'approcher la modélisation d'un phénomène réel par nature non "Markovienne" à l'aide d'un MDP avec succès. Bien que l'analyse biologique d'une

politique comportementale diffère de celles que nous pratiquons en planification réactive, la méthode de résolution proposée conserve une certaine continuité comportementale. En cela, une modélisation qui fait appel à des méthodes de discrétisation demande de prendre les précautions qui s'imposent afin de ne pas introduire de biais dans l'élaboration du comportement. La modélisation informatique d'un phénomène réel requière également une justesse dans le choix des simplifications nécessaires à toute modélisation.

Le succès de cette approximation dans un cadre mono-agent et réel conforte l'intérêt de notre approche et nous encourage dans notre démarche de conception de systèmes multi-agents à l'aide d'un MDP.

Chapitre 4

Modèle pour la conception d'agents réactifs

Dans cette thèse, nous nous intéressons aux systèmes multi-agents qui mettent en jeu des agents autonomes, situés dans un environnement complexe, capable d'interagir, et qui doivent effectuer une tâche nécessitant la collaboration de tous. Autrement dit, un agent ne peut résoudre seul la tâche qu'on lui a confié. Afin de se rapprocher de la réalité des expérimentations de la robotique mobile collective, l'environnement est partiellement observable, épisodique, dynamique et discretisé [Russel et Norvig, 1995]. Les actions des agents sont non déterministes. Cette complexité renforce les difficultés des agents à agir dans ce monde.

Le chapitre 1 a montré qu'il existait plusieurs types d'agents, et d'architectures organisationnelles possibles. Dans la littérature, les démarches suivies dans la conception des agents sont souvent empiriques. Nous distinguons principalement deux méthodes de travail. Le procédé qui consiste à concevoir des agents avec certaines propriétés puis d'étudier les problèmes qu'ils sont capables ou non de résoudre : nous parlerons de méthode ascendante. Dans ce cas, les systèmes multi-agents sont souvent réactifs. Nous y opposons la méthode descendante, il s'agit à partir d'un problème donné de concevoir le système multi-agents capable de résoudre la tâche pour laquelle on l'a créé. Cette méthode est appliquée principalement dans les cas de systèmes cognitifs.

Problématique

L'objet de notre étude est l'élaboration d'un formalisme concis permettant la conception d'un système multi-agents. Notre système devra faire face à une forte incertitude, et par conséquent il sera confronté aux comportements probabilistes des agents. Comment les agents peuvent-ils en éviter les conséquences ? La solution que nous avons choisie est de prévoir cette incertitude en coordonnant nos agents à travers l'utilisation d'un processus de planification. Pour cela, nous nous inspirons à la fois des comportements biologiques des animaux cherchant à survivre dans un environnement réel très contraignant, mais aussi de l'utilisation de modèles de décision stochastiques comme outils de planification des agents.

Dans le chapitre 2, nous avons vu que les modèles de Markov permettent de gérer la stochasticité des décisions. Certains sont dédiés à une utilisation mono-agent avec un environnement accessible (MDP), ou partiellement observable (POMDP). D'autres sont dédiés à la modélisation de plusieurs agents dans un environnement observable ou non (MMDP, DEC-MDP, DEC-POMDP). Bien que le DEC-POMDP semble être le modèle le plus respectueux des caractéristiques de notre

problème, nous avons vu que sa résolution est NEXP-complet pour un nombre d'agents supérieur ou égal à 2. Son utilisation ne convient donc pas pour une approche réaliste de la coordination des agents dans notre système multi-agents. De plus, une approche adaptée à notre travail sur les systèmes multi-agents implique nécessairement une décentralisation.

Nous proposons une heuristique du DEC-POMDP pour concevoir chaque comportement d'agent. Elle repose sur la résolution d'un processus décisionnel de Markov adapté que nous avons appelé processus décisionnel de Markov subjectif. Ainsi, notre solution approchée à ce problème est fondée sur deux propriétés fondamentales de nos agents : **la subjectivité** et **l'empathie**.

Organisation du chapitre

Dans la première partie de ce chapitre, nous définissons le modèle de notre système multi-agents, et de nos agents. Nous montrons comment nous utilisons la propriété de subjectivité de nos agents, afin de concevoir un processus décisionnel de Markov subjectif mono-agent. Dans la troisième partie de ce chapitre, nous proposons d'utiliser la propriété d'empathie dans un cadre multi-agents où l'environnement est complètement observable pour résoudre un problème de type MMDP. Enfin, dans une dernière partie, nous intégrons ces deux propriétés afin de proposer notre méthode de conception d'agents réactifs dans le cas d'une population homogène.

4.1 Définition de notre système multi-agents

Nous proposons une méthode pour concevoir les agents d'un système multi-agents. Il nous faut préciser la définition des systèmes multi-agents que nous avons choisi d'étudier pour notre problème. Elle s'inspire des définitions de Ferber [Ferber, 1999] et de Jennings *et al.* [Jennings *et al.*, 1998] que nous avons présentées dans le chapitre 1 (sous-section (1.1.1), page 7).

4.1.1 Modèle du système multi-agents coopératifs proposé

Un système multi-agents est un ensemble d'agents en interaction défini par MAS$= \langle \mathcal{A}, \mathcal{E}, \mathcal{I}, \mathcal{G}, \mathcal{R} \rangle$ avec :

- $\mathcal{A}$: l'ensemble fini d'agents possédant des propriétés de perception, de décision (réactive ou non), et d'action ;
- $\mathcal{E}$: l'environnement dans lequel évoluent les agents. Il est constitué de tout ce qui n'est pas "l'agent" en action. On y inclut également les lois de l'environnement. A tout moment l'environnement peut être décrit par une configuration s avec $s \in \mathcal{S}$ l'ensemble des configurations possibles du système.
- $\mathcal{I}$: l'ensemble des interactions possibles entre les agents et l'environnement. A définir selon la topologie du problème que l'on cherche à résoudre.
- $\mathcal{G}$: l'ensemble des objectifs/buts que le système doit atteindre.
- $\mathcal{R} : \mathcal{S} \rightarrow \mathbb{R}$: la fonction de récompense globale du système qui identifie la satisfaction du système. Cette fonction est définie telle que l'on puisse la scinder (la transformer) en récompenses individuelles pour chaque agent et de façon à ce qu'elle conserve la propriété coopérative du système.

Les lois de l'environnement comprennent les incertitudes de réalisation des actions de tout ce qui compose l'environnement : agents et objets. La fonction de récompense formalise le problème que doit résoudre notre système. Le fait d'utiliser une récompense implique une possibilité de présenter le problème à résoudre sous la forme d'un problème d'optimisation.

4.1.2 Modèle d'agent proposé

Nous définissons ici de manière plus formelle les caractéristiques de nos agents. Un agent $\mathcal{A}_i$ est défini par :

- $O_i = \{o_i\}$: l'ensemble des perceptions de l'environnement réalisées par les capteurs de l'agent.
- $\mathcal{O}_i : \mathcal{S} \rightarrow O_i$: la fonction de perception de l'agent qui fait correspondre à une configuration s de l'environnement $\mathcal{E}$ une perception o_i.
- $A_i = \{a_i\}$: l'ensemble fini d'actions qu'un agent peut décider d'effectuer.
- $\pi_i : O_i \rightarrow A_i$: la fonction de prise de décision qui fait correspondre à une perception o_i de l'agent une action a_i.
- $T_i : O_i \times A_i \times O_i \rightarrow [0,1]$: la fonction de distribution de probabilité individuelle qui nous renseigne sur l'incertitude de réalisation des actions individuelles de l'agent. Cette incertitude peut être due par exemple à l'observabilité partielle de l'environnement et au fonctionnement de l'agent lui-même (dans le cas d'un robot, un glissement, un obstacle...). En d'autres termes, elle représente avant tout calcul le bruit de fonctionnement de l'agent, et comme nous le verrons par la suite elle pourra éventuellement être mise à jour.

Enfin, selon le mode de conception centralisé ou décentralisé l'agent pourra contenir explicitement :

- $\mathcal{R}_i : O_i \rightarrow \mathbb{R}$: la fonction de récompense individuelle dérivée de la fonction de récompense globale du système. Elle récompense l'agent lorsque ce dernier se trouve dans une situation perceptive recherchée.
- Des informations sur l'environnement contenues dans la variable : *Monde*. L'importance de cette donnée reste à définir selon les caractéristiques du problème à résoudre.

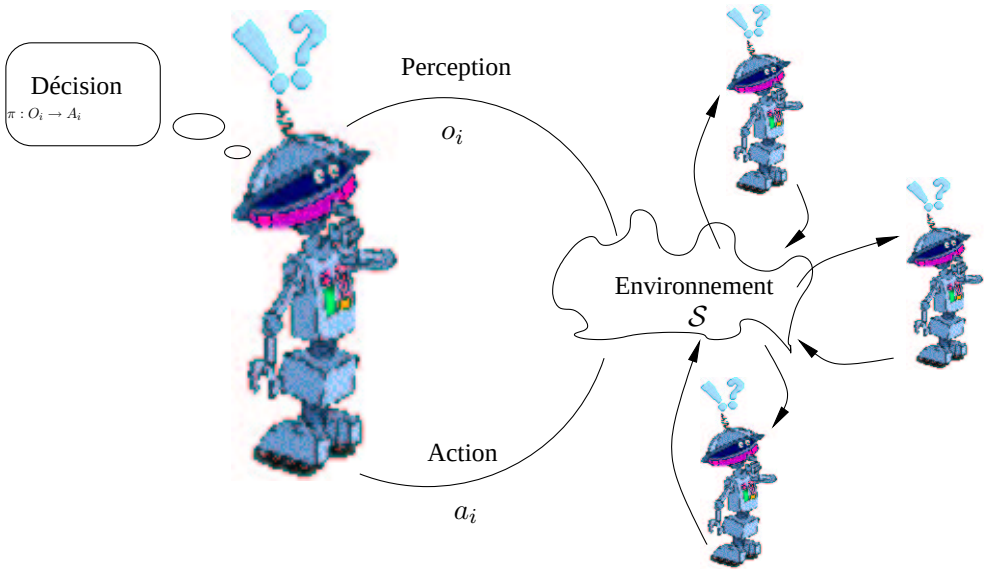


FIG. 4.1 – Modèle de notre système multi-agents.

Ainsi, la structure de nos agents est purement réactive. Elle associe à une perception o_i une décision a_i que l'agent réalise.

Étant données ces deux définitions, nous proposons de concevoir des agents cognitifs aux comportements, à l'exécution, réactifs.

4.1.3 Comment concevoir nos agents réactifs ?

En définissant précisément les systèmes multi-agents que nous cherchons à concevoir, ainsi que les problèmes qu'ils peuvent résoudre, nous venons de poser les premières fondations de notre étude. Résumons ce que nous avons étudié dans le chapitre 2 afin de préciser un peu plus notre problématique.

La conception de nos agents réactifs repose sur l'élaboration d'un plan ou d'une politique individuelle. Les propriétés de perception de nos agents de l'environnement nous place dans le cadre d'une observabilité limitée. Comme nous l'avons vu dans le chapitre 2, le POMDP est le formalisme adapté à ce genre d'agent, mais malheureusement peu réaliste en terme d'application dans un univers multi-agents. Contrairement au POMDP, le MDP est lui beaucoup moins coûteux en complexité et plus simple à utiliser dans cette configuration. Bien entendu, son utilisation dans un environnement non markovien ne garantie plus les propriétés de convergence vers une politique optimale. Notre solution est une heuristique qui tente de résoudre un DEC-POMDP. Elle s'appuie sur le formalisme d'un MDP adapté à l'observabilité partielle, et est fondée sur deux propriétés essentielles indépendantes de nos agents : **la subjectivité** et **l'empathie**. Notre méthode propose de simuler un DEC-POMDP à l'aide de plusieurs MDPs et sous certaines conditions d'utilisation, dont la plus importante est le respect de la coopération entre les agents.

Dans les sections suivantes, nous développons l'intérêt de ces deux caractéristiques indépendamment en mettant en valeur l'apport de chacune, puis nous les intégrons à notre modèle dans le cadre de notre problématique.

4.2 Subjectivité mono-agent et modèle décisionnel de Markov

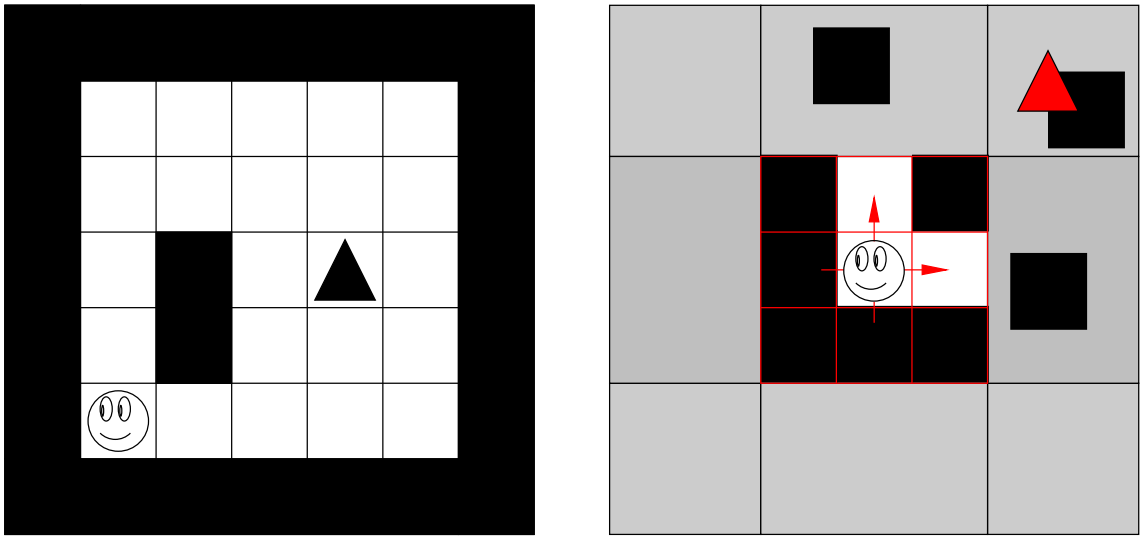
4.2.1 Subjectivité et localité

Dans la section (4.1), nous avons défini notre système multi-agents et notre modèle d'agent à partir de propriétés et de capacités des agents. La localité est l'une d'entre elles. La notion de subjectivité repose sur ce respect du paradigme multi-agents que nous voulons conserver dans le cadre des modèles décisionnels de Markov.

La subjectivité a pour effet de remplacer le repère du système global ou centralisé d'un agent utilisant un MDP, par un repère local ou ego-centré. L'agent est le centre de son monde, et n'en perçoit que des observations incomplètes.

La figure 4.2 illustre le principe de subjectivité que nous voulons utiliser dans nos MDPs subjectifs. En A, nous sommes confrontés à la vue centralisée usuellement utilisée dans les MDPs centralisés. L'agent situé en bas à gauche doit se rendre à son but que l'on a matérialisé par un triangle. B décrit une vue centrée sur l'agent. Dans ce cas de figure, l'agent ne perçoit que partiellement son environnement. Ses capteurs le renseignent avec une relative précision sur son environnement proche et ne lui donne que des indices dans les régions plus éloignées.

C'est ainsi que nous avons traduit la propriété de localité de nos agents ; adaptée au MDP, cette propriété se concrétise par l'utilisation d'un MDP subjectif. Une méthode de résolution



A− Vue centralisée B− Vue subjective de l'agent

FIG. 4.2 – Exemple de perception d'un environnement par un agent subjectif.

nous permettra de trouver la politique locale d'un agent subjectif dans ces conditions.

Afin de faciliter notre étude, nous développerons les caractéristiques du MDP subjectif dans un cadre mono-agent.

4.2.2 MDP subjectif

Comme nous l'avons vu dans le chapitre 2, un MDP est défini par $\langle S,A,T,R\rangle$:
- $S = \{s\}$: un ensemble d'états fini ;
- $A = \{a\}$: un ensemble d'actions fini ;
- $T : S \times A \times S \rightarrow [0,1]$: une fonction de transition d'états.
- $R : S \rightarrow \mathbb{R}$: une fonction de récompense.

Sous l'hypothèse que nous recherchons des plans réactifs pour nos agents, nous proposons d'utiliser des MDP subjectifs. Nous définissons un MDP subjectif de la façon suivante :
- S_i devient l'ensemble O_i des perceptions de l'agent ;
- A_i reste l'ensemble des actions ;
- $T_i : O_i \times A_i \times O_i \rightarrow [0,1]$ devient la distribution de probabilité approchée entre les perceptions.
- Enfin, R est défini sur l'ensemble des perceptions. $R_i : O_i \rightarrow \mathbb{R}$.

Notons que nous appelons MDP subjectif un POMDP dans lequel nous travaillons directement sur les observations avec une vue locale, sans utiliser d'états probables, ni d'historique. Les raisons pour lesquelles nous le qualifions de subjectif viennent des propriétés de notre système multi-agents, qui placent l'agent au centre de ses perceptions locales.

Perception et état agrégé

[Dean *et al.*, 1995] [Dean *et al.*, 1997] ont travaillé sur les meilleures façons de construire des états agrégés. Les perceptions des MDPs subjectifs sont des états s agrégés. Nous ne traitons pas ce problème dans cette thèse, mais nous tenons compte des résultats obtenus qui permettent un découpage de l'environnement intéressant [Scherrer, 2002].

Le principe que nous avons retenu s'inspire des capteurs utilisés en robotique mobile. Il s'agit d'attribuer à l'agent une perception plus ou moins fine jusqu'à une certaine distance. Au delà de cette distance, l'agent n'a pas ou peu de connaissance. Sur l'exemple de la figure 4.2 configuration B, la vue précise de l'agent s'étend à une distance 1. Ainsi, l'agent perçoit les murs sur sa gauche et en bas avec précision mais il ne perçoit que partiellement les murs haut et droit de l'environnement. Il dispose également d'une information précise sur une partie de l'obstacle qui cache son but. Le reste des informations dont il dispose situe les objets dans des régions. Ainsi, son but est situé dans la région Nord-Est.

Fonction de récompense individuelle

La fonction de récompense $R_i : O_i \rightarrow \mathbb{R}$ identifie le but local de l'agent. Reprenons notre exemple, dans ce problème, l'agent doit se rendre sur la case signalée par un triangle quelle que soit sa position initiale dans l'environnement. La figure 4.3 illustre un exemple des configurations buts possibles. Les informations contenues dans les cases adjacentes n'ont pas d'importance dans ce cas simple.

On pourra également faire apparaître dans la fonction de récompense, une sanction si l'agent entre en collision avec un obstacle ou un mur. Cette sanction prendra la forme d'une récompense négative.

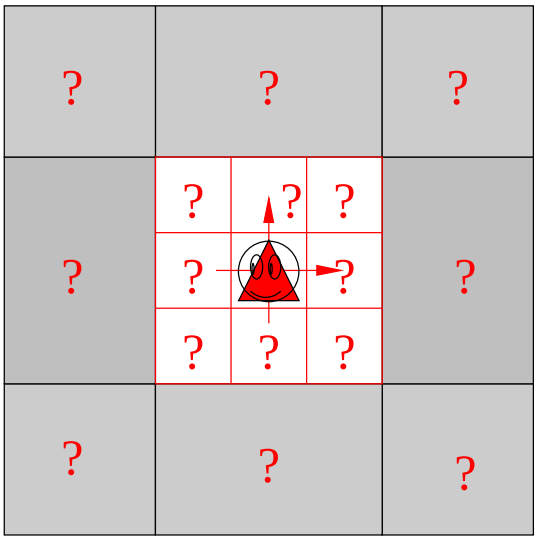


FIG. 4.3 – Exemple d'état but.

Comment calculer T ?

Dans le cas d'une observabilité complète, la connaissance de T ne pose aucun problème, il suffit de prendre en compte l'incertitude des actions occasionnées par l'agent lui-même. Par exemple pour un robot, il s'agit d'intégrer les éventuels dépassements de distance, pivotement insuffisant, etc... Dans le cas d'un MDP subjectif, comment calculer T ? Si l'on travaille sur les observations en ayant connaissance des états sous-jacents, c'est-à-dire en connaissant la fonction $O \rightarrow S$, on peut déterminer avec précision la fonction T. Il faut pour cela dénombrer le nombre des états s_i correspondant à une observation o_i et estimer la probabilité de transition que l'on peut associer à une action a_i. Dans le cas contraire, on fait l'hypothèse de ne pas connaître cette fonction, T sera alors calculée à partir du peu d'information disponible dans l'environnement.

Pour palier ce manque de connaissance, les recherches actuelles s'intéressent à l'apprentissage indirect de politiques, en apprenant la fonction T, ou directement à l'apprentissage de la politique stochastique qui approcherait la politique optimale. Dans [Buffet *et al.*, 2002], Buffet, Dutech et Charpillet s'intéressent à ce problème. Notre étude suppose que T peut être estimée ou apprise suffisamment correctement pour pouvoir planifier le comportement de l'agent. Il suffit pour cela d'avoir des connaissances sur la topologie du problème à résoudre. Par exemple, si le robot doit atteindre une balise en évitant des obstacles (murs ou roches), une information sur la densité de l'environnement permettra d'en ajuster les probabilités de T.

4.2.3 Effets de la subjectivité

Comme dans tout problème, les capacités des agents à percevoir et utiliser l'information disponible dans l'environnement sont essentielles. Dans notre cas, nous cherchons à concevoir des agents réactifs qui réagissent à leur perception en suivant un plan simple calculé au préalable, soit par les agents eux-mêmes, soit par un système centralisé, et qui fait correspondre une action à chaque perception. Nos agents ne possèdent pas de mémoire du passé, il nous est donc impossible d'utiliser des techniques faisant appel aux fenêtres d'historique comme nous l'avons vu dans le chapitre 2. Par conséquent, les politiques de nos agents seront très mauvaises dans des environnements de type labyrinthe où une connaissance totale de l'environnement est primordiale pour éviter des performances de l'agent catastrophiques.

Toutefois, dans des environnements de grande taille, où les obstacles sont rares et de faible envergure, calculer un MDP subjectif apporte les satisfactions liées à la résolution d'un MDP de petite taille avec une complexité en temps de résolution constante quelle que soit la taille de l'environnement. Dans sa thèse, McCallum a mis en valeurs les effets de l'observabilité partielle qui sont de deux ordres [McCallum, 1995] : aider le système à ne pas prendre en compte des détails inutiles, et de ce fait simplifier la résolution du problème, et cacher d'importants détails qui, s'ils sont ignorés, dégradent la qualité des solutions trouvées. C'est dans des conditions favorables que nous voulons utiliser nos MDPs subjectifs.

Un exemple

Si nous reprenons notre exemple de la figure 4.2, une politique déterminée en utilisant un MDP subjectif donnera des résultats différents si l'on considère une distance de vue exacte égale à 1 ou égale à 2. Comparons les politiques centralisées calculées à partir d'un MDP subjectif et d'un MDP complètement observable.

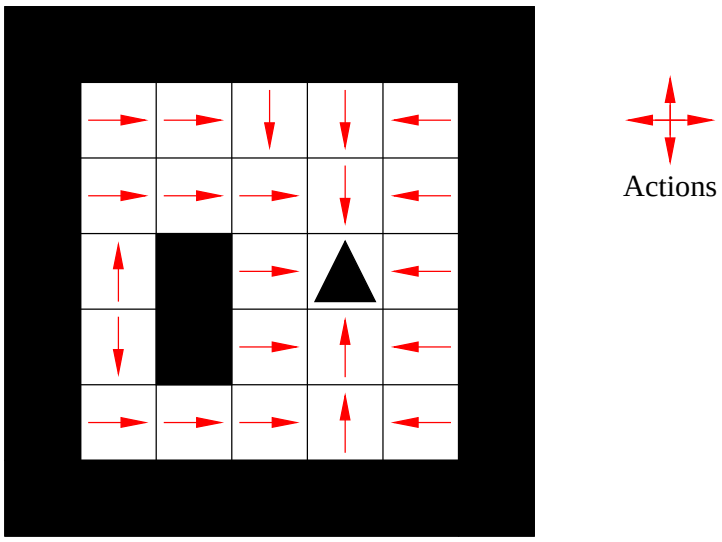


FIG. 4.4 – Politique centralisée que donnerait un MDP complètement observable.

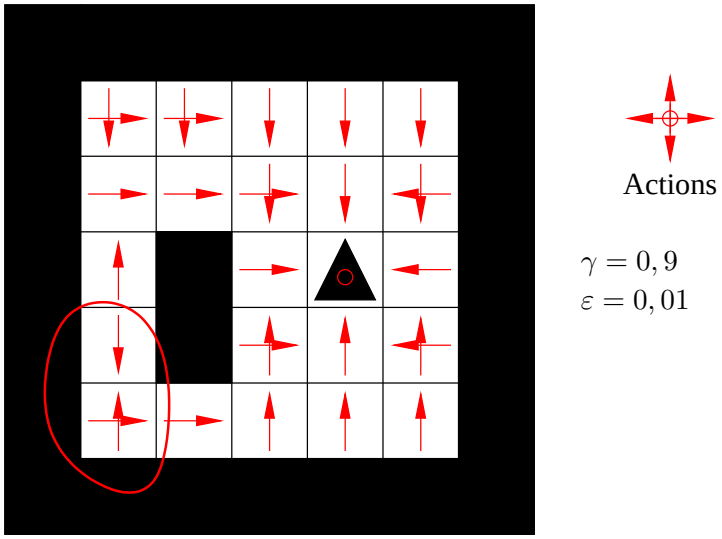


FIG. 4.5 – Politique reconstruite à partir de la projection de la politique calculée par le MDP subjectif.

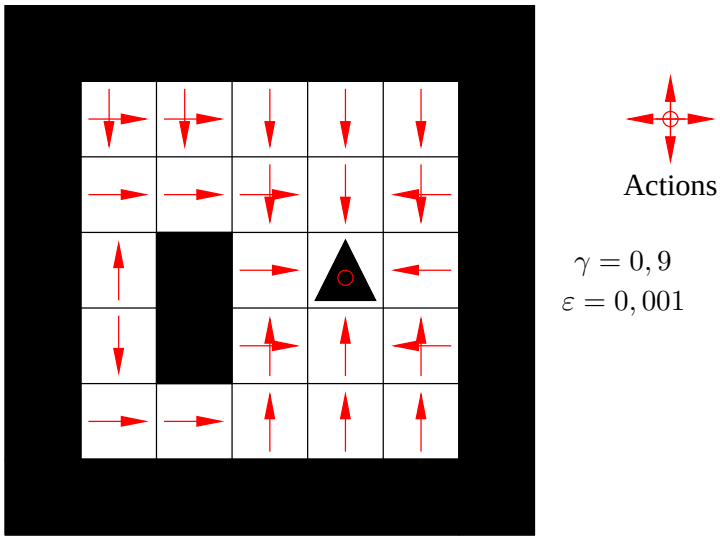


FIG. 4.6 – Politique reconstruite à partir de la projection de la politique calculée par le MDP subjectif.

La figure 4.4 représente ce que pourrait être la politique calculée par un MDP complètement observable, dans un environnement simple comprenant un mur et un état but toujours matérialisé par un triangle. Les figures 4.5 et 4.6 montrent les deux politiques calculées en utilisant l'algorithme du *Value Iteration* ((2.3) page 45) pour résoudre le MDP subjectif[18] dans ce même environnement. Les flèches nous indiquent l'action que l'agent décidera de suivre selon sa position, lorsque les valeurs des actions sont les mêmes, nous représentons toutes les actions possibles. La première figure nous livre une politique clairement sous-optimale, elle a été calculée avec l'algorithme du *Value Iteration* pour une valeur d'epsilon égale à $0,01$, l'algorithme n'a pas convergé en 45 itérations vers l'optimale. En effet, l'état mis en valeur sur la figure 4.5, nous montre que l'une des actions calculées en utilisant le MDP subjectif est de retourner dans l'état d'où il vient, l'autre action possible lui permet de se diriger vers la droite.

En modifiant la valeur d'epsilon ($\varepsilon = 0,001$), la seconde figure 4.6, nous dévoile une politique que l'on pourrait apprécier comme optimale en 67 itérations. L'action qui entraînait un blocage dans le cas précédent ne maximise plus la valeur de l'état. Notons que le calcul de T a été estimée en faisant l'hypothèse d'une distribution uniforme des obstacles sur l'environnement. Ce calcul met en avant l'importance du choix de la précision (ε), qui doit être suffisamment petite pour faire converger l'algorithme vers la politique optimale.

Indépendamment de la taille de l'environnement

Revenons sur un des avantages indéniables que crée l'utilisation d'un MDP subjectif. Tandis que la politique optimale calculée par un MDP complètement observable est dédiée à un environnement fixe, il est intéressant de noter que la politique calculée avec un MDP subjectif restera la même quelle que soit la taille et la topologie de l'environnement étudié. Ainsi, l'agent

[18] Cette politique optimale au sens du MDP subjectif, ne l'est pas forcément au sens du MDP complètement observable.

conçoit son plan de manière cognitive, alors que lors de son exécution, il comportera toute les caractéristiques d'un agent réactif : ses perceptions, décisions et actions sont locales, il n'a pas de représentation consciente du but qu'il doit atteindre.

4.2.4 Conclusion

Dans cette section, nous avons présenté le MDP subjectif que nous allons utiliser pour concevoir la politique d'un agent. Sous certaines conditions restrictives, nous avons montré que calculer un MDP en utilisant des états agrégés ou des observations incomplètes de l'environnement conduisent à l'élaboration d'une politique intéressante en terme de performance. Il s'agit à présent de considérer non plus l'agent réactif, mais bien le système multi-agents dans sa globalité. Pour cela, nous nous intéressons à la deuxième propriété fondamentale de nos agents : l'empathie.

4.3 Empathie des agents et modèle décisionnel de Markov

Dans le domaine de la psychologie, l'empathie se définit comme l'habileté à percevoir, à identifier et à comprendre les sentiments ou les émotions d'une autre personne tout en maintenant une distance affective par rapport à cette dernière.

Ainsi, l'empathie, qui est à la base de la compréhension psychologique d'autrui, reposerait sur une sorte de capacité à s'identifier avec un autre être, dont on réussirait ainsi à éprouver les sentiments, mais face à qui l'absence d'implication affective permet de préserver une certaine objectivité. En psychologie sociale, le terme empathie, dans un sens élargi, désigne aussi la capacité d'acquérir une certaine connaissance d'autrui, de se mettre à sa place. C'est cette propriété qui va nous permettre de coordonner nos agents.

Dans [Boutilier, 1999], Boutilier propose de formaliser le problème de coordination d'actions dans un système mettant en jeu plusieurs agents en utilisant un MDP centralisé, nous avons étudié ces travaux dans le chapitre 2 (paragraphe 2.7.2 page 60). C'est de ce modèle centralisé que nous partons afin de montrer comment nous allons utiliser la propriété d'empathie de nos agents et intégrer un raisonnement décentralisé. Dans cette section, nous laisserons de coté, temporairement, les contraintes de perception locale que nous imposons à nos agents, afin de nous concentrer sur le problème de la coordination d'actions. Ainsi, nous considérerons le monde comme étant parfaitement observable.

4.3.1 Notion d'empathie

Considérons de nouveau le modèle centralisé de MDP Multi-agents (MMDP)[Boutilier, 1999] pour n agents :

- S : ensemble fini des états du monde. Dans cet ensemble apparaît la position de chaque agent, et des objets du monde, c'est-à-dire toutes les informations observables. Les états buts appartiennent à cet ensemble.
- $A = A_1 \times A_2 \times ... \times A_n$ est l'ensemble fini des actions jointes des agents. Elle se définit à partir des ensembles d'actions finis A_i de chaque agent i.
- T et R sont les habituelles matrices de transitions et fonction de récompenses sachant S et A. Rappelons qu'elles sont globales.

La politique calculée π est une politique globale ou jointe. Elle fait correspondre à chaque état global du système une action jointe :

$$\pi : S \rightarrow A = A_1 \times A_2 ... \times A_n \tag{4.1}$$

Il est important de noter que le MMDP vérifie la propriété de Markov : la probabilité de transition d'un état s à un état s' en effectuant l'action jointe a ne dépend pas des états et des actions passées. Ainsi, tout comme le MDP complètement observable, il est possible de calculer la politique optimale π^*. Toutefois, l'espace des états et l'espace des actions sont largement plus importants puisqu'ils dépendent de $|S|$ et $|A|$.

Dans notre étude, calculer une politique jointe de manière centralisée va à l'encontre des principes et des problèmes que nous cherchons à résoudre : la localité et le calcul décentralisé des politiques demeurent les propriétés essentielles de nos agents, il n'est donc pas satisfaisant de faire comme si chaque agent avait une parfaite connaissance du système et du comportement de chacun. De plus, la complexité des méthodes de résolution d'un MMDP est à ce jour trop importante pour rendre réalisable le calcul d'une politique jointe pour un grand nombre d'agents dans un environnement de grande taille.

L'idée que nous voulons mettre en oeuvre afin que nos agents coordonnent leurs actions, repose sur leur capacité à prévoir le comportement de leurs compères et d'adapter ainsi leur propre comportement. Nous allons montrer que connaissant les politiques de certains agents, il est possible de concevoir des algorithmes capables de déterminer les politiques optimales individuelles des autres agents [Chadès *et al.*, 2002]. Formalisons notre approche.

4.3.2 Formalisme

Si nous reprenons la politique jointe (équation (4.1)), nous pouvons l'écrire d'une manière équivalente en faisant apparaître le comportement individuel des agents, c'est-à-dire les politiques individuelles de chaque agent ($\pi_i : S \rightarrow A_i$) :

$$\forall s \in S, \pi(s) = (\pi_1(s), ..., \pi_n(s))$$

Supposons qu'un certain nombre $n - m$ parmi les n agents aient une politique déjà définie. On note :

$$\forall s \in S, \pi(s) = (\pi_1(s), ..., \pi_m(s), ..., \pi_n(s)) \text{ avec } (\pi_i)_{m<i} \text{ connu}$$

La probabilité d'atteindre l'état s' à partir de l'état s dépend uniquement de l'incertitude de comportement des m premiers agents, et donc de leurs actions.

On peut alors définir un nouveau MMDP $M' =< S, A', T', R, \gamma >$ qui concerne seulement les m premiers agents, et auquel on a ajouté la connaissance du comportement des $n - m$ autres agents dans la fonction de transition T'. Autrement dit, nous intégrons la politique individuelle de chaque agent afin de calculer l'action optimale jointe des autres agents. C'est ce principe que nous exprimons à travers la propriété d'empathie de nos agents.

Formellement, le nouveau MMDP $M' =< S, A', T', R, \gamma >$ est défini par :
- S l'ensemble fini d'états,
- $A' = A_1 \times .. \times A_m$, l'ensemble fini d'actions jointes des m premiers agents,

– la nouvelle fonction de transition probabiliste calculée à partir de T et des $n-m$ politiques connues : $\forall(s,s') \in S^2$, $\forall(a_1,..,a_m) \in A'$,

$$T'(s,(a_1,..,a_m),s') = T(s,(a_1,..,a_m,\pi_{m+1}(s),..,\pi_n(s)),s')$$

– et enfin R, la fonction de récompense qui ne change pas.

Clairement, M' est plus facile à résoudre que M puisqu'il possède un ensemble d'actions plus petit. Montrons que si la politique optimale jointe $\pi^* = (\pi_1^*,..,\pi_m^*,..,\pi_n^*)$ est optimale pour M, $\pi' = (\pi_1^*,..,\pi_m^*)$ est une politique optimale jointe pour le nouveau MMDP M' ainsi constitué.

Théorème 1 (Stabilité de la solution optimale) :
Si tous les agents i avec $m < i \leq n$ suivent leur politique individuelle optimale au sens de M, alors la résolution de M' donnera les m' politiques individuelles optimales restantes.
Formellement, notons $(\pi_1^*,..,\pi_n^*)$ la politique optimale jointe de M. Si on définit M' à partir des politiques $(\pi_{m+1},..,\pi_n) = (\pi_{m+1}^*,..,\pi_n^*)$, alors $(\pi_1^*,..,\pi_m^*)$ est une politique optimale jointe de M'. □

Ce théorème va nous permettre de concevoir un premier algorithme prenant en compte la propriété d'empathie des agents. Notons Π et Π' respectivement l'ensemble des politiques jointes des n agents et m premiers agents avec $m < n$. Nous avons :

$$\Pi = (S \to A_1) \times .. \times (S \to A_n) \quad \text{et } \Pi' = (S \to A_1) \times .. \times (S \to A_m).$$

Pour démontrer le théorème de stabilité montrons tout d'abord que les fonctions de valeurs de Π et Π' sont égales.

Lemme 1 :
Pour tous les ensembles de politiques prédéfinies $(\pi_{m+1},..,\pi_n) \in \Pi \setminus \Pi'$ et toutes les politiques $(\pi_1,..,\pi_m) \in \Pi'$, les fonctions de valeurs de M et M', respectivement $V^{(\pi_1,..,\pi_n)}$ et $V'^{(\pi_1,..,\pi_m)}$, sont égales. □

Preuve du lemme 1—
$V'^{(\pi_1,..,\pi_m)}$ vérifie le même système linéaire que $V^{(\pi_1,..,\pi_n)}$ et a les mêmes propriétés, elle peut donc s'écrire sous la forme de l'équation de Bellman (2.11). En effet, pour tout $s \in S$:

$$V'^{(\pi_1,..,\pi_m)}(s) \quad = \quad R(s) + \gamma \sum_{s' \in S} T'(s,(\pi_1(s),..,\pi_m(s)),s') V^{(\pi_1,..,\pi_m)}(s')$$

Or, par définition :

$$T'(s,(a_1,..,a_m),s') = T(s,(a_1,..,a_m,\pi_{m+1}(s),..,\pi_n(s)),s')$$

$V'^{(\pi_1,..,\pi_m)}$ peut également s'écrire sous la forme :

$$V'^{(\pi_1,..,\pi_m)}(s) \quad = \quad R(s) + \gamma \sum_{s' \in S} T(s,(\pi_1(s),..,\pi_n(s)),s') V^{(\pi_1,..,\pi_m)}(s')$$

On a bien $V'^{(\pi_1,..,\pi_m)} = V^{(\pi_1,..,\pi_n)}$. ■

Démontrons à présent le théorème 1 en utilisant le lemme 1.

Preuve du théorème 1—
D'après le lemme 1, *et si les politiques* $(\pi^*_{m+1}, .., \pi^*_n) \in \Pi \setminus \Pi'$ *sont optimales pour* M, *alors* $\forall(\pi_1, .., \pi_m) \in \Pi'$. *Posons :*

$$V'^{(\pi_1,..,\pi_m)} = V^{(\pi_1,..,\pi_m,\pi^*_{m+1},..,\pi^*_n)}$$

En particulier, nous avons :

$$V^* = V^{(\pi^*_1,..,\pi^*_n)} = V'^{(\pi^*_1,..,\pi^*_m)}$$

Or si V^* *est optimale pour* M *:*

$$\begin{aligned}
&\Leftrightarrow \quad \forall(\pi_1, .., \pi_n) \in \Pi,\ V^* \geq V^{(\pi_1,..,\pi_n)} \\
&\Rightarrow \quad \forall(\pi_1, .., \pi_m) \in \Pi',\ V^* \geq V^{(\pi_1,..,\pi_m,\pi^*_{m+1},..,\pi^*_n)} \\
&\Leftrightarrow \quad \forall(\pi_1, .., \pi_m) \in \Pi',\ V'^{(\pi^*_1,..,\pi^*_m)} \geq V'^{(\pi_1,..,\pi_m)} \\
&\Leftrightarrow \quad V'^{(\pi^*_1,..,\pi^*_m)} \text{ est optimale pour } M'.
\end{aligned}$$

Donc, $(\pi^*_1, .., \pi^*_m)$ *est bien une politique optimale pour* M'. ∎

Nous venons de montrer que connaissant la politique optimale de certains agents il est possible de déterminer la politique optimale des autres agents en résolvant un MMDP de plus petite taille. Utilisons ce théorème pour concevoir un algorithme de co-évolution itératif alternatif et un algorithme de co-évolution itératif simultané.

4.3.3 Algorithme itératif de co-évolution alternatif

Le principe de co-évolution a été décrit en biologie comme un phénomène naturel qui illustre un cycle d'évolution : les espèces se transforment et par cette évolution elles transforment leur environnement, qui, lui-même, à son tour, modifie les espèces et ainsi de suite. Nous distinguerons l'évolution successive et l'évolution parallèle.

Dans notre cas, l'évolution parallèle de nos agents soulève les habituels problèmes de conflits, qui entraînent parfois des situations de blocage (deadlock). Notre objectif étant d'éviter les phases de conflit et de négociation, nous avons choisi de nous intéresser à la co-évolution dans le sens du cycle d'évolution successive. Nous l'évoquons sous deux approches, l'évolution alternative d'individus, et l'évolution simultanée d'un groupe d'individus.

En s'inspirant de ce principe, nous proposons deux algorithmes itératifs pour résoudre un MMDP, c'est-à-dire trouver les politiques individuelles des agents qui composent le MMDP.

Algorithme de co-évolution alternatif

Algorithme 4.1 Co-évolution alternative

Entrée: Un ensemble de politiques individuelles $(\pi_1^0, ..., \pi_n^0) = \Pi^0$

1: $t \leftarrow 0$;
2: **Répéter**
3: $m \leftarrow random(n)$;
4: $\mathcal{A}^t \leftarrow (\pi_1^t, ..., \pi_{m-1}^t, \pi_{m+1}^t, \ldots, \pi_n^t)$;
5: $b^t \leftarrow (\pi_m)$;
6: $b^{t+1} \leftarrow ResoudreMMDP(\mathcal{A}^t, b^t)$;
7: $t \leftarrow t + 1$;
8: **Jusqu'à** Convergence vers un point fixe

Sortie: $(\pi_1^t, ..., \pi_n^t) = \Pi^t$

On commence avec un ensemble de politiques individuelles arbitraires données en paramètre d'entrée $\Pi^0 = (\pi_1^0, ..., \pi_n^0)$. Puis, à chaque pas de temps t, on fixe les politiques d'un groupe d'agents $\mathcal{A}^t$ (ligne 4). L'agent b^t (ligne 5), trouve la politique optimale complémentaire en incorporant dans son modèle du monde les politiques fixes du premier groupe (ligne 6). La politique π_m ainsi calculée remplace la précédente. Enfin, l'algorithme s'arrête lorsqu'il converge vers un point fixe (ligne 8).

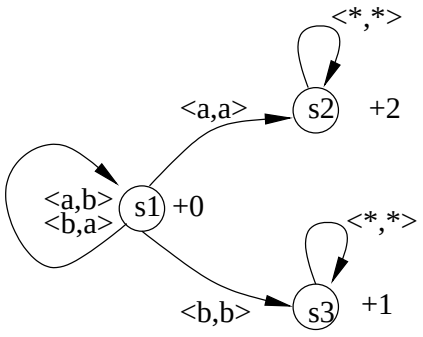


FIG. 4.7 – Exemple de convergence possible vers une politique sous-optimale.

La figure 4.7 montre qu'il peut exister des points fixes sous-optimaux. Si l'on applique notre algorithme à ce MMDP à deux agents, la politique trouvée peut être de deux types :

1. Si la politique fixée du premier agent $\pi_{\mathcal{A}}$ est égale à $\langle \mathbf{s_1} \rightarrow \mathbf{b}; s_2 \rightarrow a; s_3 \rightarrow a \rangle$, la politique optimale calculée par l'agent b π_b sera $\langle \mathbf{s_1} \rightarrow \mathbf{b}; s_2 \rightarrow *; s_3 \rightarrow * \rangle$. L'algorithme convergera vers une politique qui est un point fixe sous-optimal de la forme $\pi_{\mathcal{A} \cup b} = \langle \mathbf{s_1} \rightarrow < \mathbf{b}, \mathbf{b} >; s_2 \rightarrow < *, * >; s_3 \rightarrow < *, * > \rangle$.

2. Dans le cas favorable, si la politique fixée du premier agent $\pi_{\mathcal{A}}$ est égale à $\langle s_1 \rightarrow a; s_2 \rightarrow a; s_3 \rightarrow a\rangle$, la politique optimale calculée par l'agent b π_b sera $\langle \mathbf{s_1} \rightarrow \mathbf{a}; s_2 \rightarrow *; s_3 \rightarrow *\rangle$. L'algorithme convergera vers une politique qui est un point fixe optimal de la forme $\pi_{\mathcal{A}\cup b} = \langle \mathbf{s_1} \rightarrow < \mathbf{a}, \mathbf{a} >; s_2 \rightarrow < *, * >; s_3 \rightarrow < *, * >\rangle$.

Dans ces deux cas, le système a atteint un point d'équilibre dans lequel aucun agent ne peut améliorer la politique du système sans diminuer sa fonction de valeur individuelle V_i. Dans cet exemple, les points fixes obtenus par convergence sont des équilibres de Nash, dont certains sont optimaux. La solution optimale ne peut être trouvée à coup sûr, que par une résolution globale du système. Montrons que notre algorithme converge vers des équilibres de Nash.

Etude de convergence

Nous faisons évoluer deux sous-populations en alternance, les changements de politiques du premier groupe permettent de faire évoluer les politiques d'un autre groupe.

Le théorème (1) montre que la solution optimale jointe est un point fixe de cet algorithme, ce qui reste vrai pour un groupe composé d'un seul agent.

Théorème 2 (Convergence vers un point fixe) :
La fonction de valeur globale est monotone croissante et l'algorithme converge. □

Preuve du théorème 2—
Pour chaque t, posons $\mathcal{A}^t$ l'ensemble des politiques fixes, et $\mathcal{B}^t$ l'ensemble des politiques apprises.

D'après le lemme 1 :

$$V^{\mathcal{A}^{(t)}\cup\mathcal{B}^{(t)}} = V'^{\mathcal{B}^{(t)}}$$

et comme $\mathcal{B}^{t+1}$ est meilleur que $\mathcal{B}^t$ pour M', nous avons :

$$V^{\mathcal{A}^{(t+1)}\cup\mathcal{B}^{(t+1)}} = V'^{\mathcal{B}^{(t+1)}\cup\mathcal{A}^{(t)}} = V'^{\mathcal{B}^{(t+1)}} \geq V'^{\mathcal{B}^{(t)}} = V^{\mathcal{A}^{(t)}\cup\mathcal{B}^{(t)}}$$

Nous avons montré qu'à chaque itération $V^{\mathcal{A}^{(t+1)}\cup\mathcal{B}^{(t+1)}} \geq V^{\mathcal{A}^{(t)}\cup\mathcal{B}^{(t)}}$. La fonction de valeur globale $V^{\mathcal{A}^{(t+1)}\cup\mathcal{B}^{(t+1)}}$ est croissante. Comme elle est majorée par la fonction optimale théorique, elle converge. ■

Théorème 3 (Convergence vers un équilibre de Nash -1) :
L'algorithme itératif de co-évolution alternative converge vers un équilibre de Nash qui peut être la politique optimale. □

Nous avons déjà montré que l'algorithme convergeait vers un point fixe. Montrons que ce point fixe est un équilibre de Nash.

Preuve du théorème 3 (Raisonnement par l'absurde)—
Notons $\Pi = (\pi_1, \ldots, \pi_n)$ la politique jointe et $\Pi_{\pi_i \rightarrow \pi'_i}$ la politique jointe où l'on a remplacé la politique individuelle π_i de l'agent i par π'_i.
Par définition, la politique jointe Π est un équilibre de Nash si et seulement si

$$V_i^{\Pi}(s) \geq V_i^{\Pi_{\pi_i \rightarrow \pi'_i}}(s),\ \forall s \in S,\ \forall \pi'_i$$

D'après nos hypothèses, à chaque itération la fonction de valeur V_i de l'agent i qui améliore sa politique, est définie comme suit :

$$V_i^{\Pi}(s) = R(s) + \gamma \sum_{s'} T(s, \Pi(s), s') V_i^{\Pi}(s') = V^{\Pi}(s)$$

Posons Π la politique vers laquelle l'algorithme a convergé. Supposons qu'il existe une politique $\Pi' = \Pi_{\pi_i \to \pi_i'}$ telle que : $V_i^{\Pi'} > V_i^{\Pi}(s)$

Si une telle politique π_i' existe alors Π' est optimale pour la résolution du MMDP à $n - m$ agents et par conséquent Π ne l'est plus. Ce qui est contraire à l'hypothèse de convergence du théorème (2).

L'algorithme itératif de co-évolution alternative converge donc bien vers un équilibre de Nash. ■

Nous venons de montrer que notre algorithme de co-évolution alternative converge vers un équilibre de Nash. Nous proposons à présent d'étendre cet algorithme à une évolution simultanée de groupes d'agents.

4.3.4 Algorithme itératif de co-évolution simultané

Comme l'a montré la figure 4.7, l'algorithme itératif de co-évolution alternatif converge vers des optimums locaux qui sont des équilibres de Nash. Dans la première situation décrite, il n'est pas possible de sortir de cet optimum local. L'idée de ce nouvel algorithme est de faire évoluer successivement deux groupes d'agents afin de favoriser l'exploration de l'espace des politiques, et d'améliorer les valeurs des équilibres de Nash rencontrés.

Il s'agit de résoudre un MMDP avec m agents puis d'utiliser la solution trouvée afin de résoudre un MMDP avec n agents ($m < n$). Nous précisons que suivant les hypothèses faites précédemment nous conservons la fonction de récompense globale R pour chaque sous MMDP.

Algorithme 4.2 Co-évolution simultanée

Entrée: Un ensemble de politiques individuelles $(\pi_1^0, ..., \pi_n^0) = \Pi^0$

1: $t \leftarrow 0$;
2: **Répéter**
3: $m \leftarrow random(n)$;
4: $\mathcal{A}^t \leftarrow random(\{\pi_1^t, ..., \pi_n^t\}; m)$;
5: $\mathcal{B}^t \leftarrow \Pi^t \setminus \mathcal{A}^t$;
6: $\mathcal{B}^{t+1} \leftarrow ResoudreMMDP(\mathcal{A}^t, \mathcal{B}^t)$;
7: $t \leftarrow t + 1$;
8: **Jusqu'à** Convergence vers un point fixe

Sortie: $(\pi_1^t, ..., \pi_n^t) = \Pi^t$

On commence avec un ensemble de politiques individuelles arbitraires données en paramètre d'entrée $\Pi^0 = (\pi_1^0, ..., \pi_n^0)$. Puis, à chaque pas de temps t, on choisit deux groupes d'agents.

Les agents du premier groupe ont des politiques fixes $\mathcal{A}^t$. Les agents du deuxième groupe, $\mathcal{B}^t$, trouvent la politique optimale complémentaire en incorporant dans leur modèle du monde les politiques fixes du premier groupe. L'ensemble des nouvelles politiques $(\pi_1^t, ..., \pi_n^t)$ ainsi calculé remplace l'ensemble précédent. Enfin, l'algorithme s'arrête lorsqu'il converge vers un point fixe.

Etude de convergence

Le théorème (2) a montré que l'algorithme convergeait vers un point fixe. Montrons que si l'algorithme itératif de co-évolution simultanée converge, la politique obtenue est un équilibre de Nash.

Théorème 4 (Convergence vers un équilibre de Nash -2) :
L'algorithme itératif de co-évolution simultanée converge vers un équilibre de Nash qui peut être la politique optimale. □

Preuve du théorème 4—
Intuitivement si l'algorithme itératif de co-évolution simultanée converge vers une politique jointe, il n'existe alors plus d'amélioration de politiques possible quel que soit le groupe $\mathcal{B}$. Et en particulier, il n'existe pas de politique individuelle π' calculée par un groupe formé d'un agent qui permet d'obtenir une meilleure fonction de valeur $V_i^{\pi}(s)$. D'après le théorème 3, nous en déduisons que l'algorithme itératif de co-évolution simultanée converge vers un équilibre de Nash. ■

Remarquons que l'algorithme simultané permet de passer d'un point fixe à un autre en améliorant la fonction de valeur globale, tandis que l'algorithme alternatif ne saura pas sortir de ce point fixe. Par conséquent, l'algorithme simultané constitue un bon compromis entre une résolution globale capable de trouver la politique optimale mais cependant impossible à résoudre en terme de complexité. L'algorithme alternatif, quant à lui, diminue la complexité d'une itération mais il est plus souvent sujet aux équilibres de Nash de faibles valeurs.

4.3.5 Etude comparative

Dans ce paragraphe, nous proposons de comprendre le fonctionnement des deux algorithmes sur un exemple qui met en jeu trois agents et dont la configuration comprend plusieurs équilibres de Nash de qualités différentes. Sur cet exemple nous essayerons d'appréhender la vitesse de convergence des deux algorithmes ainsi que les qualités des solutions trouvées.

Exemple

La figure 4.8 illustre l'exemple simple que nous avons choisi de traiter. Ici, la politique optimale est clairement l'action jointe *bbb* en s_0 qui assure aux agents une récompense de $+3$ *ad vitam eternam*.

Clairement, les performances des algorithmes itératifs de co-évolution dépendront de la politique initiale π_0. Partons de $\pi_0 = aaa$. Quel que soit l'agent sélectionné, et ne pouvant améliorer qu'une politique individuelle à chaque fois, l'algorithme alternatif ne trouvera pas de meilleure solution. L'algorithme simultané trouvera quant à lui la politique optimale. En effet, la récompense attribuée à l'action *bab* est aussi élevée que celle attribuée à *aaa*. De ce nouveau choix, l'algorithme simultané peut converger vers la politique optimale, en calculant la politique optimale de l'agent numéro deux, ou d'un groupe comprenant l'agent numéro deux. Cette politique

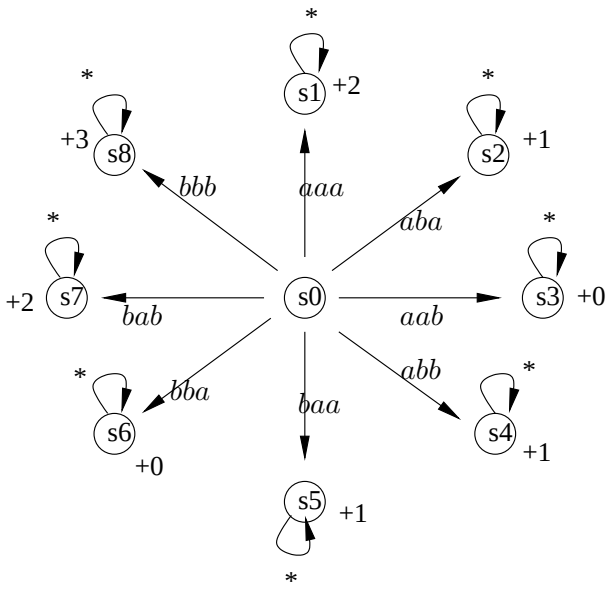


FIG. 4.8 – Exemple de MMDP à 3 agents et un état avec une fonction de transition probabiliste.

optimale n'était pas accessible en utilisant la méthode alternative en ayant $\pi_0 = aaa$ comme politique initiale, en revanche si la politique initiale comprenait une des actions jointes optimales, l'algorithme alternatif aurait convergé vers l'optimale.

Comportement des algorithmes

Comme nous l'a confirmé l'exemple précédent, l'algorithme simultané est plus apte à sortir des optimums locaux, tandis que l'algorithme alternatif n'a pas été conçu pour y faire face. L'algorithme simultané donnera donc au moins des politiques aussi bonnes que son prédécesseur. De plus, on peut supposer que dans les cas où la politique optimale est accessible aux deux algorithmes, la méthode de résolution simultanée convergera plus rapidement si l'on tient compte du nombre d'itérations nécessaires. Toutefois, si l'on s'interesse à la complexité d'une itération, l'algorithme alternatif a l'avantage de ne prendre en compte à chaque itération que la résolution d'une politique optimale individuelle, et de ce fait l'espace des actions est celui de l'agent évoluant.

Coordination et fonction de récompense

La question qui se pose est la suivante, étant données des situations de coordination nécessaires, est ce que l'algorithme co-évolutif fait se coordonner les agents ?

Tandis que Boutilier suggérait d'utiliser des états de coordination de manière explicite dans le cas où des politiques optimales individuelles étaient globalement sous-optimales [Boutilier, 1999], notre démarche tend à faire émerger la coordination de manière implicite en projetant le comportement des autres agents à travers un cycle d'apprentissage qui tend à faire croître la fonction de valeur optimale globale.

En d'autres termes, la coordination des actions est ici possible car la récompense du système est globale. La politique optimale recherchée tient compte de la fonction de récompense $R(s)$.

Dans un système coopératif, cette récompense peut s'exprimer en tenant compte de la situation des autres agents, si leur configuration individuelle est importante à la satisfaction du but globale.

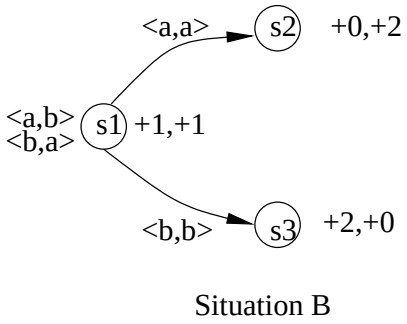


FIG. 4.9 – Exemple de MMDP non coopératif à fonction de récompenses individuelles.

La figure 4.9 illustre un MMDP où les agents ont des récompenses individuelles. Ce système est qualifié de non-coopératif, la satisfaction d'un agent passe par le mécontentement d'un autre. L'objet de notre étude concerne les systèmes coopératifs, dans lesquels la fonction de récompense individuelle de chacun tient compte de la situation d'une partie des autres agents.

Résoudre un MMDP

Bien que nous ayons réduit la complexité de résolution du MMDP pour un nombre n d'agents, le nombre d'états et d'actions pour un petit nombre d'agents restera trop important et tend à exploser. Pour ces raisons, bien que les résultats théoriques de l'algorithme simultané soient plus intéressant dans le cadre de notre problématique, résoudre un MMDP pour un groupe d'agents supérieur ou égale à 3 n'est en général pas réalisable. Or, la qualité de l'algorithme dépend du nombre de politiques que l'on est capable de faire évoluer simultanément. La question qui se pose alors est : quel nombre minimal d'agents compris dans un groupe faut-il faire évoluer afin d'obtenir une qualité maximale ? La réponse à cette question est bien sûr dépendante de la topologie du problème que l'on doit résoudre.

4.4 Subjectivité et empathie : algorithme de planification

Nous avons montré dans les deux sections précédentes, comment nous avons exprimé les propriétés de subjectivité et d'empathie de manière indépendante. Nous proposons à présent de nous intéresser à notre problème dans son ensemble, et nous présentons notre algorithme de conception de plans pour des agents à exécution réactive. La manière dont peuvent être conçus les plans de nos agents peut se dérouler de deux façons.

La première est une conception décentralisée. Elle fait appel à des agents cognitifs coopérant, capable d'interagir et de communiquer leur plan intermédiaire afin de parvenir à une éventuelle convergence.

La seconde est une conception centralisée. Connaissant les données d'un problème multi-agents à résoudre, une phase de conception des agents précède leur utilisation. Cette phase de conception est centralisée.

Dans les deux cas, nous ne pouvons adapter qu'un seul algorithme, l'algorithme alternatif. En effet, bien qu'il soit possible d'estimer de manière approchée la fonction de transition probabiliste pour un groupe d'agents, il n'existe pas, à notre connaissance, de méthode pour résoudre des MMDP subjectifs où plus généralement des Multi-agents POMDPs. Utiliser l'algorithme de co-évolution simultané décrit précédemment nécessite donc à ce jour une connaissance parfaite de l'environnement.

4.4.1 Système décentralisé : agents cognitifs à exécution réactive

Chaque agent cognitif possède un MDP-subjectif, chaque agent a son propre but qui peut être dépendant de ses compères. Nous proposons d'utiliser une forme modifiée de l'algorithme de co-évolution alternative.

Algorithme décentralisé de conception alternative

L'approche consiste à faire calculer alternativement par chaque agent sa propre politique en tenant compte de celles des autres agents.

Algorithme 4.3 Conception de politique décentralisée (itératif co-évolution alternatif)

Entrée: Une politique individuelle π_i^0

1: $t \leftarrow 0$;
2: **Répéter**
3: **Si** ACTIF **Alors**
4: Recevoir($\{\Pi^t\}$) ;
5: $\pi_i^t \leftarrow ResoudreMDPsubjectif(CalculT_i(\Pi^t))$;
6: **Sinon**
7: Envoyer(π_i^t) ;
8: **Fin Si**
9: $t \leftarrow t + 1$;
10: **Jusqu'à** Convergence vers un point fixe ou $t = tMax$

Sortie: π_i^t

Ainsi, conformément à l'algorithme itératif de co-évolution alternative, tous les agents commencent avec une politique arbitraire (algorithme (4.3)). A chaque pas de temps, un agent (choisi au hasard) demande aux autres agents leur politique. Il les intègre à son modèle du monde dans sa fonction de transition T_i et il calcule la politique optimale correspondant à sa connaissance. Les agents apprennent ainsi leur politique individuelle pendant un certain nombre d'itérations ($tMax$).

Algorithme d'estimation de T_i

Dans le cas d'un MDP subjectif, les états peuvent être des états agrégés, intégrer les politiques des autres dans le modèle de l'agent n'est pas aussi simple que pour un MDP. Pour chaque perception o, l'agent doit calculer $T_i(o,.,.)$ la distribution de probabilité de la fonction de transition. L'algorithme (4.4) décrit la manière dont l'agent i estime T_i.

Algorithme 4.4 Calcul de $T_i(o,.,.)$

Entrée: Un ensemble de politiques individuelles $(\pi_1, ..., \pi_n) = \Pi$

1: **Pour tout** $o_i \in \mathcal{O}_i$ **Faire**
2: $d_s^i \leftarrow OtoS(o_i, Monde)$ // Exploration des possibles
3: **Pour tout** Agent $j \neq i$ **Faire**
4: $d_o^j \leftarrow StoO(d_s^i, Monde)$ // Empathie phase 1
5: **Fin Pour**
6: **Pour tout** $a_i \in \mathcal{A}_i; j \neq i$ **Faire**
7: $d'^i_s \leftarrow Estimation(d_o^j, a_i, \{\pi_j\})$ // Empathie phase 2
8: **Fin Pour**
9: $d'^i_o \leftarrow StoO(d'^i_s, Monde)$
10: **Fin Pour**

Sortie: $d'^i_o = T_i(o,.,.)$

Détaillons l'algorithme précédent illustré par les figures 4.10, 4.11 et 4.12 :

- Ligne 2 : A partir de o_i, calculer les distributions d_s^i sur les états pour tous les agents j en utilisant son modèle du monde. L'agent i doit explorer tous les états possibles que comprend l'observation o_i. La figure 4.10 illustre cette exploration des états possibles. Deux agents sont situés dans un environnement sous forme de grille torique. Le triangle est un objet de cet environnement (A). (B) montre l'observation de l'agent i, et (C) présente les trois états possibles du monde.

- Ligne 4 : A partir des d_s^i, l'agent i obtient les d_o^j les distributions sur les perceptions de tous les agents en utilisant le modèle du monde. Dans l'exemple que nous avons choisi, il n'y a qu'un seul autre agent. Aux trois états du monde, trois observations possibles correspondent (figure 4.11).

- Ligne 7 : En utilisant les $\{\pi_j\}$ communiqués, l'agent i estime la prochaine distribution de probabilité sur les états des agents d'^i_s (figure 4.12).

- Ligne 9 : Enfin, l'agent i convertit d'^i_s en $d'^i_o = T_i(o,.,.)$.

Autrement dit, l'agent i doit estimer à partir de ses perceptions o_i où les autres agents se situent, ce qu'ils perçoivent, ce qu'ils font, et quelles conséquences ont ces actions sur le monde et ainsi sur sa perception o. Par dénombrement des possibles, l'agent i estime $T_i(.,.,.)$.

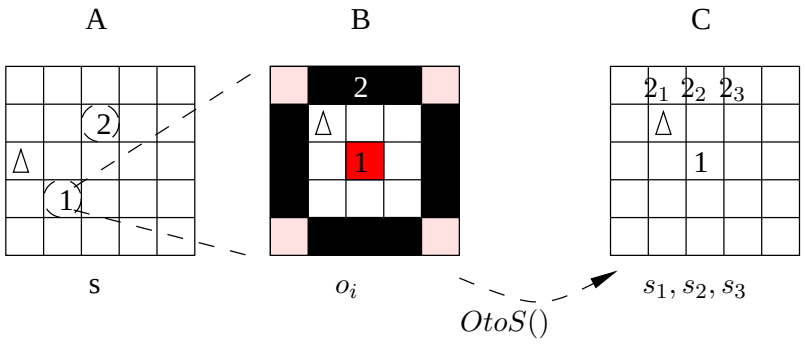


FIG. 4.10 – Exploration des possibles. (A) l'état du monde. (B) la perception o_i de l'agent 1. (C) les trois états possibles du monde pour l'observation o_i

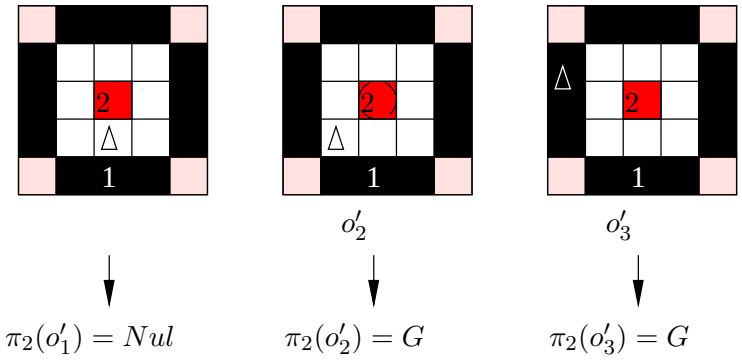


FIG. 4.11 – Empathie phase 1 et 2 – $StoO(d_s^2)$: Une observation possible pour chacun des s_1, s_2, s_3. A chacune des observations une action de l'agent 2 d'après π_2.

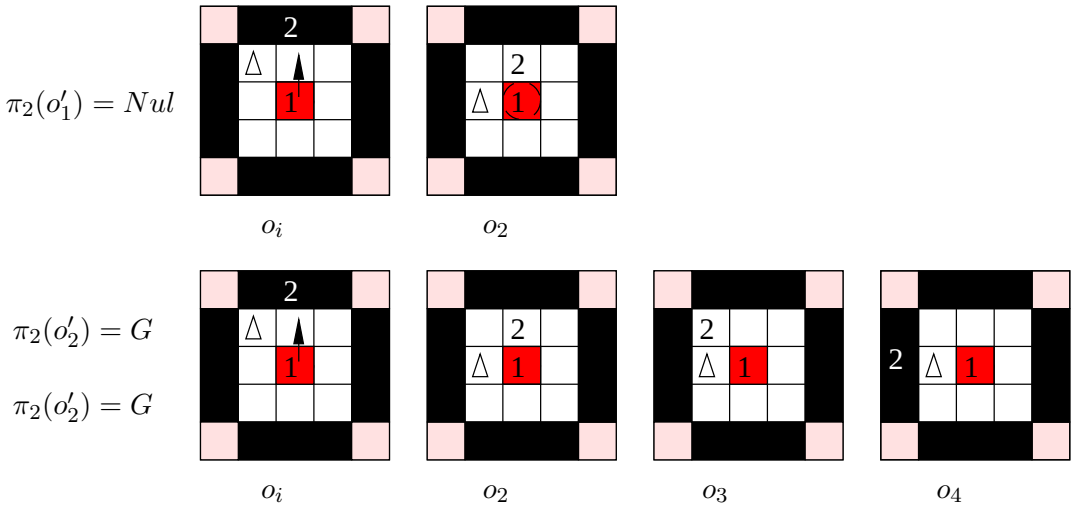


FIG. 4.12 – Empathie phase 2 – Estimation de $T_i(o_i, Haut, .)$: $T_i(o_i, Haut, o_2) = 3/7$; $T_i(o_i, Haut, o_3) = T_i(o_i, Haut, o_4) = 2/7$;

4.4.2 Système centralisé : conception d'agents réactifs

Dans le cas d'un système centralisé la démarche est quasi-similaire, il s'agit de calculer le plan des agents avant de les concevoir. Pour cela, un algorithme centralisé prend en charge l'évolution des politiques individuelles de chaque agent. Le bénéfice de cette centralisation s'exprime à travers la définition globale de la fonction de récompense des MDPs subjectifs. Il devient alors possible, en théorie, d'utiliser l'algorithme de co-évolution simultané afin de faire évoluer non plus un seul agent, mais un groupe d'agents.

Algorithme centralisé de conception alternative

Tous les agents commencent avec une politique arbitraire. A chaque pas de temps, la politique d'un agent est améliorée. La fonction de transition T_i est calculée de manière similaire. Puis la résolution du MDP subjectif nous donne la politique approchée correspondant aux informations disponibles. Les agents sont ensuite conçus. Notons qu'il n'y a plus convergence vers un équilibre de Nash, puisque nous travaillons sur l'espace des observations, et que la résolution du MDP subjectif ne converge pas vers une politique optimale au sens des états.

4.5 Cas particulier d'une population homogène

Considérons le cas d'une population homogène d'agents dans le cadre d'un système multi-agents réactif communiquant ou non communiquant. Les agents ont les mêmes informations sur l'environnement : le même ensemble d'actions A_i, le même ensemble de perceptions O_i et ils peuvent communiquer ou non pour construire leur plan π_i. Ils doivent résoudre une tâche globale exprimée de manière commune par des buts locaux R_i.

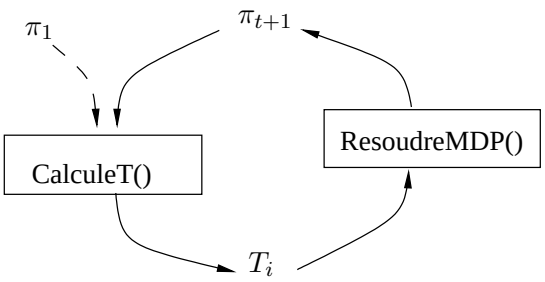


FIG. 4.13 – Algorithme de conception population homogène.

L'idée que nous soutenons est que les agents devraient calculer le même plan puisqu'ils sont identiques en tous points. Ainsi la construction d'un plan pour un agent est la même pour tous. Il suffit donc de calculer le plan d'un seul agent.

Algorithme décentralisé de conception d'agents réactifs non communicants

L'algorithme que nous proposons pour ce problème particulier est similaire au cas d'une population hétérogène, excepté qu'il tient compte des avantages et des inconvénients de l'isolement, en ayant connaissance de sa situation de "clone". L'algorithme ne considère alors qu'une seule politique à la fois pour tous les agents. La figure 4.13 illustre le principe de notre algorithme.

Algorithme 4.5 Conception décentralisée de π_i pour une population homogène non communicante

Entrée: Une politique individuelle π_i

1: $t \leftarrow 0$;
2: **Répéter**
3: $\quad \pi_{t+1} \leftarrow ResoudreMDPsubjectif(CalculeT(\pi_i^t))$
4: $\quad t \leftarrow t+1$;
5: **Jusqu'à** Un cycle de politiques

Sortie: π_i

L'agent isolé ou seul va considérer que tous ces jumeaux suivent π_i. Ainsi, il va calculer sa nouvelle politique adaptée au comportement de ses paires. Ce processus est matérialisé par l'équation :

$$\pi_{t+1} = ResoudreMDPsubjectif(CalculeT(\pi_i^t)) \tag{4.2}$$

$CalculeT(\pi_i^t)$ est la fonction de distribution de probabilité T_i calculée en tenant compte du fait que tous les agents sauf un suivent la politique π_i^t. Ce calcul est fait de manière similaire au cas d'une population hétérogène, nous ne le détaillerons pas.

Théorème 5 (Convergence vers un cycle fini de politiques) :
Si l'espace d'états est fini et si la fonction

$$ResoudreMDPsubjectif(CalculeT(.))$$

est déterministe, le processus itératif de l'équation (4.2) converge vers un cycle fini de politiques. □

Preuve du théorème 5 (Démonstration par récurrence)—
Quelle que soit la politique de départ π_i^0, considérons la trajectoire $\mathcal{T} = (\pi_i^0, \pi_i^1, ..)$ calculée par l'équation (4.2).

Le nombre des politiques possibles étant fini et la trajectoire $\mathcal{T}$ étant potentiellement infinie, il y a au moins deux politiques identiques : il existe $m < n$ tel que $\pi_i^m = \pi_i^n$.

Pour tout $t \geq 0$, $\pi_i^{m+t} = \pi_i^{n+t}$, la résolution de

$$ResoudreMDPsubjectif(CalculeT(.))$$

étant déterministe,

En particulier, pour tout $k > 0$

$$\pi_i^{m+k.(n-m)} = \pi_i^{n+k.(n-m)} = \pi_i^{m+(k-1).(n-m)}$$

Par récurrence sur k nous avons : $\forall k \geq 0$

$$\pi_i^m = \pi_i^{m+k.(n-m)}$$

Précédemment nous avions : $\forall k \geq 0$, $t \geq 0$

$$\pi_i^{m+t} = \pi_i^{m+k.(n-m)+t}$$

C'est-à-dire, qu'à partir du rang m, $\mathcal{T}$ fait des cycles de taille $(n-m)$. ■

Remarquons que l'équation (4.2) signifie que π_i^{t+1} est la meilleure politique pour un agent si il sait que tous les autres agents suivent la politique π_i^t.

Cette remarque suggère que, quel que soit t, les résultats doivent être meilleurs avec un leader : un agent suit la politique π_i^{t+1} tandis que les autres suivent la politique π_i^t. Ce résultat sera montré dans la partie expérimentation.

Algorithme de conception d'agents réactifs communicants

Dans le paragraphe précédent, nous avons étudié le cas d'une conception de plan pour un système multi-agents non communiquant. Dans ce cas, nous ne bénéficions plus de la preuve de convergence de l'algorithme de co-évolution alternative vers un équilibre de Nash pour des états. Étudions brièvement le cas où nous utilisons de nouveau notre algorithme de co-évolution alternative. Nous adaptons ce dernier en permettant une éventuelle communication de politique entre les agents. On peut également percevoir cette approche comme une conception d'un système multi-agents réactif de manière centralisé (dans ce cas les agents n'ont pas besoin de communiquer leur plan).

Algorithme 4.6 Conception centralisée des $\pi_1, \ldots, \pi_n$ pour une population homogène

Entrée: n politiques individuelles $\pi_1, \ldots, \pi_n$

1: $t \leftarrow 1$;

2: **Répéter**

3: $\pi_t \leftarrow ResoudreMDPsubjectif(CalculeT(\pi_1, \ldots, \pi_n))$

4: $t \leftarrow (t+1)\%n \ +1$;

5: **Jusqu'à** Un cycle de politiques

Sortie: $\pi_1 \ldots \pi_n$

Cet algorithme calcule les plans individuels de chaque agent : on parlera de phase de conception des agents. A chaque itération, une nouvelle politique π_t est améliorée connaissant les n politiques fixes des autres agents. L'algorithme se termine lorsque l'amélioration successive des politiques individuelles converge vers un cycle fini de politiques qui comme nous allons le voir, peut être sous certaines hypothèses, un équilibre de Nash sur les observations.

Nous avons vu qu'il n'existait pas toujours de politiques déterministes optimales pour un agent lorsque nous travaillons sur l'espace des observations. Toutefois, nous avons également simulé le calcul de politique dans un environnement partiellement observable, et nous avons pu apprécier l'optimalité de la politique résultante calculée à partir d'un MDP subjectif. Nous pensons, que sous certaines hypothèses, travailler sur les observations n'interdit pas l'existence d'une politique déterministe optimale, et que bien évidemment cela dépend de la qualité de la fonction de perception $\mathcal{O}$. Idéalement, cette fonction d'observation devrait permettre de limiter les ambiguïtés sur les observations contenant des informations importantes pour la recherche d'une politique optimale. Dans ces conditions, l'algorithme précédent nous permettrait de converger vers un équilibre de Nash. Nous discuterons du choix des observations dans les perspectives de ce document.

Enfin, remarquons que le calcul de T implique maintenant la considération de n politiques différentes. Ce qui contribue à augmenter la complexité de résolution de cette procédure.

4.6 Comportement réactifs des agents à l'exécution

Dans les sections précédentes nous avons vu comment un agent peut calculer sa politique qu'il soit dans un système multi-agents homogène ou hétérogène, en utilisant les propriétés d'empathie et de subjectivité de manière décentralisée.

Lors de la phase d'exécution, à chaque pas de temps, un agent a un comportement très simple :

- il perçoit une situation o ;
- il choisit l'action correspondante en appliquant uniquement sa propre politique ;
- il agit.

Durant cette phase d'exécution l'agent n'a pas de représentation du but qu'il doit atteindre. Il ne communique pas avec les autres agents. Il peut seulement percevoir une partie de son environnement. Suivant sa propre politique, il agit réactivement.

Le processus de résolution se termine lorsque les agents ne font plus d'actions. D'un point de vue centralisé, cette situation apparaît lorsque tous les agents ont atteint leur but. D'un point de vue des SMA, la solution est un phénomène émergent.

4.7 Conclusion

Dans ce chapitre, nous avons étudié indépendamment les deux propriétés de subjectivité et d'empathie de notre approche de conception descendante de systèmes multi-agents. Et nous avons proposé différents algorithmes de conception impliquant ces deux principes.

La localité est une propriété essentielle des agents qui constituent les systèmes multi-agents. Concevoir un système multi-agents nécessite donc de prendre en compte cette caractéristique qui nous entraîne dans la prise en compte des perceptions partielles. En dépit de leur adéquation théorique, nous avons pris le parti de ne pas utiliser le formalisme des POMDPs au profit de la simplicité, car leur complexité de résolution dans un contexte multi-agents les rendent inexploitables. Nous avons proposé de contourner la difficulté en choisissant le formalisme des MDPs subjectifs qui dans des conditions d'utilisation favorables (ambiguïté minimale des perceptions, connaissance de l'évolution de l'environnement), nous permettent d'envisager le calcul de politiques de qualité appréciable. De plus, travailler sur l'espace des états subjectifs, nous permet de réduire la complexité de résolution d'un MDP, motivés par notre contexte d'étude multi-agents, nous perdons toutefois les propriétés de convergence vers la politique optimale.

L'empathie est la propriété qui nous permet de prévoir le comportement des agents dans le calcul de politiques jointes ou individuelles. Elle est l'élément coordinateur de leur planification d'actions. Inspirés du comportement biologique, nous avons proposé deux algorithmes de co-évolution, l'un alternatif, l'autre simultané, dans un contexte complètement observable et coopératif. Nous avons montré que tous deux convergeaient vers un équilibre de Nash. Ce résultat théorique constitue une base de référence dans notre approche, et nous laisse espérer que dans des conditions favorables d'observabilité partielle, ces algorithmes livreront des politiques de qualité intéressante.

Enfin, nous avons lié ces deux propriétés en proposant plusieurs méthodes de conception descendantes de SMA coopératifs. Selon les exigences du problème à résoudre, nous sommes en mesure de concevoir, de manière centralisée ou décentralisée, des systèmes multi-agents aux comportements réactifs à l'exécution, connaissant le problème d'optimalité que nos agents coopérants seront amenés à résoudre. De plus, remarquons que le principe de la méthode proposée définit une heuristique de résolution du formalisme DEC-POMDP et permet ainsi de contourner sa complexité.

Chapitre 5

Expérimentations

Notre étude théorique nous a donné des preuves de convergence de certains de nos algorithmes comme les algorithmes itératifs de co-évolution simultanée ou alternative. Les complexités de résolution des procédures ne nous ont malheureusement pas encore permis de les expérimenter.

Toutefois, nous proposons dans ce chapitre d'étudier l'influence des principes sur lesquels nos algorithmes reposent : la subjectivité et l'empathie. Ainsi, nous proposons d'évaluer la performance de l'une et l'autre en réalisant quelques simulations. Les résultats de ces simulations nous permettrons d'apprécier et de mieux cerner les avantages et les limites de notre approche dans son ensemble. En particulier, nous resterons prudents sur les conditions de mise en application des solutions obtenues par le calcul de politiques subjectives.

Organisation du chapitre

Dans une première section, nous présentons une étude expérimentale sur l'évaluation de la qualité des politiques calculées en utilisant un MDP subjectif dans un contexte mono-agent. Puis, nous évaluerons les performances d'un système multi-agents avec le dernier algorithme présenté. Bien que ce dernier ne présente pas de convergence théorique vers un équilibre de Nash, nous apprécierons les effets de l'empathie. Enfin, à la connaissance de ces nouvelles données nous pourrons envisager, dans le chapitre suivant, les perspectives de ce travail.

5.1 Processus Décisionnel de Markov subjectif

Dans le chapitre 4, nous avons brièvement présenté un exemple de plan calculé par un MDP subjectif. Nous proposons dans les paragraphes suivants d'approfondir les limites et intérêts de leur utilisation.

5.1.1 Conception de la politique universelle

A titre expérimental, nous avons calculé la politique d'un agent possédant des perceptions partielles de son environnement et dont la tâche à résoudre consiste à capturer une proie, quelles que soient la taille et la topologie de l'environnement dans lequel l'agent sera plongé.

Etats subjectifs

Nous gardons le principe d'état subjectif mentionné dans le chapitre 4. Si l'agent doit prévoir la présence d'un obstacle et d'une proie dans ses perceptions précises, l'ensemble des états subjectifs de notre MDP adapté compte près de 6400 états subjectifs. Soit 3 fois plus que la cardinalité de l'ensemble des états que compterait un problème MDP dans un environnement précis de taille 7×7 avec deux obstacles et une proie.

Travailler sur des états subjectifs augmente la complexité de résolution du MDP, en effet il s'agit de prévoir toutes les occurrences projetées possibles de l'environnement en accord avec les propriétés de perception de l'agent.

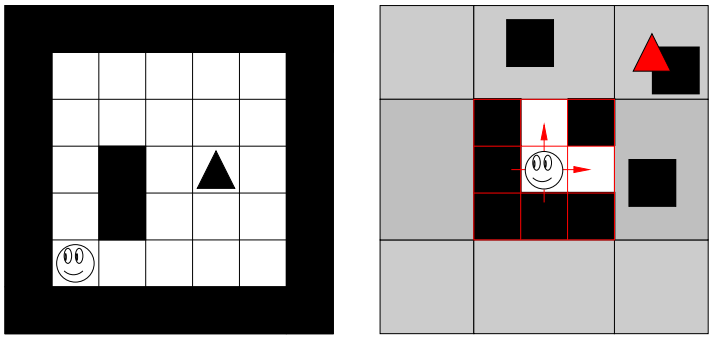


FIG. 5.1 – Perception partielle et subjective de l'environnement de l'agent.

La figure 5.1 reprend l'exemple d'état subjectif déjà étudié dans le chapitre 4. Le triangle symbolise cette fois la proie à capturer. Nous avons choisi de conserver une précision complète jusqu'à une distante égale à 1. L'importance de cette précision de perception prendra toute sa signification dans un environnement encombré d'obstacles.

Récompense

La fonction de récompense est égale à 0 si l'agent est sur la proie, -1 sinon.

Estimation de T

Lorsque l'environnement est connu, c'est-à-dire lorsque l'agent sait dans quel environnement il sera amené à évoluer, l'estimation de T consiste en une projection adaptée des états vers les perceptions partielles (observations). Lorsque l'environnement n'est pas précisé, il convient d'apprendre T ou bien de l'estimer. Du degré d'estimation de T dépendent les performances des politiques ainsi calculées.

Désireux d'étudier le comportement du MDP subjectif proche de ses limites, nous avons choisi d'estimer T de manière assez approximative et sans connaissance au préalable de la topologie de l'environnement. Cela se traduit simplement par l'énumération des perceptions futures possibles à partir d'un état perceptif et d'une action, et ceci de manière équiprobable.

5.1.2 Evaluation

La figure 5.2 montre l'application dans un environnement de la politique calculée indépendamment de l'environnement. Ces différents plans sont obtenus par projection de l'agent dans toutes ses positions possibles dans l'environnement.

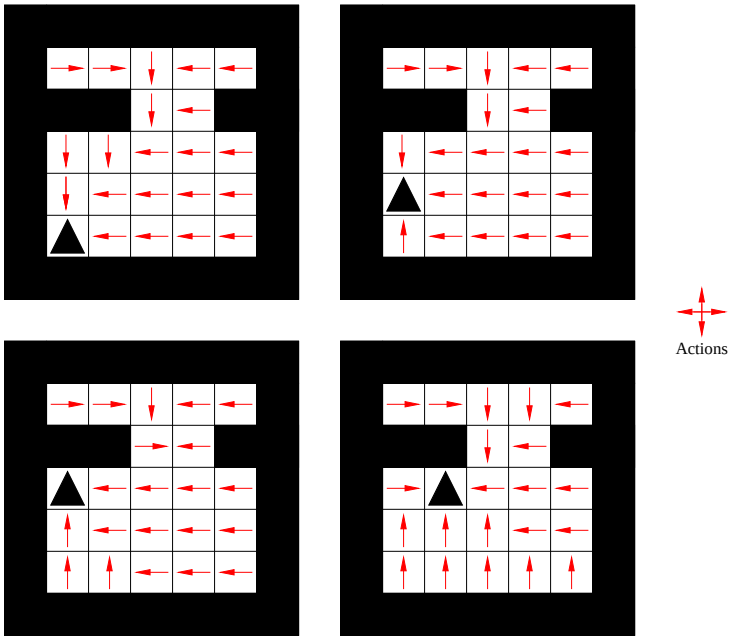


FIG. 5.2 – Evaluation de la politique universelle calculée par la résolution d'un MDP subjectif.

Comme l'on pouvait s'y attendre, dans certaines situations, la politique calculée ne donne pas l'action optimale. C'est effectivement le cas lorsque l'agent ne perçoit pas suffisamment son environnement. Il ne bénéficie alors pas d'une information suffisante qui lui permettrait d'agir au mieux. Typiquement, dans la troisième configuration de la figure 5.2, l'agent préfère revenir dans un état perceptif de plus grande valeur (il percevra alors le prédateur à sa gauche et sans obstacles directs). Pour pallier ce manque de précision, il faut soit affiner le calcul de T et rendre compte du peu d'obstacles présents dans l'environnement, soit changer la fenêtre de perception de l'agent : une perception exacte jusqu'à une distance égale à 2 permettrait de contourner avec plus d'efficacité des obstacles de petites tailles.

Il convient de remarquer le comportement sans faute de l'agent dans un environnement sans obstacle et ce quelle que soit sa taille : l'agent se dirige immanquablement vers sa proie.

5.1.3 Analyse

Ces simulations ont montré les limites du calcul d'une politique subjective. Cette dernière n'est pas toujours optimale. Cependant, elle octroie une grande liberté d'évolution d'un agent dans un grand nombre d'environnements, pour peu que les caractéristiques du MDP subjectif le reflètent. Nous pensons qu'il faut judicieusement choisir les facteurs de précision selon les caractéristiques de l'environnement : encombrement, taille de l'environnement, obstacles de grandes longueurs.

Ce choix réalisé, un des avantages de la politique ainsi calculée est qu'elle permet simplement d'être appliquée dans un ensemble d'environnements de topologies semblables, sans avoir à effectuer de nouveau le calcul de la politique.

Enfin, nous ferons remarquer l'intérêt d'un calcul de politique stochastique. Etant dans une situation d'équivalence ou de blocage, tenir compte d'une action de moins bonne qualité permettrait à l'agent d'accomplir sa tâche. Dans [Kwee *et al.*, 2001], les auteurs proposent une méthode de planification (GREP[19]) fondée sur un calcul de renforcement du gradient (principe présenté dans le chapitre 2) qui planifie et améliore sa politique avant que l'agent agisse dans son environnement. Nous ferons référence à cette direction de recherche dans nos perspectives de travail.

5.2 Poursuite de proie

L'application choisie pour valider notre approche est la poursuite de proie par des prédateurs [Benda *et al.*, 1986] [Korf, 1992][Stone et Veloso, 2000]. Ce problème académique est souvent utilisé. Il bénéficie d'une grande variété d'approches et permet des instantiations différentes pour illustrer différents scénari des systèmes multi-agents. Ce problème implique le déplacement d'agents dans un monde, il est donc particulièrement approprié comme abstraction de SMA robotique. La simplicité de l'environnement permet de concrétiser beaucoup de concepts élémentaires.

Le problème de la poursuite de proie a été introduit par [Benda *et al.*, 1986].

5.2.1 Paramètres

Les paramètres autour desquels s'articule notre application "poursuite de proie" sont les suivants :

- La tâche à résoudre par les prédateurs est la capture d'une proie. Celle-ci est réalisée lorsque les prédateurs ont encerclé la proie.
- L'environnement dans lequel évoluent les différents protagonistes est assimilé à un monde discrétisé, de type grille, torique.
- Les mouvements des prédateurs et de la proie sont orthogonaux. Les acteurs se déplacent simultanément. Tandis que les prédateurs suivent leur propre politique individuelle, la proie se déplace aléatoirement. Tout comme en robotique mobile, les mouvements des prédateurs sont soumis à de fortes incertitudes. Ces derniers sont contraints par la qualité de perception de l'environnement. Comme le montre la figure 5.3, ces perceptions sont exactes jusqu'à une certaine distance, elles deviennent floues au-delà.
- Enfin, la communication explicite entre prédateurs est interdite.

5.3 Modélisation d'une population homogène

Chaque agent i est modélisé par un MDP subjectif.

5.3.1 Définition des MDP subjectifs

A chaque agent i on fait correspondre un MDP subjectif M_i définit comme suit :

[19] "Gradient-based REinforcement Planning".

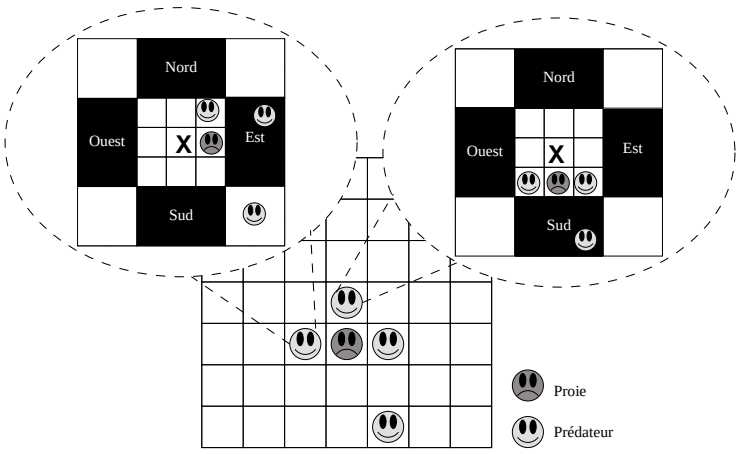


FIG. 5.3 – Proie-prédateurs, exemples d'états subjectifs

- $S = \{s_1, .., s_k, .., s_m\}$ l'ensemble des états subjectifs. Avec $s_k = \langle$ position proie , { positions des autres prédateurs}$\rangle$. Les positions de la proie ou des agents correspondent soit à un numéro de cellule occupée, soit à l'identification d'une région (cellules agrégées).
- $A = \{Nord, Est, Sud, Ouest, Stop\}$ l'ensemble des actions.
- La récompense individuelle d'un agent est égale à 1 s'il perçoit la situation suivante :
  - La proie est devant moi à une distance 1,
  - Il y a un prédateur de chaque côté de la proie,
  - Le dernier prédateur est quelque part dans une région de l'autre coté de la proie.

  La figure 5.3 illustre pour l'agent le plus haut une perception d'état but. Dans ce cas, sa récompense sera égale à 1. D'un point de vue global, l'ensemble des buts individuels des prédateurs identifie clairement la tâche collective à résoudre.

5.4 Calcul de politiques

La politique est calculée par notre algorithme de co-évolution décrit section 4.2 pour une population homogène. Nous avons exécuté le processus de calcul de plan avec différents paramètres. L'algorithme est toujours initialisé avec une politique statique (les agents ne bougent pas).

5.4.1 Amélioration des politiques

La figure 5.4 décrit l'évolution des politiques à travers le nombre d'actions mises à jour entre deux politiques successives pendant le déroulement de l'algorithme choisi.

Pendant les premières itérations, un agent met à jour la plupart des couples perceptions-actions. Après dix itérations en général, le nombre de mises à jour devient pratiquement constant : la plus grande partie de la politique coordonnée est planifiée rapidement. La partie restante concerne les états conflictuels.

5.5 Simulations

Nous montrons deux évaluations des politiques successivement calculées. La première met en jeu des agents ayant tous la même politique individuelle, la seconde met en jeu des agents suivant

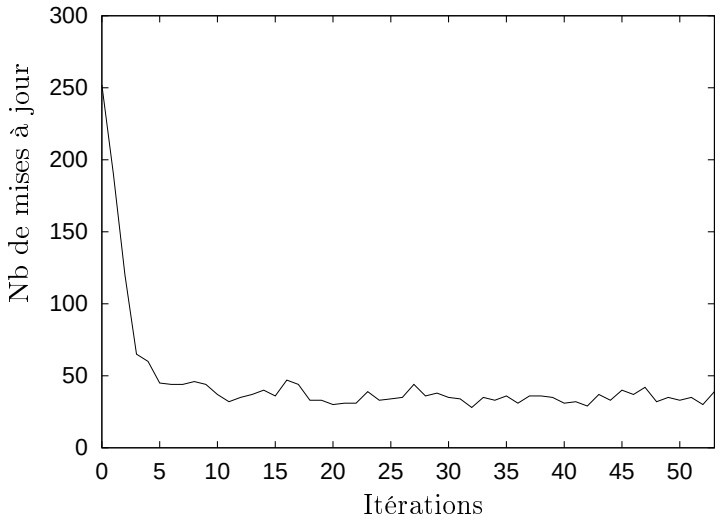


FIG. 5.4 – Nombre de mises à jour.

tous la même politique individuelle sauf un leader qui suit la politique suivante. Chacune montre le nombre moyen et le pire cas en nombre de pas de simulation nécessaire à la capture de la proie. Ces statistiques ont été réalisées sur la base de 100000 exécutions à partir de configurations de départ aléatoires.

5.5.1 Sans bruit

Les courbes de la figure 5.5 représentent l'évaluation de chacune des politiques calculées au cours de l'application de l'algorithme dans le cas où les agents ne sont pas confrontés aux bruits de leurs déplacements.

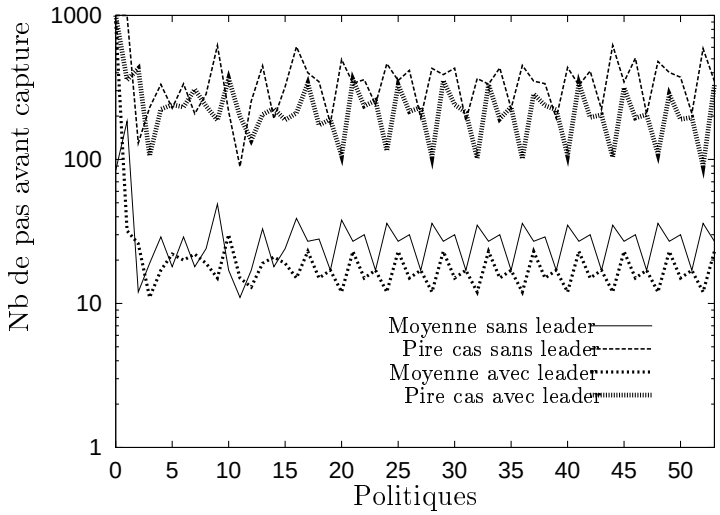


FIG. 5.5 – Performances sans bruit.

Fort logiquement, les performances obtenues avec un leader en moyenne et dans le pire cas

sont presque toujours meilleures que sans leader. Ces courbes reflètent également des oscillations non négligeables qui semblent régulières en terme de fluctuations de performances.

5.5.2 Avec bruit

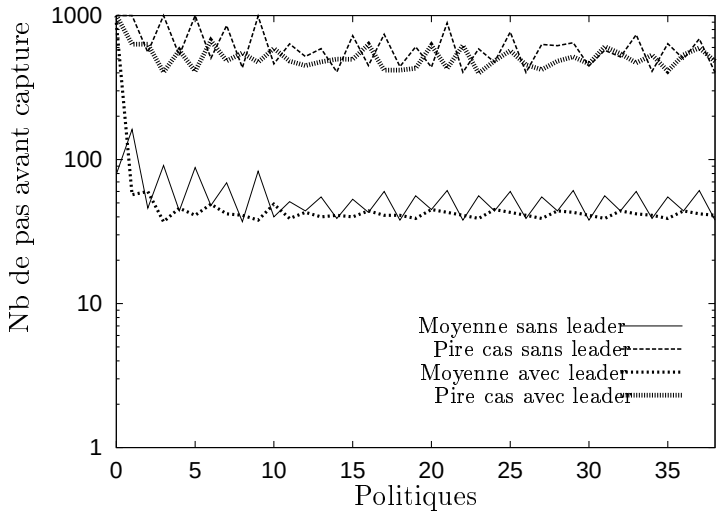


FIG. 5.6 – Performances avec bruit.

Avec bruit, nous constatons également les meilleures performances du système avec leader (figure 5.6). L'insertion de bruits dans le comportement des agents et le calcul de politiques rend à la fois le problème plus difficile en terme de nombre de pas avant capture mais il permet également d'atténuer les oscillations précédemment observées.

5.5.3 Analyse

De manière générale, nous pouvons faire les remarques suivantes :
- Les performances deviennent rapidement meilleures. La meilleure politique parmi l'ensemble des politiques calculées en terme de performance n'appartient pas au cycle de politique vers lequel l'algorithme a convergé[20].
- Le bruit permet d'atténuer les grandes oscillations de performances et tend à homogénéïser les résultats obtenus.
- En accord avec nos prédictions théoriques, utiliser un leader donne de meilleurs résultats.

5.6 Mise en valeur de la coordination

Nous nous intéressons maintenant à mettre en valeur la coordination émergente des actions de nos agents : anticiper les actions des autres agents est plus efficace que de ne rien prévoir.

5.6.1 Prévoir ou non

A cet effet, nous avons calculé les politiques des agents lorsqu'ils considèrent que leurs semblables sont immobiles. Puis nous avons simulé et mesuré les performances lors d'une *exécution*

[20] Ce cycle comprend en moyenne les dix dernières politiques étudiées.

séquentielle : lorsqu'un agent se déplace, les autres demeurent immobiles. Ainsi, à chaque pas de temps, les uns après les autres, chaque agent décide, à partir de ses perceptions instantanées, une action à exécuter. On compte un pas d'exécution lorsque tous les agents ont effectué leur déplacement (tableau 5.1).

	3 prédateurs grille 7x7	3 prédateurs grille 9x9	4 prédateurs grille 7x7
Sans bruit	28.66 ~ 340	71.73 ~ 1187	1534.07 > 10000
Avec bruit	53.08 ~ 510	97.64 ~ 1073	3180.60 > 10000

TAB. 5.1 – Moyenne du nombre de déplacements et pire cas en situation d'*exécution séquentielle*.

Le tableau 5.1 résume les résultats observés. Les moyennes obtenues sont de moins bonne qualité compte tenu de la difficulté progressive des simulations. En moyenne, $28,66$ déplacements pour trois prédateurs évoluant dans une grille torique de dimension 7×7. Notons que dans le cas de quatre prédateurs, les prédateurs sont parfois en situation d'échecs (pire cas > 10000), dans ce cas la moyenne n'est pas un outil adapté. Nous retiendrons la faible efficacité des politiques ainsi calculées.

	3 prédateurs grille 7x7	3 prédateurs grille 9x9	4 prédateurs grille 7x7
Sans bruit	11.24 89	15.02 127	11.19 91
Avec bruit	37.72 403	61.44 759	129.17 >1000

TAB. 5.2 – Moyenne du nombre de déplacements et pire cas en situation d'*exécution simultanée*.

En guise de comparaison, nous avons simulé les politiques calculées pour une population homogène sans leader (tableau 5.2). Les performances sont à tous niveaux meilleures.

5.6.2 Analyse

Bien qu'une exécution séquentielle soit un problème *a priori* plus facile à résoudre (les agents décident tour à tour d'une action, ce qui permet d'éviter de nombreux conflits) les résultats obtenus pour cette simulation sont nettement inférieurs à ceux obtenus lorsque les agents agissent simultanément et sans leader en ayant prévu dans le calcul de leur politique le comportement de leurs semblables. Ce résultat met clairement en évidence l'intérêt de l'empathie dans le calcul de politique, et nous laisse présager de meilleurs résultats quant à l'application de nos algorithmes théoriques.

5.7 Conclusion

Le calcul approché d'une politique universelle pour l'évitement d'obstacles et la poursuite d'une cible mobile par la résolution d'un MDP subjectif nous a révélé l'importance du choix de la fenêtre de précision. Cette dernière dépend de la méthode d'agrégation d'états choisie

et des capacités de l'agent simulé. En accord avec la précision des informations disponibles, la politique universelle ainsi calculée donne des résultats appréciables. En réponse aux décisions sous-optimales, nous pensons que planifier une politique stochastique [Kwee *et al.*, 2001] peut remédier aux situations de blocage dues à la faible qualité de perception de l'agent. Nous garderons en mémoire cette perspective de recherche.

Enfin, nous avons simulé l'algorithme de calcul de politique individuelle pour une population homogène non communicante. Cet algorithme nous a permis de simuler le comportement d'une population d'agents homogènes. Comme nous l'avions fait remarquer dans le chapitre précédent, les résultats obtenus sont meilleurs avec un leader : un agent suit la politique π_{t+1} tandis que les autres suivent la politique π_t. Enfin, nous avons mis en évidence la coordination des agents par planification : calculer les plans en prévoyant le comportement des autres agents donne de meilleurs résultats que de ne pas le faire.

Bien que nous n'ayons pas réalisé des simulations pour tous nos algorithmes, l'analyse des résultats obtenus par l'algorithme possédant les qualités les moins intéressantes nous permet de valider notre approche de conception ascendante fondée sur les propriétés de subjectivité et d'empathie.

Conclusion

Cette thèse fait le lien entre deux domaines distincts de l'intelligence artificielle. D'un coté, les systèmes multi-agents, discipline nouvelle dont l'une des préoccupations de recherche actuelles est de formaliser les comportements des agents et/ou du système. De l'autre la théorie de la décision et en particulier les modèles décisionnels de Markov, très utilisés en planification et en apprentissage de comportement d'agents autonomes, mais limités par les complexités des méthodes de résolutions lorsqu'il s'agit de calculer le comportement d'un système de plusieurs agents.

1 Résumé du travail réalisé

Deux aspects caractérisent ce travail. Tout d'abord, l'apport d'un formalisme pour la conception d'agents aux comportements réactifs pour des problèmes mettant en jeu des systèmes multi-agents coopératifs cherchant à accomplir une tâche collective nécessitant la coordination des agents.

Le deuxième point porte sur l'approximation d'un modèle décisionnel de Markov décentralisé partiellement observable (DEC-POMDP) dont la complexité est qualifiée de NEXP pour un nombre d'agents supérieur ou égal à deux. Dans ce contexte, l'approche proposée dans cette thèse est une heuristique pour résoudre ce problème.

Enfin, ce qui relie ces deux composantes sont les propriétés de subjectivité (localité) et d'empathie (coordination) que nous avons prêté à nos agents et qui nous ont permis de concevoir des agents capables de résoudre les problèmes pour lesquels ils ont été conçus.

Concevoir des systèmes multi-agents aux comportements réactifs

Nos agents, doués de capacités de perception et d'actions locales plongés dans un environnement et devant résoudre une tâche collective, sont sujets à des comportements incertains. Alors que les méthodes de résolution de problèmes fondés sur l'utilisation de systèmes multi-agents réactifs adoptent des approches ascendantes, nous avons présenté une approche descendante qui prend en compte cette incertitude.

La conception de nos agents repose sur les recherches que nous avons mené sur le DEC-POMDP. Après avoir cerné les complexités de chacun des modèles décisionnels de Markov, nous nous sommes attachés à exploiter les propriétés de nos agents : la subjectivité et l'empathie.

Approximation d'un problème non Markovien par un MDP

Avant de nous attacher au cas des systèmes multi-agents, nous nous sommes rapportés à l'étude d'un phénomène biologique réel dans un cadre mono-agent. Nous avons ainsi étudié l'adéquation d'un formalisme MDP pour la prédiction de comportements d'une araignée. Cette approximation d'un problème non "Markovien", dans un cadre mono-agent, nous a permis de déterminer, sur la base d'un critère de performance lié à la gestion de ressources, le comportement optimal que devrait suivre l'araignée d'après nos hypothèses. Par comparaison directe, les résultats obtenus sur le terrain ont validé l'intérêt de cette approche.

Systèmes multi-agents et agents

Dans un premier temps, nous avons précisé quelle sorte de systèmes multi-agents nous nous proposions de concevoir. A cet effet, nous avons défini notre SMA coopératif doté d'agents aux propriétés de localité : perceptions, actions, plans réactifs. Un point important de cette définition est qu'elle permet de spécifier une fonction de récompense globale qui identifie la tâche collective que les agents se doivent d'accomplir. Cette fonction générale autorise la définition des fonctions de récompense individuelle de nos agents de manière à maintenir la coopération de l'ensemble.

Subjectivité

La subjectivité, reflet de la propriété de localité d'un agent, intègre les principes de perceptions partielles de l'environnement et d'action locale. Nous avons présenté notre modèle de MDP subjectif, dans lequel nous rendons à l'agent l'autonomie dont il devra faire usage dans le système multi-agents. Cette approximation d'un MDP peut se révéler judicieuse dans des conditions favorables : connaissance suffisante de l'environnement, agrégation d'états adaptée, etc ...

Nous avons montré que, sous certaines hypothèses sur la complexité de l'environnement, utiliser un MDP subjectif afin de calculer une stratégie individuelle de comportement était intéressant. En effet, la qualité des comportements produits qui peuvent s'avérer insuffisant dans un contexte mono-agent, est dans une certaine mesure très compétitive dans un cadre multi-agents comparativement à la complexité des systèmes dédiées de type MMDPs inutilisables concrètement.

Empathie

L'empathie est la propriété qui permet à nos agents de coordonner leurs actions en tenant compte du comportement de leurs semblables.

Notre travail sur l'empathie des agents, dans un contexte de systèmes multi-agents coopératifs et où l'environnement est complètement observable, s'est traduit par la conception de deux algorithmes itératifs de co-évolution. Tous les deux convergent vers un ensemble de politiques individuelles possédant les propriétés d'un équilibre de Nash. Tandis que l'algorithme de co-évolution alternative converge en moyenne vers des équilibres de Nash de moins bonne qualité, il bénéficie à chaque itération d'un processus de résolution de moindre complexité (celle d'un processus décisionnel de Markov). L'algorithme simultané, quant à lui, nous permet d'explorer une partie des optimums locaux, mais sa complexité bien que moins élevée que celle d'un MMDP centralisé reste importante.

La propriété d'empathie prend toute son importance lorsque l'on considère la localité des agents. Elle devient alors le moyen de coordonner les agents dans la conception de plans réactifs.

Conception descendante de SMA : subjectivité et empathie

Enfin, utilisant nos résultats théoriques sur la subjectivité et l'empathie, nous avons adapté nos algorithmes au cas de perception partielle et de contrôle décentralisé.

Ainsi, il nous est maintenant possible, connaissant les paramètres d'un problème d'optimisation à résoudre, de concevoir des agents aux comportements réactifs et capables de se coordonner avec les autres agents grâce à leurs capacités empathiques.

Nous avons proposé deux algorithmes centralisés pour la conception de systèmes multi-agents homogènes et hétérogènes. Nos réflexions sur les contraintes de conception de systèmes multi-agents nous ont amené à en proposer une version décentralisée. Ainsi, selon la configuration du problème nous sommes en mesure de faire se coordonner nos agents dans une phase d'apprentissage de comportements réactifs (co-évolution) où nous avons autorisé la communication afin de favoriser la propriété d'empathie.

Enfin, nous avons mis en valeur, au cours des simulations que nous avons réalisées sur le problème académique "poursuite de proie", la coordination effective de nos agents conçus par l'algorithme le moins performant que nous avons proposé.

2 Perspectives et discussions

Nos perspectives se rapportent à des aspects théoriques et applicatifs de notre travail.

Une des propriétés fondamentales de notre modèle de conception repose sur la possibilité de partager une récompense collective globale du système multi-agents considéré en récompense individuelle de manière à ce que la propriété d'intéraction de type coopération demeure. Notre travail de recherche à venir pourrait s'intéresser à la question du comment automatiser ce découpage des récompenses globales en récompenses individuelles. Les travaux actuels sur le sujet ne donnent, à ce jour, pas de réponse à cette question [Wolpert et Tumer, 1999].

Les simulations de MDP subjectifs ont révélé les limites d'une telle approximation lorsque l'agent ne possédait pas assez d'information sur les caractéristiques de son environnement. Travailler sur des techniques de planification de politiques subjectives et stochastiques serait plus approprié compte tenu de l'observabilité partielle. En complément de notre démarche de planification, nous envisageons également dans les cas de manque d'information, d'allier à notre processus de planification une phase d'apprentissage.

Notre approche de conception de SMA met en avant la possibilité de concevoir un SMA au comportement réactif connaissant les données d'un problème sous certaines conditions, et en particulier celle de pouvoir écrire le modèle sous la forme de notre définition de SMA. Nous désirons à présent approfondir l'étude des approches intelligence collective ascendante qui utilisent également des agents réactifs. Quel genre de problèmes communs notre approche sait-elle résoudre ? Est-il nécessaire de pouvoir les exprimer en fonction d'un critère d'optimalité et d'une récompense globale ? Dans [Meuleau et Dorigo, 2000], les auteurs font apparaître un lien entre

les techniques qui utilisent la quantité de phéromones déposés sur un état et d'un point de vue décisionnel l'utilité de cet état. Cette direction de recherche pourrait faire l'objet d'une étude plus approfondie dans notre contexte. La recherche des limites et de l'adéquation des modèles décisionnels de Markov dans les SMA pose la question de déterminer avec précision quel type de problème SMA nous pouvons résoudre avec notre approche ? En formalisant le type de SMA auxquels nous nous intéressions nous avons répondu en partie à cette question : tout problème SMA pouvant s'exprimer de cette manière est *a priori* concevable par notre approche. Reste à définir quels problèmes peuvent s'exprimer de cette façon ...

Enfin, dans un proche avenir, nous chercherons à appliquer nos algorithmes les plus intéressants. Il nous faut pour cela, avoir à disposition ou apprendre les transitions du monde d'une plate-forme applicative de type multi-robots. Ainsi, le travail effectué avec Samuel Venner sur la gestion de l'énergie correspond aux difficultés rencontrées par les robots qui doivent économiser leur niveau d'énergie. Nous chercherons également à généraliser l'utilisation des principes de gestion d'énergie dans ce contexte multi-robots.

Pour conclure...

Les SMA d'inspiration biologique ont un objectif de simulation de comportements qui répond à la question du "comment". Ils permettent ainsi de comprendre et de mettre en évidence certains comportements de groupe. Cependant, c'est aussi la perspective d'une réutilisation de ces modèles simples qui les rend intéressants. Or comment savoir s'ils sont adaptés à la résolution d'un problème sans connaître leur évolution collective et les objectifs qu'ils peuvent atteindre ? C'est là tout le paradoxe des SMA d'inspiration biologique : ne pas savoir à l'avance s'ils seront capables de résoudre une tâche. Au contraire, notre approche conçoit des agents réactifs, et nous savons dès la formulation d'un problème, si tant est que ce dernier corresponde à la définition de notre système, que ses agents seront construits dans le but de sa résolution. Telle est notre contribution.

Bibliographie

[Baxter *et al.*, 2001] J. Baxter, P. Bartlett, et Lex Weaver. Experiments with infinite-horizon, policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15 :351–381, 2001.

[Baxter et Bartlett, 2001] J. Baxter et P. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15 :319–350, 2001.

[Bellman, 1957] R.E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, New Jersey, 1957.

[Benda *et al.*, 1986] M. Benda, V. Jagannathan, et R. Dodhiawala. On optimal cooperation of knowledge sources - an empirical investigation. Rapport technique, Boeing Advanced Technology Center, July 1986.

[Bernstein *et al.*, 2000] D. S Bernstein, S. Zilberstein, et N. Immerman. The complexity of decentralized control of markov decision processes. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, July 2000.

[Bonabeau *et al.*, 1999] Eric Bonabeau, Marco Dorigo, et Guy Theraulaz. *Swarm Intelligence : From Natural to Artificial Systems.* SFI Studies in the Science of Complexity, Oxford University Press, New York, NY, 1999.

[Bourdon, 1997] François Bourdon. The interactional semantics of knowledge. In *Proceedings of the poster session of the 15th International Joint Conference on Artificial Intellligence*, pages 23–29, Nagoya, Japan, August 1997. Martha E. Pollack, Ed.

[Boutilier, 1999] Craig Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 1999.

[Brooks, 1986] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1) :14–23, Mars 1986.

[Buffet *et al.*, 2002] O. Buffet, A. Dutech, et F. Charpillet. Adaptive combination of behaviors in an agent. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'02)*, éditeur "Frank van Harmelen", pages "48–52". "IOS Press", juillet 2002.

[Burago *et al.*, 1996] Dima Burago, Michel de Rougemont, et Anatol Slissenko. On the complexity of partially observed Markov decision processes. *Theoretical Computer Science*, 157(2) :161–183, 1996.

[Camps et Gleizes, 1995] Valérie Camps et Marie-Pierre Gleizes. Principe et évaluation d'une méthode d'auto-organisation. In *Journées francophones IAD&SMA*, pages 337–348, Saint Baldoph, mars 1995.

[Camps et Gleizes, 1996] Valérie Camps et Marie-Pierre Gleizes. Cooperative and mobile agents to find relevant information in a distributed ressources network. In *Workshop on Artificial Intelligence-based tools to help W3 users, Fifth international conference on World Wide Web*, pages 337–348, Paris, France, mai 1996.

[Cassandra *et al.*, 1994] Anthony R. Cassandra, Leslie Pack Kaelbling, et Michael L. Littman. Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, volume 2, pages 1023–1028, Seattle, Washington, USA, 1994. AAAI Press/MIT Press.

[Cassandra *et al.*, 1997] Anthony R. Cassandra, Michael L. Littman, et N. L. Zhang. A simple, fast, exact method for partially observable markov decision processes. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)*, pages 54–61, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

[Cassandra, 1998] Anthony R. Cassandra. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes.* Thèse de Doctorat, Brown University, Departement of Computer Science, Providence, RI, 1998.

[Chadès *et al.*, 2002] Iadine Chadès, Bruno Scherrer, et François Charpillet. A heuristic approach for solving decentralized-pomdp : Assessment on the pursuit problem. In *Proceedings of the 2002 ACM Symposium on Applied Computing*, 2002.

[Chadès, 1998] Iadine Chadès. Un modèle de coordination pour la résolution de problèmes distribués sous contraintes de ressources. Mémoire de DEA, Ecole Normale Supérieure de Lyon, 1998.

[Cheng, 1988] Hsien-Te Cheng. *Algorithms for Partially Observable Markov Decision Processes.* Thèse de Doctorat, University of British Columbia, British Columbia, Canada, 1988.

[Clark et Mangel, 1988] C.W. Clark et M. Mangel. *Dynamic State Variable Models in Ecology. Methods and applications.* Oxford University Press, Oxford, U.K., 1988.

[Dean *et al.*, 1995] Thomas Dean, Leslie Pack Kaelbling, Jak Kirman, et Ann Nicholson. Planning under time constraints in stochastic domains. *Artificial Intelligence*, 76(1-2) :35–74, 1995.

[Dean *et al.*, 1997] T. Dean, R. Givan, et S. Leach. Model reduction techniques for computing approximately optimal solutions for Markov Decision Processes. In *Uncertainty in Artificial Intelligence*, pages 124–131, 1997.

[Demazeau, 1995] Yves Demazeau. From Interactions to Collective Behaviour in Agent-Based Systems. In *Proceedings of the First European Conference on Cognitive Science*, Saint-Malo, France, Avril 1995.

[Deneubourg *et al.*, 1990] J.-L. Deneubourg, S. Aron, S. Goss, et J.-M. Pasteels. The self-organizing exploratory pattern of teh argentine ant. *Journal of Insect Behavior*, 3 :159–168, 1990.

[Dorigo *et al.*, 1991] M. Dorigo, V. Maniezzo, et A. Colorni. Positive feedback as a search strategy. Rapport Technique 91-016, Politecnico di Milano Dipartimento di Elettronica, Milan, Italie, 1991.

[Dorigo et Di Caro, 1999] Marco Dorigo et Gianni Di Caro. Ant colony optimization : A new meta-heuristic. In *Proceedings of the Congress on Evolutionary Computation*, éditeurs Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, et Ali Zalzala, volume 2, pages 1470–1477, Mayflower Hotel, Washington D.C., USA, 6-9 1999. IEEE Press.

[Drogoul et Ferber, 1993] Alexis Drogoul et Jacques Ferber. From tom-thumb to the dockers : Some experiments with foraging robots. In *From Animals to Animats II*, pages 451–459, Cambridge, 1993. MIT Press.

[Durfee et Lesser, 1991] Edmund H. Durfee et Victor R. Lesser. Partial global planning : A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5) :1167–1183, 1991.

[Dutech, 1999] A. Dutech. *Apprentissage d'environnement : approches cognitives et comportementales.* Thèse de Doctorat, ENSAE, Toulouse, 1999.

[Eberhard, 1986] W. G. Eberhard. *W.A. Shear, ed.*, chapitre Spiders : webs, behavior and evolution., pages 70–120. Stanford University Press, Stanford., 1986.

[Ferber et Müller, 1996] Jacques Ferber et Jean-Pierre Müller. Influences and reaction : a model of situated multiagent systems. In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96).* AAAI Press, 1996.

[Ferber, 1995] J. Ferber. *Les Systèmes Multi-Agents. Vers une intelligence collective.* InterEditions, Paris, 1995.

[Ferber, 1999] J. Ferber. *Multi-Agent Systems. An introduction to Distributed Artificial Intelligence.* John Wiley & Sons Inc., New York, 1999.

[Ferguson, 1992] David E. Ferguson. Bit-tree, a data structure for fast file processing. *CACM*, 35(6) :114–120, 1992.

[Foisel, 1998] Rémy Foisel. *Modèle de réorganisation de systèmes multi-agents : une approche descriptive et opérationnelle.* Thèse de Doctorat, université Henri Poincaré, Nancy 1, Novembre 1998.

[Fox, 1981] Mark S. Fox. An organizational view of distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1) :70–80, January 1981.

[Franklin et Graesser, 1996] Stan Franklin et Art Graesser. Is it an agent, or just a program ? : A taxonomy for autonomous agents. In *Third International Workshop on Agent Theories, Architectures, and Languages.* Springer-Verlag, 1996.

[Goss *et al.*, 1989] S. Goss, S. Aron, J-L Deneubourg, et J-M Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76 :579–581, 1989.

[Grassé, 1959] P. P Grassé. La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes sp.*. la théorie de la stigmergie : Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6 :41–81, 1959.

[Grislin-Le Strugeon *et al.*, 1993] e. Grislin-Le Strugeon, R. Mandiau, et G. Libert. Proposition d'organisation dynamique d'un groupe d'agents en fonction de la tâche. In *Communication présentée aux 1ères Journées Francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, Toulouse, France, Avril 1993.

[Guichard, 1996] Frédéric Guichard. *La réorganisation dynamique dans les systèmes multi-agents.* Thèse de Doctorat, Université de Savoie, 1996.

[Hanna et Mouaddib, 2002] H. Hanna et A. I. Mouaddib. Task selection problem under uncertainty as decision-making. In *Proceedings of the International Joint conference on Autonomous Agents & Multi-Agent Systems, AAMAS02*, pages 1303–1308, 2002.

[Horvitz *et al.*, 1988] Eric J. Horvitz, John S. Breese, et Max Henrion. Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2 :247–302, 1988.

[Houston et McNamara, 1999] A. I. Houston et J. M. McNamara. *Models of adaptive behavior : an approach based on state.* Cambridge University Press, Cambridge, U.K., 1999.

[Howard, 1960] R. A. Howard. *Dynamic Programming and Markov Processes.* The MIT Press, Cambridge, Massachusetts, 1960.

[Hu et Wellman, 1998a] J. Hu et M. Wellman. Multiagent reinforcement learning : theoretical framework and an algorithm. In *Proceedings of the 15th International Conference on Machine Learning (ICML'98)*, pages 242–250, 1998.

[Hu et Wellman, 1998b] J. Hu et M. Wellman. Online learning about other agents in a dynamic multiagent system. In *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 239–246, 1998.

[Ishida *et al.*, 1992] Toru Ishida, Les Gasser, et Makot Yokoo. Organization self-design of distributed production systems. *IEEE Transactions on Data and Knowledge Engineering*, 4(2) :123–134, 1992.

[Jaakkola *et al.*, 1995] Tommi Jaakkola, Satinder P. Singh, et Michael I. Jordan. Reinforcement learning algorithm for partially observable Markov decision problems. In *Advances in Neural Information Processing Systems*, éditeurs G. Tesauro, D. Touretzky, et T. Leen, volume 7, pages 345–352. The MIT Press, 1995.

[Jennings *et al.*, 1998] N. R. Jennings, K. Sycara, et M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1) :7–38, 1998.

[Kearns *et al.*, 2000] Michael Kearns, Yishay Mansour, et Andrew Y. Ng. Approximate planning in large pomdps via reusable trajectories (long version). In *Advances in Neural Information Processing Systems 12*. MIT Press, 2000.

[Kimura *et al.*, 1997] H. Kimura, K. Miyazaki, et S. Kobayashi. Reinforcement learning in pomdps with function approximation. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pages 152–160. D. H. Fisher, 1997.

[Korf, 1992] R. E. Korf. A simple solution to pursuit games. In *Working Papers of the 11th International Workshop on Distributed Artificial Intelligence*, pages 183–194, 1992.

[Kwee *et al.*, 2001] Ivo Kwee, Marcus Hutter, et Juergen Schmidhuber. Gradient-based reinforcement planning in policy-search methods. In *Proceedings of the 5th European Workshop on Reinforcement Learning (EWRL-5)*, pages 27–29, Manno(Lugano), CH, 2001.

[Laroche, 2000] Pierre Laroche. *Processus décisionnels de Markov appliqués à la plannification sous incertitude.* Thèse de Doctorat, Université Henri Poincaré, Nancy, France, 2000.

[Littman *et al.*, 1995] Michael L. Littman, Thomas L. Dean, et Leslie Pack Kaelbling. On the complexity of solving markov decision problems. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI'95)*, Montreal, Québec, Canada, 1995.

[Littman, 1994a] M. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ICML'94), San Fransisco, CA*, 1994.

[Littman, 1994b] Michael L. Littman. Memoryless policies : Theoretical limitations and practical results. In *From Animals to Animats 3 : Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, éditeurs Dave Cliff, Philip Husbands, Jean-Arcady Meyer, et Stewart W. Wilson, Cambridge, MA, 1994. The MIT Press.

[Littman, 1994c] Michael L. Littman. The witness algorithm : Solving partially observable markov decision processes. Rapport Technique CS-94-40, Department of Computer Science, Brown University, 1994.

[Littman, 1996] M. L. Littman. *Algorithms for Sequential Decision Making.* Thèse de Doctorat, Department of Computer Science, Brown University, 1996.

[Lovejoy, 1991] W. S. Lovejoy. A survey of algorithmic methods for solving partially observable markov decision processes. *Annals of Operations Research*, 28(1) :47–65, 1991.

[Madani *et al.*, 1999] Omid Madani, Steve Hanks, et Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence AAAI/IAAI*, pages 541–548, 1999.

[Mahadevan, 1996] Sridhar Mahadevan. An average-reward reinforcement learning algorithm for computing bias-optimal policies. In *Proceedings of the National Conference on Artificial Intelligence AAAI/IAAI, Vol. 1*, pages 875–880, 1996.

[Malone, 1987] T. W. Malone. Modeling coordination in organizations and markets. *Management Science*, 33(10) :1317–1332, 1987.

[Malville, 1999] Eric Malville. *L'auto-organisation de groupes pour l'allocation de tâches dans les Systèmes Multi-Agents : Application à CORBA.* Thèse de Doctorat, Université de Savoie, mars 1999.

[Mangel et Clark, 1988] M. Mangel et C.W. Clark. *Dynamic Modeling in Behavioral Ecology.* Princeton University Press, Princeton, N.J., 1988.

[MARCIA, 1996] Groupe MARCIA. Auto-organisation := evolution de structure(s). In *Journées PRC-GDR Intelligence Artificielle sur le thème des Systèmes Multi-agents*, Toulouse, février 1996.

[McCallum, 1995] Andrew Kachites McCallum. *Reinforcement Learning with Selective Perception and Hidden State.* Thèse de Doctorat, Department of Computer Science, University of Rochester, December 1995.

[McNamara et Houston, 1986] J. M. McNamara et A. I. Houston. The common currency for behavioral decisions. *American Naturalist*, 127 :358–378, 1986.

[Meuleau et Dorigo, 2000] N. Meuleau et M. Dorigo. Ant colony optimization and stochastic gradient descent. Rapport Technique IRIDIA/2000-36, IRIDIA, Universite Libre de Bruxelles, Belgium, 2000.

[Monahan, 1982] G. E. Monahan. A survey of partially observable markov decision processes : Theory, models, and algorithms. *Management Science*, 28(1) :1–16, 1982.

[Müller, 1997] J.P. Müller. Control architectures for autonomous and interacting agents : A survey. In *Intelligent Agent Systems : Theoretical and Practical Issues, Lecture Notes in AI*, éditeurs L. Cavedon, A. Rao, et W. Wobcke, volume 1209. Springer, 1997.

[Munos, 2000] Rémi Munos. A study of reinforcement learning in the continuous case by the means of viscosity solutions. *Machine Learning Journal*, 40(3) :265–299, september 2000.

[Ng *et al.*, 1999] Andrew Y. Ng, Daishi Harada, et Stuart Russell. Policy invariance under reward transformations : theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning (ICML'99)*, pages 278–287. Morgan Kaufmann, San Francisco, CA, 1999.

[Ünsal et Bay, 1994] C. Ünsal et J. S. Bay. Spatial self-organization in large populations of mobile robots. In *IEEE International Symposium on Intelligent Control*, Columbus, Ohio, August 1994.

[Ünsal, 1993] Cem Ünsal. Self-organization in large populations of mobile robots. Mémoire de DEA, Science in Electrical Engineering, Blacksburg, Virginia, may 1993.

[Owen, 1982] G. Owen. *Game Theory : Second Edition.* Academic Press, Orlando, Florida, 1982.

[Papadimitriou et Tsitsiklis, 1987] C. H. Papadimitriou et J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3) :441–450, 1987.

[Peshkin *et al.*, 1996] L. Peshkin, K-E. Kim, N. Meuleau, et L. P. Kaelbling. Learning to cooperate via policy search. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Stanford, California*, July 1996.

[Platzman, 1977] Loren Platzman. *Finite-memory estimation and control of finite probabilistic systems.* Thèse de Doctorat, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1977.

[Puterman, 1994] L. Puterman, M. *Markov Decision Processes.* J. Wiley & Sons, 1994.

[Rao et Georgeff, 1995] A. S. Rao et M. P. Georgeff. Bdi agents : From theory to practice. Rapport Technique 56, Australian Artificial Intelligence Institute, Melbourne, Australia, April 1995.

[Russel et Norvig, 1995] Stuart Russel et Peter Norvig. *Artificial Intelligence : A Modern Approach.* Prentice Hall, 1995.

[Scherrer, 2002] B. Scherrer. A connectionist architecture that adapts its representation to complex tasks. In *International Joint Conference on Neural Networks*, Mai 2002.

[Shapley, 1953] L. S. Shapley. Stochastic games. *National Academy of Sciences of the United States of america*, 39(1095–1100), 1953.

[Sherman, 1994] P. M. Sherman. The orb-web : an energetic and behavioural estimator of spider's dynamic foraging and reproductive strategies. *Animal Behaviour*, 48 :19–34, 1994.

[Simonin, 2001] Olivier Simonin. *Le modèle satisfaction-altruisme : coopération et résolution de conflits entre agents situés réactifs, application à la robotique.* Thèse de Doctorat, Université Montpellier, 2001.

[Singh *et al.*, 1994] Satinder P. Singh, Tommi Jaakkola, et Michael I. Jordan. Learning without state-estimation in partially observable markovian decision processes. In *Proceedings of the International Conference on Machine Learning (ICML'94)*, pages 284–292, 1994.

[Smallwood et Sondik, 1973] R. D. Smallwood et E. J. Sondik. The optimal control of partially observable processes over a nite horizon. *Operations Research*, 21 :1071–1088, 1973.

[Smith, 1980] Reid Smith. The contract net protocol : High level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12) :1104–1113, December 1980.

[So et Durfee, 1993] Y. So et Edmund H. Durfee. An organizational self-design model for organizational change, 1993.

[So et Durfee, 1996] Y. So et Edmund H. Durfee. Designing tree-structured organizations for computational agents. *Computational and Mathematical Organization Theory*, 2(3) :219–246, 1996.

[Sondik, 1971] Edward J. Sondik. *The Optimal Control of Partially Observable Markov Processes.* Thèse de Doctorat, Stanford University, 1971.

[Steels, 1989] L. Steels. Diagnosis with a function-fault model. *Applied Artificial Intelligence*, 3 :213–237, 1989.

[Stone et Veloso, 2000] P. Stone et M. Veloso. Multiagent systems : A survey from a machine learning perspective. *Autonomous Robotics*, 8(3), July 2000.

[Sutton et Barto, 1998] R.S. Sutton et A.G. Barto. *Reinforcement Learning, An introduction.* BradFord Book. The MIT Press, 1998.

[Venner, 2002] Samuel Venner. *un titre.* Thèse de Doctorat, Université Henri Poincaré, Nancy 1, 2002.

[Watkins, 1989] C. J. C. H. Watkins. *Learning from Delayed Rewards.* Thèse de Doctorat, King's College, Cambridge, 1989.

[Williams, 1992] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8 :229–256, 1992.

[Wolisz et Tschammer, 1993] A. Wolisz et V. Tschammer. Performance aspects of trading in open distributed systems. *Computer Communications*, 16 :277 – 287, May 1993.

[Wolpert et Tumer, 1999] D. Wolpert et K. Tumer. An introduction to collective intelligence. Rapport Technique NASA-ARC-IC-99-63, NASA AMES Research Center, 1999.

[Wooldridge et Jennings, 1995] M. Wooldridge et N. R. Jennings. Intelligent agents : Theory and practice. *Knowledge Engineering Review*, 10(2), 1995.

[Wooldridge, 2002] Michael Wooldridge. *An Introduction to Multiagent Systems.* John Wiley & Sons, February 2002.

Résumé

Le sujet de cette thèse s'inscrit dans le domaine de l'Intelligence Artificielle Distribuée (IAD). Il s'agit de résoudre des problèmes distribués à l'aide d'agents en interaction évoluant dans un environnement complexe (dynamique et stochastique). Les données, les connaissances et le contrôle nécessaires à la résolution d'un problème sont distribués parmi l'ensemble des agents. Dans les systèmes multi-agents, la résolution d'un problème est alors la résultante des interactions entre les agents.

Dans ce contexte général, notre intérêt se porte sur les systèmes multi-agents coopératifs. Les agents ont, à l'exécution, un comportement réactif (absence de communication) et sont soumis aux conséquences probabilistes de leurs actions (incertitude). Les systèmes multi-agents possédant ces caractéristiques réactives sont souvent conçus de manière empirique et ascendante. L'approche ascendante consiste à concevoir le système multi-agents, puis à en adapter les paramètres afin de satisfaire le fonctionnement recherché. A cette approche ascendante, nous opposons l'approche descendante : connaissant les caractéristiques d'un problème, comment concevoir le système multi-agents qui saura le résoudre ?

La théorie de la décision introduit des méthodes de planification réactive qui, en plus d'être adaptées à la prise de décision dans l'incertain, nous apportent un formalisme théorique. Coordonner des agents à travers l'élaboration de plans individuels, c'est résoudre un problème de type DEC-POMDP dont la complexité est NEXP-complet dans le cas le plus favorable. Afin de contourner cette difficulté, nous proposons un ensemble d'algorithmes de conception d'agents utilisant les Processus Décisionnels de Markov (MDP) et fondé sur deux propriétés fondamentales que nous prêtons à nos agents : la *subjectivité* et l'*empathie*. La subjectivité prend en compte la localité des perceptions et des actions, tandis que l'empathie est ici utilisée pour coordonner les actions de nos agents par planification réactive.

Mots-clés: Systèmes Multi-agents, Processus décisionnels de Markov, Planification