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

Proposition d'un formalisme pour la construction automatique d'interactions dans les systèmes multi-agents réactifs

THÈSE

présentée et soutenue publiquement le 18 Novembre 2005

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Vincent THOMAS

Composition du jury

Président :	Didier GALMICHE, Professeur, UHP Nancy 1
Rapporteurs :	Joël QUINQUETON, Professeur, Université Paul Valery, Montpellier Philippe MATHIEU, Professeur, Université de Lille
Examinateurs :	Olivier SIGAUD, Professeur, Paris 6 Vincent CHEVRIER, Maître de conférences, UHP Nancy 1 (directeur de thèse) Christine BOURJOT, Maître de conférences, Université Nancy 2 (co-directrice de thèse)

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

Mis en page avec la classe thloria.

Remerciements

Les années d'une thèse constituent une période d'événements, parfois de doutes, de rencontres, d'échanges. Le travail qui est présenté dans ce manuscrit est la synthèse de nombreuses idées qui ont des origines diverses et variées mais aussi la conséquence d'échanges et de relations plus personnelles. Ce travail n'aurait ainsi pu arriver à terme sans l'aide de nombreuses personnes.

Dans un premier temps, mes remerciements et ma gratitude vont à mes deux directeurs de thèse Christine Bourjot et Vincent Chevrier. Je ne les remercierai jamais assez de m'avoir soutenu et accompagné au cours de la thèse tout en m'ayant laissé une liberté d'explorer mes propres pistes. Leurs encouragements m'ont permis d'avancer et les longues discussions que l'on a pu avoir ensemble m'ont permis d'appréhender le domaine extrêmement vaste que sont les systèmes multi-agents. Cette thèse ne serait pas ce qu'elle est sans leur soutien, leur confiance et leur aide scientifique.

Dans un second temps, mes remerciements vont à Francois Charpillet qui m'a accueilli il y a désormais 5 ans, au sein de l'équipe MAIA dont il est le responsable. Il m'a fait découvrir un domaine de recherche passionnant et m'a accordé sa confiance à plusieurs reprises. Ses qualités humaines ont su donner à l'équipe une ambiance de travail chaleureuse et m'ont offert un cadre privilégié que je ne suis pas prêt d'oublier.

Je suis trés honoré que Philippe Mathieu et Joël Quinqueton aient accepté de rapporter cette thèse et aient pris le temps de lire ce manuscrit et d'évaluer ce travail. Mes remerciements les plus sincères sont aussi adressés à Olivier Sigaud et à Didier Galmiche pour avoir suivi mes travaux et accepté de faire parti du jury.

Je tiens en outre à remercier particulièrement Olivier Sigaud, Olivier Simonin, Hervé Frezza-Buet et Remi Coulom qui ont constitué l'assemblé de mon comité de thèse. Ce comité se déroulait à une étape charnière de la thèse et les remarques et les réflexions que la soutenance a pu susciter ont influencé ce mémoire et la manière dont j'ai pu aborder mes travaux par la suite.

Je n'oublie pas Didier Desor et Henri Schroder du laboratoire de neurosciences comportementales de l'université Henri Poincaré Nancy 1. Les discussions animées que l'on a pu avoir sur les expériences qu'ils ont pu mener et sur l'éthologie de manière plus générale ont sans nul doute guidé les idées présentes dans ce document.

Enfin, j'ai bien entendu une pensée particulière pour toutes les personnes que j'ai pu croiser au cours des années pendant lesquelles j'ai conduit la thèse. Ces personnes constituent un environnement propice aux discussions scientifiques (et moins scientifiques) qui font de la thèse une expérience humaine et culturelle des plus enrichissantes. Mes remerciements se dirigent naturellement vers l'équipe MAIA mais aussi vers les autres équipes (PAROLE, ISA, CORTEX,...) : Alain D., Regis L., Simon L., Sylvain C., Rodolphe C., Iadine C., Laurent J., Franck G. qui au cours de mes déménagements successifs dans les bâtiments ont du partager leur bureau avec mes jeux de mots ; Anne B., Eric L., Loic P., Bruno S., Amine B., Julien T., Daniel S., Raghav A., Jamal S., Cedric R., Fabrice L., Renato D., Romaric C., Rédimé H., Monique S., Francois K qui ont su rendre les réunions d'équipe des plus agréables ; Celine S., Nadine B. et Martine K. pour leur disponibilité et leur grande aptitude à résoudre les aspects administratifs liés à la thèse ; David L., Vincent B., Vincent C., Yann B., Bernard G., Jean-Christophe U., Nicolas R., Blaise

P.,... . Je tiens à remercier plus particulièrement Alain et Romaric qui ont pris le temps de lire intégralement un manuscrit encore inachevé pour me faire part de leurs remarques, tous ceux qui en ont lu des parties et toutes les personnes (et elles sont nombreuses) qui m'ont soutenu, m'ont aidé dans les moments difficiles et ont participé à l'élaboration des idées présentes dans ce document.

Enfin, merci à toute ma famille et mes amis proches pour leur soutien indéfectible.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction générale

Cette thèse s'inscrit dans le domaine de l'intelligence artificielle (IA) qui s'est donné comme objectif (parmi d'autres) la compréhension de mécanismes d'adaptation d'entités perçues comme intelligentes et l'utilisation de ces mécanismes pour la résolution de problèmes. La problématique abordée dans ce manuscrit concerne plus particulièrement une branche de l'Intelligence Artificielle : l'Intelligence Artificielle Distribuée (IAD).

L'objectif que nous nous sommes fixés consiste à concevoir et construire de manière automatique un collectif d'agents, plongés dans un environnement, dotés de perceptions partielles et de connaissances incomplètes, qui de par leurs interactions vont évoluer ensemble jusqu'à converger vers un état stable pour construire une solution à des problèmes complexes (cf [Jea97]).

L'aspect distribué des systèmes multi-agents que nous souhaitons construire apparaît à plusieurs niveaux

- au niveau de l'**exécution** : chaque agent est amené à prendre des décisions en fonction des connaissances limitées à sa disposition
- au niveau de la construction et de l'**adaptation** des comportements des entités en interaction : chaque agent est amené à remettre en cause et à modifier ses règles comportementales à partir des connaissances limitées à sa disposition.

Le problème général de conception que nous souhaitons aborder peut alors se décomposer en deux sous-problèmes :

- le premier problème consiste à déterminer comment représenter le système et son exécution décentralisée.
- le second consiste à déterminer comment construire et adapter de manière distribuée les comportements des entités en interaction

Les formalismes mathématiques constituent un outil pour répondre en partie à ces problématiques. D'une part, ils permettent de représenter un système, son exécution ainsi que les problèmes à résoudre à partir d'une syntaxe constituée d'éléments non ambigus pour faire émerger une structure générique à un problème distribué. D'autre part, ils constituent une première étape pour proposer des algorithmes génériques manipulant ces éléments syntaxiques pour construire une réponse au problème formalisé. Formalisme et algorithmes sont bien entendus liés puisque les algorithmes se basent sur l'abstraction constituée par le formalisme pour construire des solutions. Ainsi, proposer un formalisme (limité par son expressivité) constitue déjà un début de réponse au problème de conception de collectifs d'agents.

Le domaine de la théorie de la décision s'intéresse à la construction automatique de comportements dans des environnements incertains. Il propose des formalismes pour représenter des agents solitaires devant résoudre un problème représenté sous la forme d'un problème d'optimisation. Il propose en outre des algorithmes permettant de construire automatiquement son comportement.

Des avancées récentes dans ce domaine, comme le formalisme DEC-POMDP, se sont intéressées à la description de systèmes dont l'exécution est distribuée et au problème posé par la construction des comportements des agents. Cependant, comme ce cadre formel ne préconise rien quant aux méthodes de résolution à mettre en œuvre, il n'a pas cherché à donner aux agents des éléments leur permettant d'appréhender, à la construction, la présence d'autres agents dont le comportement peut évoluer. Le problème de construction des comportements représenté dans un DEC-POMDP est en outre extrêmement complexe (NEXP) et rend impossible en pratique la recherche de la solution optimale dans le cas général.

Or, on observe tous les jours, des organisations complexes humaines et animales qui parviennent à construire de manière autonome et distribuée des réponses à des problèmes posés à la collectivité. Ces organisations constituent des structures collectives créées par les individus, à partir des interactions qu'ils peuvent entretenir entre eux et qui influencent en retour leurs comportements. Le postulat qui nous a conduit à cette thèse a été de spécifier **au niveau individuel** certaines des interactions possibles entre agents pour permettre aux agents de considérer la présence d'autres agents.

Ainsi, plutôt que de chercher à résoudre mathématiquement un problème extrêmement complexe posé dans le formalisme DEC-POMDP, nous avons préféré nous intéresser à des formalismes plus adaptés à la résolution décentralisée et à partir desquels il peut être plus facile de construire des solutions adaptatives. Au cours de ce manuscrit, nous proposons ainsi un formalisme original, inspiré des DEC-POMDPs mais plus proche de nos préoccupations. Ce formalisme est caractérisé par des prises de décision collectives locales et intègre explicitement au niveau individuel une instance d'un concept d'interaction. A partir de ce formalisme, nous proposons en outre des techniques inspirées de l'apprentissage par renforcement mono-agent pour effectuer des apprentissages multi-agents entièrement décentralisés permettant de générer des organisation liées à la tâche à résoudre. Ces algorithmes ont pour objectif non pas de construire des solutions optimales mais de montrer qu'il est possible de construire plus facilement des comportements collectifs dans ce nouveau formalisme que dans les DEC-POMDPs.

Dans ce chapitre, nous présentons de manière plus détaillée notre approche de la conception de système multi-agents.

1.1 Contexte scientifique

Cette thèse se situe à la confluence de deux champs de recherche : les systèmes multi-agents et la théorie de la décision.

1.1.1 Intelligence artificielle et Système Multi-agents

Les systèmes multi-agents constituent une sous-branche de l'intelligence artificielle (IA). L'intelligence artificielle s'est donnée pour objectifs :

- la résolution de problèmes complexes semblant requérir les capacités possédées par les êtres humains
- la compréhension des mécanismes et des processus impliqués dans les comportements intelligents humains et animaux
- l'émulation de comportements intelligents.

Nous nous intéressons plus particulièrement au premier objectif consistant à construire des systèmes multi-agents adaptatifs permettant de résoudre des problèmes de manière automatique.

Afin d'expliciter cette problématique, nous aborderons succinctement dans les parties suivantes ce que nous entendons par système intelligent et par système multi-agents.

1.1.1.1 Système Intelligent

L'intelligence est une notion difficile à cerner. Derrière ce terme se cachent de nombreux acceptions différentes. Un système pourra être qualifié d'intelligent parce que, pour un observateur extérieur, il semblera doté de capacités cognitives habituellement attribuées à l'homme (intelligence émulée) ou parce qu'il cherchera à reproduire les mécanismes par lesquels l'homme ou l'animal prend des décisions complexes (intelligence simulée).

Russel et Norvig dans [RN95] fournissent une taxonomie des systèmes "intelligents" selon deux axes de classification :

- Le premier axe traite de l'objet d'étude de l'intelligence. Russel et Norvig distinguent les systèmes pour lesquels l'intelligence réside dans les raisonnements et les manipulations de représentations internes et les systèmes pour lesquels l'intelligence réside dans le comportement du système et les interactions que ce dernier entretient avec son environnement.
- Le second axe caractérise la façon dont la notion d'intelligence sera évaluée. L'intelligence d'un système pourra être évaluée comme la conformité du comportement/raisonnement artificiel par rapport à des comportements/raisonnements humains ou de manière objective par rapport à des critères numériques de performance (on parlera alors de système rationnel).

Cette taxonomie répartit les systèmes intelligents en quatre grandes classes de systèmes :

- les systèmes qui pensent comme un humain
- les systèmes qui pensent de manière rationnelle
- les systèmes qui agissent comme un humain
- les systèmes qui agissent de manière rationnelle

Les travaux effectués au cours de cette thèse s'intéressent à la dernière classe de systèmes : les systèmes qui agissent de manière rationnelle. Ils cherchent à construire des systèmes distribués qualifiés d'intelligents du fait des actions qu'ils émettent et des interactions qu'ils entretiennent avec l'environnement dans lequel ils sont plongés et ce par rapport à un critère numérique. L'unité de base de tels systèmes est l'agent qui, comme son nom l'indique, est caractérisé par sa capacité à exercer des actions sur son environnement.

1.1.1.2 de l'IA aux systèmes multi-agents réactifs

L'IA a tout d'abord été anthropomorphique et s'est initialement inspirée de la métaphore du penseur solitaire : les chercheurs dans ce domaine ont cherché à produire des programmes isolés en émulant les processus cognitifs humains pour résoudre des problèmes complexes.

Cette approche va cependant subir trois bouleversements qui vont voir apparaître de nouveaux courants :

Dans un premier temps, le courant de **'l'intelligence incarnée'** va remettre en cause l'intelligence artificielle comme la manipulation de symboles et de représentations de connaissances pour s'intéresser à la problématique de l'action et de l'interaction avec un monde extérieur. Ce courant s'est intéressé aux systèmes qui agissent plutôt qu'aux systèmes qui raisonnent, la manipulation de représentations ne constituant qu'un aspect de cette nouvelle problématique.

Dans un second temps, l'apparition de l' **intelligence artificielle distribuée** (IAD) a remis en question l'étude des systèmes constitués d'un agent. Weiss [Wei99] présente l'intelligence artificielle distribuée de la manière suivante : *"l'Intelligence Artificielle Distribuée est l'étude, la construction et l'application des systèmes multi-agents, c'est-à-dire des systèmes dans lesquels des agents intelligents en interaction poursuivent un ensemble de buts ou effectuent une certaine tâche."*. La métaphore du penseur solitaire a été remise en question et s'est accompagnée d'une nouvelle problématique : celle de l'interaction entre plusieurs entités. Une nouvelle question se pose alors : comment un agent peut-il prendre en considération la présence d'autres acteurs dans le système pour interagir au mieux avec eux ?

Dans un troisième temps, l'utilisation des processus de prise de décision complexes basés sur des représentations symboliques a été remise en cause par le développement **d'agents réactifs** basés sur des règles simples de type stimulus-réponse [Bro91]. Cette approche inspirée en partie par l'éthologie, revendique la possibilité de s'affranchir de processus cognitifs élaborés pour résoudre un problème complexe. L'enchaînement de règles comportementales extrêmement simples peut conduire à l'apparition d'un comportement complexe pouvant être qualifié d'intelligent par un observateur extérieur.

L'approche réactive a trouvé ses échos en IAD et pose la problématique de l'apparition de comportements collectifs et des relations entre les comportements individuels simples et locaux des agents, leur agencement à l'exécution et la tâche collective à résoudre.

Nos travaux se placent dans une vision de l'intelligence qui est incarnée, distribuée et réactive.

1.1.2 La théorie de la décision

La théorie de la décision quant à elle est une discipline mathématique qui s'intéresse à la notion d'incertitude et de préférence. Elle se fonde sur les axiomes de probabilité et d'utilité.

Alors que le langage des probabilités fournit un cadre de travail pour décrire un monde, ses lois d'évolution et les croyances incomplètes sur ce monde, la théorie de l'utilité fournit un cadre pour rendre cohérents des ensembles de préférences. La théorie de la décision qui en est le produit constitue un cadre permettant d'évaluer des prises de décisions, consistant en une allocation irrévocables de ressources sous contrôle, effectuées en fonction d'informations incomplètes et d'un modèle de préférence (cf [HBH88]).

Ainsi, elle propose des éléments syntaxiques permettant de formaliser des problèmes de prises de décision en environnement incertain et des outils permettant de calculer les réponses à ces problèmes sous la forme de fonctions comportementales.

Certains modèles qui en sont issus permettent de représenter des systèmes multi-agents et intègrent la notion d'incertitude inhérente à ces systèmes pour lesquels :
- chaque agent ne dispose que d'une perception partielle de son environnement
- chaque agent ne connaît pas parfaitement les comportements des autres agents qu'il côtoie

En formalisant le problème de prise de décision multi-agents, ils permettent de poser la problématique de la construction des comportements des entités mises en présence, de fournir des algorithmes pour apporter des solutions à cette problématique. Ils constituent, de fait, un cadre intéressant pour construire une théorie de l'intelligence collective fondée sur la notion de rationalité comme cela a pu être fait pour des agents autonomes solitaires (cf [MAI02]).

1.2 Conception de systèmes intelligents

1.2.1 Problématique de conception

Maintenant que nous avons introduit les domaines dans lesquels nous nous situons, il est possible de définir la problématique de conception de systèmes intelligents multi-agents ainsi que les difficultés qui y sont associées.

1.2.1.1 Propriétés des systèmes à concevoir

Les systèmes multi-agents sont caractérisés par une **exécution décentralisée** : chaque agent décide individuellement des actions qu'il souhaite entreprendre à partir de ses perceptions mais tous les agents contribuent globalement à l'évolution du système. Chacun des agents est doté de capteurs pour percevoir le monde dans lequel il est plongé et d'effecteurs pour tenter de le modifier. Les capacités d'un agent sont cependant limitées : un agent ne peut percevoir intégralement l'environnement dans lequel il est plongé et les actions qu'il peut émettre ne peuvent modifier l'environnement que localement.

De plus, le système est plongé dans un environnement **incertain et initialement inconnu**. Il doit donc pouvoir faire preuve de **capacités d'adaptation** pour fournir une réponse adéquate et tirer parti de ses expériences passées. Comme l'agent constitue l'unité d'exécution du système et que les communications restent limitées, nous souhaitons que l'adaptation du système soit faite au niveau de l'agent en fonction de ses connaissances limitées pour pouvoir s'adapter à toutes sortes de situations sans avoir besoin de considérer le système dans sa globalité.

1.2.1.2 Définition de la problématique de conception

Concevoir un système multi-agents intelligent, tel que nous l'avons décrit, consiste alors à répondre aux questions formulées par Ferber dans [Fer97] :
- **Q1** - *Quelle est l'architecture de l'agent, sachant que son comportement dépend de cette architecture ?* Répondre à cette question consiste à déterminer les capteurs, les effecteurs de l'agent et ses capacités de raisonnement lui permettant d'évoluer dans son environnement.
- **Q2** - *Quelles sont les formes d'interactions permettant à plusieurs agents de maximiser leur satisfaction* (dans notre cas la fonction caractéristique du problème) ? Répondre à cette question consiste à trouver des mécanismes et des processus permettant d'assurer une cohérence à la collectivité dans le but d'exprimer un comportement global intelligent.

- **Q3** - *Comment faire évoluer les comportements des agents pour qu'ils puissent tirer parti des expériences passées et quelles en sont les conséquences sur le comportement collectif ?* Répondre à cette question consiste à construire et à adapter les règles permettant à un agent de choisir les actions ou les interactions qu'il souhaite utiliser pour faire avancer la résolution de la tâche collective tout en considérant la présence d'autres acteurs dans le système.
- **Q4** - *Comment implémenter et réaliser de tels systèmes ?* Répondre à cette question consiste à définir un langage et des protocoles permettant de mettre en œuvre ces systèmes dans des conditions réelles.

1.2.1.3 Difficultés associées à la conception

Répondre aux questions posées par Ferber est extrêmement complexe du fait de la présence de deux niveaux de granularité. Cette problématique est exprimée par le concept d'émergence (cf [Jea97]). Le calcul émergent est défini par Forrest comme : *"un ensemble d'entités en interactions : le processus ; un épiphénomène produit par ce processus : un état stable, un invariant ou une trace d'exécution ; une interprétation de cet épiphénomène comme un calcul ou le résultat d'un calcul"*. Cette définition correspond aux systèmes que l'on souhaite construire : les agents en interaction sont définis à un niveau local et l'agencement de leurs comportements produit une trace d'exécution. Cette trace d'exécution correspond au comportements des agents, constitue une réponse au problème à résoudre et s'exprime au niveau global du système auquel les agents n'ont pas accès.

De plus, l'intérêt des systèmes multi-agents consiste à tirer parti de phénomènes émergents pour lesquels le tout est plus que la somme des parties. Or, le concept d'émergence s'oppose à celui de réduction qui consiste à considérer qu'un phénomène peut toujours être expliqué par des processus sous-jacents (cf [Jea97]). Cette remarque fait qu'il est en général déjà très difficile de prédire le comportement global d'un système à partir des règles locales. La conception d'un système multi-agents, encore plus complexe, se heurte à la non-décomposabilité du problème à résoudre du fait de l'interaction entre les différents constituants du système.

Cette remarque est d'autant plus vraie que les systèmes que l'on cherche à construire ont deux niveaux de complexité distincts : ils doivent pouvoir émettre des comportements collectifs complexes à partir de l'interaction de composants aux comportements simples.

1.2.2 Décomposition du problème de conception

Afin de mieux cerner le problème de conception de systèmes multi-agents intelligents, nous proposons de le séparer en deux sous-problèmes (cf figure 1.1).

Le premier sous-problème consiste à décrire les possibilités offertes aux agents et correspond aux deux premières questions posées par Ferber. Répondre à ce problème consiste à décrire et à représenter les capteurs et les effecteurs d'un agent, les processus et les structures grâce auxquels les agents parviennent à prendre une décision ainsi que les formes d'interactions que les agents peuvent exercer entre eux. Ce premier sous-problème peut lui aussi se décomposer en deux parties : la première consiste à proposer des classes de possibilités aux agents (cf figure 1.1 1a), la seconde consiste à instancier ces classes pour choisir des effecteurs, des senseurs et des formes

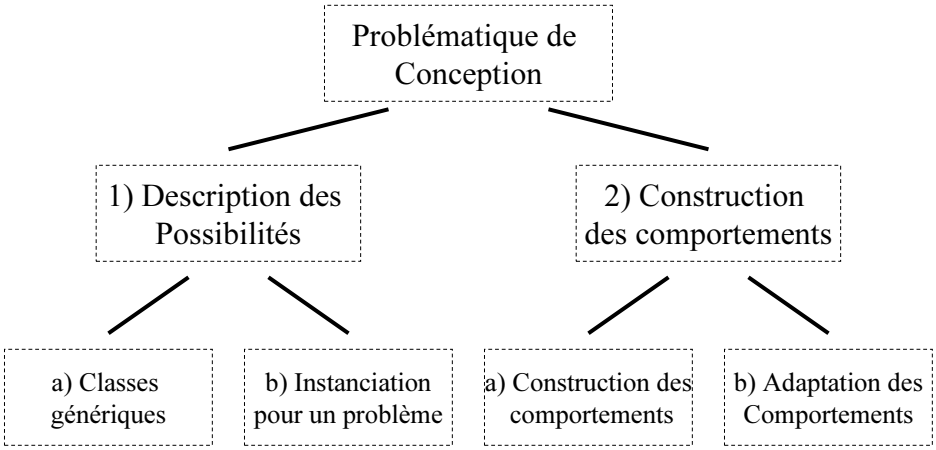


FIG. 1.1 – Décomposition de la problématique de conception

d'interaction adaptés à la résolution d'un problème particulier (cf figure 1.1 1b).

Le second sous-problème est le problème de construction des comportements et correspond à la troisième question posée par Ferber. Il s'agit de déterminer les fonctions de décision des agents (cf figure 1.1 2a) ainsi que des processus d'adaptation (cf figure 1.1 2b) afin que les agents puissent tirer parti au mieux des possibilités qui leur sont offertes.

De manière évidente, ces deux sous-problèmes sont liés puisque les comportements des agents se fondent sur les possibilités qui leur sont offertes et que les possibilités offertes aux agents conditionnent la performance du système.

1.3 Notre approche de la conception

Nous souhaitons au cours de cette thèse proposer des techniques permettant de concevoir automatiquement des systèmes multi-agents intelligents capables de s'adapter pour résoudre un problème posé sous la forme d'une fonction de performance à optimiser.

1.3.1 Contraintes fixées

Les constructions que nous envisageons sont guidées par plusieurs souhaits :
- Nous voulons que ces approches soient suffisamment génériques pour pouvoir traiter plusieurs problèmes grâce aux mêmes outils.
- Nous voulons réduire le rôle du concepteur dans le processus de construction du système et nous concentrer sur des constructions automatiques.
- Nous souhaitons des approches décentralisées permettant au système de s'adapter à l'exécution sans représentation globale de l'environnement

Ces considérations nous ont orientés vers l'utilisation de cadre formel pour exprimer un problème global, caractériser les comportements des agents et les manipuler pour produire une réponse collective à ce problème.

1.3.2 Proposition d'un formalisme

Notre proposition consiste à utiliser des cadres formels. L'objectif d'un cadre formel est d'exprimer un problème sous la forme d'une abstraction pouvant être manipulée par la suite. Proposer un cadre formel consiste à répondre en partie au sous-problème de la description des possibilités offertes aux agents (problème 1a de la figure 1.1). Il s'agit de proposer un certain nombre d'éléments syntaxiques permettant de décrire et de représenter les constituants d'un système multi-agent. Ces éléments doivent pouvoir représenter les agents mais aussi les possibilités qui leur sont offertes ainsi que les formes d'interaction possibles.

Une fois qu'un cadre formel a été proposé, il est possible de répondre de manière générique à la seconde problématique consistant à proposer des processus de construction génériques des comportements des agents (problème 2a et 2b de la figure 1.1). Une réponse à cette problématique est un ensemble d'algorithmes permettant de manipuler les éléments du cadre formel pour construire les comportements des agents dans ce contexte.

La proposition d'un formalisme conditionne cependant les types de système que l'on va pouvoir représenter et les approches de construction possibles. En effet, le cadre formel présente l'avantage de simplifier les système multi-agents en dégageant une structure. Mais cette structure limite en retour les systèmes qu'il est possible d'envisager.

1.3.3 De l'agent vers l'agent social

De plus, puisque nous souhaitons des systèmes capables de s'adapter au niveau de l'agent, il est nécessaire que l'agent puisse considérer la présence d'autres agents dans le système.

Ceci a deux conséquences :
- d'une part **les règles de construction et d'adaptation** des comportements des agents doivent s'exprimer au niveau local tout en prenant en compte la présence d'autres agents dans le système
- d'autre part, **la formalisation** du système multi-agents doit pouvoir fournir à l'agent des éléments lui permettant d'appréhender son environnement mais aussi la présence d'autres agents.

Comme nous le verrons par la suite, il existe déjà des cadres formels permettant de représenter des agents devant résoudre un problème dans un environnement incertain. Des algorithmes existent dans ce cadre et parviennent à répondre exactement à la problématique de construction et d'adaptation des comportements d'un agent isolé.

La question à laquelle nous avons cherché à répondre était de savoir s'il était possible d'adapter ces processus de construction de comportements individuels pour construire des agents sociaux, c'est-à-dire, capables de prendre en compte la présence d'autres agents afin de construire un comportement collectif.

Cette question a été posée par Wooldridge qui l'énonce comme constituée de deux sous-problématiques :
- La problématique de conception d'un agent isolé (nommée "agent designing" par Wooldridge [Woo01]) consistant à construire des agents capables d'action autonome pour ac-

complir avec succès les tâches qui lui sont confiées.
- La problématique de conception de système multi-agents et des interactions (nommée "society designing" par Wooldridge [Woo01]) consistant à construire des agents capables d'interagir avec d'autres agents afin d'accomplir avec succès les tâches qui leur sont confiées particulièrement lorsque les autres agents ne partagent pas forcément les mêmes intérêts

Ces deux problématiques se répondent l'une l'autre et ne peuvent pas être facilement décorrélées :
- Les comportements des agents doivent prendre en considération la présence d'autres agents dans le système avec lesquels ils peuvent interagir
- Les interactions entre agents doivent être fondées sur les capacités individuelles des agents et leurs comportements.

1.4 Organisation du manuscrit

Ce manuscrit s'organise ainsi autour de la question du passage de l'agent isolé à l'agent social.

Dans le chapitre 2, cette question sera abordée sous la facette des **concepts**. Nous décrirons le point de vue de l'agent isolé et le point de vue des systèmes multi-agents. Cette partie aura pour objectif de présenter les principales différences en terme de concept : la notion d'agent isolé telle que nous l'abordons se fonde sur la théorie de la décision et sur la notion de rationalité sur laquelle se fonde les Processus de Décision Markoviens. La notion de système multi-agents met l'accent sur l'interaction permettant le passage des comportements individuels à un comportement collectif. La question qui se posera à l'issue de cette partie sera de savoir s'il est possible de trouver un point de jonction entre ces deux conceptions consistant à introduire une rationalité dans les systèmes multi-agents et à introduire de l'interaction et un aspect social à la rationalité.

Dans le chapitre 3, la question du passage de l'agent isolé à l'agent social sera abordée sous la facette des cadres formels que sont les **processus de décision markoviens** . Nous présenterons ainsi les MDPs (Markov Decision Process) et leurs extensions qui permettent de représenter des systèmes distribués. Ces cadres formels issus de la théorie de la décision ont pour objectif de représenter un système, sa dynamique et les moyens disponibles pour influer sur cette dynamique. Ils permettent de formaliser le problème de construction des comportements d'agents. Un certain nombre d'algorithmes permettent de construire des comportements d'agents isolés rationnels et constitue une brique de base pour notre approche. Nous mettrons en évidence l'absence de la représentation explicite de l'interaction pourtant fondamentale dans ces systèmes. Cette absence de la représentation d'interactions locales induit des aspects centralisés dans les approches de résolution qui se heurtent à nos principes de localité.

Dans le chapitre 4, la question du passage de l'agent isolé à l'agent social sera abordée selon le point de vue **des systèmes biologiques**. La question que nous nous sommes alors posée a été de voir comment des systèmes naturels parviennent à intégrer une composante sociale pour produire des comportements collectifs complexes issus de mécanismes d'adaptation individuels. Nous décrirons dans cette partie le modèle Hamelin qui simule un phénomène de spécialisation collective observé dans des groupes animaux. Le modèle Hamelin parvient à construire des comportements collectifs complexe à partir de mécanismes d'adaptation individuels et d'interactions directes locales entre les agents. Ces interactions directes parviennent à expliquer comment les

agents parviennent à intégrer la présence d'autres agents sans nécessiter de représentation complexe du système global et du comportement des autres.

Comme nous souhaitons de manière analogue tirer parti d'algorithmes de construction de comportements individuels pour produire des comportements collectifs complexes, nous avons souhaité intégrer cette instanciation du concept d'interaction directe dans un cadre markovien.

Dans le chapitre 5, nous proposons un cadre formel inspiré des Processus de Décision Markovien et intégrant une formalisation de l'interaction directe présente dans le modèle Hamelin. Dans ce cadre formel, les agents ont la possibilité d'agir mais aussi d'interagir localement entre eux. Cette possibilité d'interaction directe est représentée explicitement dans le système et constitue un élément de premier ordre du formalisme original interac-DEC-POMDP. L'interaction directe permet de définir de nouvelles entités constitués par les agents en interaction et permet d'envisager l'intégration d'une composante sociale au sein des agents puisque ceux-ci peuvent désormais savoir avec quels autres agents ils peuvent être localement en interaction.

Afin de mettre en œuvre le formalisme proposé et tirer parti de la notion d'interaction, nous nous concentrons sur une sous-classe de problèmes pour laquelle l'interaction directe constitue les seules influences possibles entre agents. Nous proposons une approche fondée sur des apprentissages décentralisés et des échanges locaux d'information pour construire une réponse collective au problème posé. Le problème sur lequel nous nous concentrerons et la proposition de processus de construction seront décrits dans le chapitre 6.

Cette partie présentera les résultats quantitatifs et qualitatifs que nous avons obtenus grâce à ce processus de construction entièrement décentralisé. Nous nous concentrerons plus particulièrement sur la spécificité de l'interaction et l'aspect décentralisé des constructions.

Enfin, la dernière partie propose un résumé du travail effectué et présente les perspectives ouvertes par nos travaux.

Chapitre 2

De l'agent aux systèmes multi-agents

Dans ce chapitre, nous décrirons les systèmes que nous souhaitons construire **au niveau conceptuel**. Ce chapitre a plusieurs objectifs.

Tout d'abord, la notion d'agent et la notion de système multi-agents sont vastes. Ce chapitre présente les concepts sous-jacents à ces systèmes. Le premier objectif de ce chapitre va consister à préciser les entités que nous souhaiterons formaliser dans les parties suivantes ainsi que leurs propriétés.

Nous présentons ainsi l'agent et les systèmes multi-agents dans deux sous-parties.

La première partie traite de l'agent et sera liée aux cadres formels que nous présenterons par la suite. Elle mettra ainsi l'accent sur le concept de rationalité qui trouve ses échos dans les formalismes de processus de décision markoviens que nous aborderons dans le chapitre suivant.

La seconde partie traite des systèmes multi-agents. Elle présente ce domaine de recherche et mettra l'accent sur la notion d'interaction fondamentale dans ces systèmes. Cette notion d'interaction nécessitera d'introduire de nouvelles compétences à l'agent présenté précédemment.

Chacune de ces parties sera décomposée en 4 sous-parties : une définition, les propriétés attendues sur l'agent ou le système (au vu du problème de conception de systèmes intelligents que nous abordons), les caractéristiques de l'agent ou du système multi-agents et enfin la manière dont la problématique de conception s'exprime dans ce cadre.

Comme nous souhaitons construire des systèmes multi-agents intelligents à partir de processus de construction de comportement d'agent isolé, nous ferons une synthèse des concepts associés à l'agent centrés autour de la notion de rationalité et des concepts associés aux systèmes multi-agents centrés autour de la notion d'interaction. Le second objectif de ce chapitre est de répondre à la question : existe-t-il un point de jonction permettant d'aborder notre problématique de conception ?

2.1 Agent intelligent isolé

Cette partie se concentre sur l'agent et a pour objectif d'aborder la problématique de la conception d'un agent rationnel. Elle permettra de mettre en évidence par la suite la spécificité des systèmes constitués de plusieurs agents.

Dans cette partie, nous présenterons tout d'abord la notion d'agent, nous montrerons ensuite comment les attentes des systèmes que l'on cherche à construire s'expriment en terme de propriétés classiquement associées au concept d'agent et quels sont les constituants d'un agent.

2.1.1 Définition de la notion d'agent

Il n'existe pas de définition universelle de ce que peut être un agent. La définition de Russel et Norvig reste très générale et constitue une bonne entrée en matière. Selon Russel et Norvig ([RN95])

Définition : "Un agent est tout ce qui peut être compris comme percevant son environnement à travers des senseurs et comme agissant sur cet environnement par l'intermédiaire d'effecteurs"

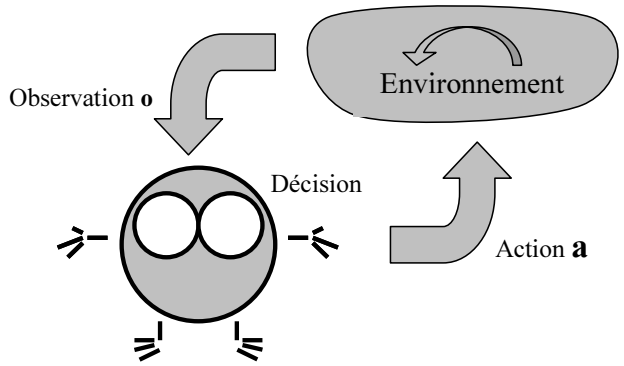


FIG. 2.1 – Boucle sensori-motrice

Un agent est donc défini comme une entité située dans un environnement, dotée de capteurs et d'actionneurs, observant l'environnement et cherchant à modifier son évolution. De manière plus précise, le fonctionnement général d'un agent est décrit par une boucle fermée appelée boucle sensori-motrice (ou perception-action) illustrée par la figure 2.1.

L'exécution de cette boucle peut se décomposer en plusieurs étapes :
- Initialement, l'agent se trouve dans une certaine configuration interne.
- Il perçoit une partie de l'environnement dans lequel il est plongé grâce à ses capteurs.
- Il choisit une action à entreprendre en fonction de sa configuration interne et de ses perceptions. Ce choix sera considéré comme le résultat d'une fonction de prise de décision (nous nous intéresserons aux différentes formes que peut prendre cette fonction par la suite)
- Il effectue cette action dans le but de modifier l'état de l'environnement ou/et sa configuration interne.

– Il reçoit de nouvelles perceptions et le processus se répète.

Pour analyser le comportement d'un agent (c'est-à-dire ses réactions aux diverses configurations possibles du monde) et pouvoir parler d'agents intelligents tels que nous voulons les construire[1], nous devons faire appel à une autre notion : la notion d'agent rationnel.

2.1.2 Propriétés attendues d'un agent

2.1.2.1 Rationalité

La rationalité constitue la manière dont la notion d'intelligence que nous avons choisie peut s'exprimer au niveau de l'agent. Une première caractérisation relativement naïve stipule qu'un agent rationnel est un agent qui "effectue les bonnes actions au bon moment" [RN95].

Afin de raffiner cette caractérisation, Russel et Norvig préconisent d'utiliser une **mesure de performance** pour caractériser l'objectif de l'agent et évaluer les actions qu'il a pu choisir. Cette mesure de performance constitue une représentation du problème à résoudre. Elle est indépendante de l'agent et nécessite d'être définie par un observateur extérieur au système qui peut analyser dans quelle mesure le système parvient à répondre au problème posé. Une fois une mesure de performance établie, c'est à l'agent rationnel d'exprimer un comportement apte à maximiser cette mesure dans le long terme.

Cependant, comme nous nous intéressons à des agents dotés de perceptions partielles, nous souhaitons répondre à des problèmes pour lequel un agent n'a pas forcément accès à l'ensemble de l'information lui permettant de choisir au mieux son action. S'il y a deux couloirs, l'un menant à un trésor l'autre à un danger, et que l'agent n'a aucun moyen de distinguer ces couloirs, on ne peut lui refuser le statut d'agent rationnel sous prétexte qu'il a choisi d'emprunter le mauvais couloir. Le caractère rationnel d'un agent doit donc être évalué par rapport à la quantité d'informations à laquelle l'agent peut avoir accès.

Qualifier une décision de rationnelle sera donc fonction de [RN95] :
– la mesure de performance définissant l'objectif de l'agent
– l'historique des perceptions de l'agent
– les connaissances qu'a l'agent de l'environnement
– les actions que l'agent peut effectuer

Ainsi, dans certaines situations, le comportement rationnel d'un agent peut être d'acquérir de l'information sur son environnement afin d'améliorer ses performances individuelles à long terme.

Afin de voir comment le concept de rationalité peut s'exprimer dans le cadre de perceptions partielles, prenons l'exemple du 'tiger problem' dont on peut trouver une formalisation dans [CKL94]. Un agent se trouve dans une pièce contenant deux portes. Derrière l'une d'elle se trouve un tigre, derrière l'autre, un trésor (cf figure 2.2). L'objectif de l'agent consiste à choisir la porte menant au trésor le plus rapidement possible tout en évitant le tigre. Une mesure de performance possible consiste à attribuer une note positive inversement proportionnelle au temps mis par l'agent pour accéder au trésor et une note négative lorsque l'agent rencontre le tigre. On suppose en outre que la configuration initiale des salles a été déterminée de manière aléatoire et

[1] c'est-à-dire dont l'intelligence est évaluée par rapport aux relations que l'agent entretient avec son environnement

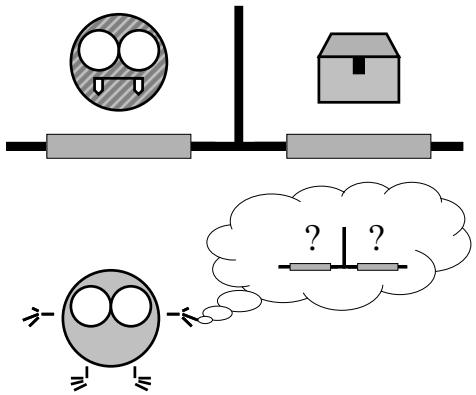


FIG. 2.2 – "Tiger Problem"

qu'après chaque tentative de l'agent, le tigre et le trésor sont de nouveau placés aléatoirement derrière les portes. L'agent a la possibilité d'écouter à l'une des portes avant d'en ouvrir une mais cela lui prend un peu de temps. De plus, il n'est pas sûr de percevoir la présence du tigre de manière systématique, même si celui-ci se trouve derrière la porte.

Si les deux salles sont insonorisées et que l'agent dispose de cette information, ouvrir n'importe quelle porte constitue une action rationnelle puisque l'agent a conscience qu'il ne peut avoir accès à aucune information. Si, par contre, les salles ne sont pas insonorisées, écouter aux portes peut constituer une action rationnelle : en fonction des bruits qu'il aura entendus, l'agent peut choisir au mieux son action future. Dans certaines conditions (lorsque les capteurs de l'agent sont défectueux par exemple), il peut même être amené à écouter à plusieurs reprises les bruits provenant de derrière l'une ou les deux portes. Enfin, si les salles sont insonorisées et que l'agent ne le sait pas, écouter aux portes peut constituer une décision rationnelle. Ceci peut lui permettre de savoir à la suite de mauvaises expériences qu'écouter aux portes n'apporte pas d'information de bonne qualité. Il décidera donc par la suite de ne plus perdre de temps à essayer de détecter la présence du tigre et pourra augmenter son critère de performance à long terme.

Pour prendre en compte ces différents facteurs, Russel et Norvig définissent la notion d'agent rationnel "idéal" qui correspond au concept de rationalité limitée :

Définition : "Pour toute suite de percepts, un agent rationnel idéal effectue l'action dont il est espéré qu'elle maximise la mesure de performance, et ce, sur la base de la preuve fournie par la suite de percepts et par les connaissances intégrées de l'agent"

Cette propriété pourra s'exprimer dans les cadres formels issus de la théorie de la décision que nous aborderons par la suite (cf partie 3) et permettra de guider le processus de construction des comportements d'agents isolés.

2.1.2.2 Autonomie

Les capacités d'adaptation dont nous souhaitons doter nos agents s'expriment quant à elles selon le concept d'autonomie de Russel et Norvig.

Dans [RN95], Russel et Norvig définissent la notion d'autonomie par rapport à la quantité d'informations dont dispose initialement un agent : si le comportement d'un agent est déterminé a priori, Russel et Norvig affirment que l'agent manque alors d'autonomie.

Un système autonome est défini par (cf [RN95]) :

Définition : "Un système est autonome dans le sens ou son comportement est déterminé par ses propres expériences".

Cette définition de l'autonomie est fortement liée à la notion d'apprentissage et d'adaptation. Un agent autonome selon cette définition est un agent qui parvient à adapter son comportement en utilisant ses expériences passées. Un tel agent possède alors de grandes capacités d'adaptation à des environnement inconnus. Comme dans les problèmes qui nous intéressent, un agent n'a pas accès à l'ensemble de l'information utile et ne connaît pas forcément la configuration globale du système, nous nous concentrerons sur des agents autonomes selon cette définition.

Cette notion trouvera elle aussi des échos dans la partie suivante traitant des formalismes de prises de décision markovien et sera associée à la notion d'apprentissage (cf partie 3.2.4).

2.1.2.3 Point de vue de Jennings

Jennings, Sycara et Wooldridge [JSW98] proposent une autre définition de la notion d'agent qui reste néanmoins proche de celle de Russel et norvig.

Définition : "Un agent est un système informatique, situé dans un environnement, qui est capable d'action autonome et flexible dans le but de répondre à ses objectifs de conception" .

Cette définition est fondée sur les trois mots clefs suivants :
- **situé** signifie que l'agent reçoit des données sensorielles directement à partir de l'environnement et peut effecteur des actions qui visent à le modifier.
- **autonome** signifie que l'agent peut agir de lui-même sans intervention directe extérieure.
- **flexible** est lié à la notion d'objectif et d'intelligence. Un agent est flexible s'il possède les trois caractéristiques suivantes :
  - il répond à temps : son temps de réponse est court par rapport au temps d'évolution de l'environnement
  - il est pro-actif : il n'agit pas uniquement en réponse à l'environnement mais peut prendre l'initiative lorsque cela est approprié pour atteindre son objectif
  - il est social : lorsqu'il est mis en présence d'autres agents, il est capable d'interagir pour résoudre ses problèmes et d'aider les autres agents. La notion de capacité sociale sera abordée de manière plus détaillée lorsque l'on traitera les systèmes multi-agents.

Bien que cette définition d'autonomie corresponde aussi à une des attentes de notre système, elle s'avère être plus faible que celle de Russel et Norvig. Nous adopterons donc la première mais

garderons à l'esprit cette seconde définition qui stipule que toute action exercée sur le système est à l'initiative de l'agent ce qui aura des conséquences lorsque nous décrirons cette notion d'autonomie dans les SMAs.

2.1.3 Caractéristiques d'un agent

La partie précédente s'est attardée sur la définition d'un agent rationnel : un agent est qualifié de rationnel si son comportement maximise à long terme une mesure de performance objective. Cette définition ne spécifie rien quant aux capacités de l'agent ni quant à ses processus de prise de décision.

Proposer un agent rationnel consiste alors à décrire (en plus de la fonction de mesure de performance individuelle liée au problème) :
- une architecture externe spécifiant les moyens dont dispose l'agent pour agir et percevoir son environnement
- une architecture interne spécifiant les représentations internes de l'agent et ses mécanismes de prises de décision
- des fonctions de décision basées sur cette architecture interne et éventuellement des processus d'adaptation de ces fonctions.

afin que l'agent puisse agir rationnellement au sens défini précédemment.

Les prochaines parties vont présenter ces différentes facettes de l'agent afin de pouvoir exprimer plus précisément la problématique de conception d'un agent isolé.

2.1.3.1 Architecture externe et relation au monde

Un agent a été défini comme un système en interaction avec son environnement. Il a donc besoin de capteurs pour percevoir celui-ci et d'effecteurs pour y exercer des influences.

Principe de localité Dans les problèmes que nous souhaitons aborder par la suite, chaque agent est situé et est limité dans ses capacités d'action et de perception. Ces contraintes sont réunies sous la notion de **principe de localité**. Un agent vérifiant ce principe ne perçoit et ne peut modifier qu'une partie de l'environnement contenue dans un certain voisinage. Cette notion de voisinage est fortement dépendante du problème et est souvent définie de manière implicite.

Caractéristiques de l'environnement Afin de préciser les systèmes que l'on cherche à construire, nous allons spécifier les caractéristiques des environnements sur lesquels nous nous concentrerons par la suite.

Russel et Norvig classent les environnements en plusieurs catégories. Nous précisons avant tout qu'il s'agit plus d'une caractérisation de la relation entre un agent et son environnement que des propriétés de l'environnement lui même ce qui explique que ces caractéristiques soient associées à l'architecture externe de l'agent. Selon [RN95], un environnement peut être :
- accessible ou inaccessible : Si l'appareil sensoriel de l'agent lui permet de percevoir à chaque instant l'état complet de l'environnement, celui-ci sera qualifié d'*accessible*.

- déterministe ou non-déterministe : Si l'état de l'environnement futur peut être prédit avec exactitude à partir de l'état actuel et de l'action effectuée par l'agent, l'environnement sera qualifié de *déterministe*. Par contre, si l'environnement est inaccessible, l'agent ne dispose peut être pas d'assez d'information sur l'état réel du monde pour en déduire l'état suivant. Russel et Norvig préconisent de penser l'aspect déterministe d'un environnement "à partir du point de vue de l'agent".
- statique ou dynamique : L'environnement est qualifié de *statique* s'il ne subit pas de modification durant le temps nécessaire à un agent pour prendre sa décision.
- discret ou continu : L'environnement est qualifié de *discret* si le nombre d'états le caractérisant est dénombrable.

Afin de spécifier ultérieurement des cadres formels, nous avons dû effectuer des choix concernant les classes d'environnement que nous considérerons. Nous nous concentrerons sur des environnements discrets plus simples à représenter (à la base des formalismes présentés dans la partie 3). Nous supposerons, au vu du principe de localité, que l'environnement est inaccessible et non-déterministe. Nous supposerons que la prise de décision de l'agent est rapide et donc qu'il fait face à des environnements qualifiés de statiques selon Russel et Norvig (même si l'environnement possède une dynamique du fait de sa relation avec l'agent).

2.1.3.2 Architecture interne

Le terme d'architecture interne désigne l'ensemble des structures de données et des processus internes à un agent lui permettant de prendre une décision (éventuellement rationnelle) consistant à choisir une action en vue de modifier l'environnement.

Différents types d'agent Bien que la frontière soit relativement floue, on distingue deux types d'agents en fonction de leur architecture interne (cf [WJ95]) :

- les agents cognitifs
- les agents réactifs

Un **agent cognitif** est un agent disposant de capacités de raisonnement développées. Il est caractérisé par :

- la représentation explicite de ses objectifs
- une représentation évoluée de l'environnement
- une capacité à manipuler ces représentations pour anticiper ou réévaluer ces objectifs.

Les architectures BDI (Belief Desire Intention) (cf [RG95]) constituent un type d'architecture d'agents cognitifs. Cette architecture est basée sur les notions d'attitudes mentales que sont la Croyance (Belief), le Désir (Desire) et l'Intention (Intention) :

- les croyances correspondent aux informations (éventuellement incomplètes et incorrectes) qu'a l'agent de son environnement
- les désirs correspondent aux états de l'environnement que l'agent souhaiterait voir réalisés
- les intentions correspondent aux projets de l'agent pour satisfaire ses désirs.

Les prises de décisions d'un agent BDI sont alors effectuées à partir de la manipulation des états mentaux de l'agent et de la révision de ceux-ci au cours du temps.

Brooks dans [Bro91] s'oppose à l'utilisation d'agents cognitifs pour construire des systèmes intelligents. Selon lui, concevoir des agents intelligents cognitifs se heurte au problème de la représentation explicite de l'environnement au sein de l'agent. Il propose plutôt d'utiliser directement

les perceptions de l'agent pour la prise de décision et de construire des agents intelligents de manière incrémentale en commençant par des agents aux comportements très simples.

Dans [Bro91], il présente une architecture constituée de plusieurs couches de contrôle pour produire un comportement pouvant être qualifié d'intelligent sans qu'à aucun moment, une représentation explicite de l'environnement ne soit nécessaire. Il reprend ainsi la remarque de Simon (citée dans [Bro91]) selon laquelle *"la complexité du comportement d'un système ne réside peut être pas dans la complexité de l'agent mais dans celle de l'environnement et de leurs interactions"*.

Un **agent réactif** est un agent issu de ces considérations. Il est régi par des règles stimulus-réponses et ne dispose pas de représentation interne explicite de son environnement. Les prises de décision d'un tel agent peuvent être représentées par une machine à états finis et son comportement apparaît du fait du chaînage des différentes règles comportementales et de leurs conséquences sur l'environnement (cf figure 2.3).

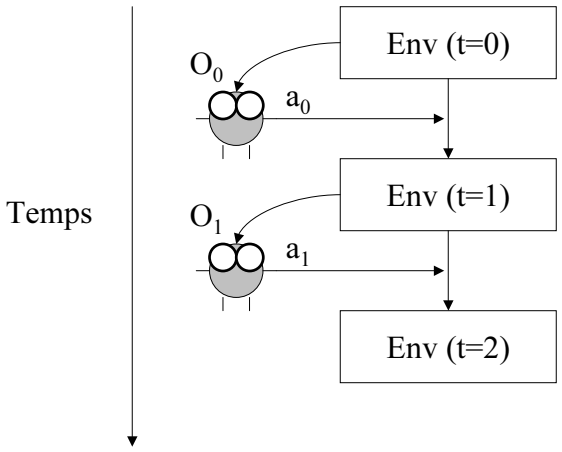


FIG. 2.3 – Chaînage de règles comportementales

Considérons un agent petit poucet (inspiré des travaux de Drogoul [DF93a], lui même inspiré des comportements de fourmis) qui évolue dans un monde discret à deux dimensions. Cet agent possède deux comportements : un comportement d'exploration consistant à partir de la maison et à se déplacer dans l'environnement en posant des cailloux et un comportement de retour consistant à retourner à la maison en suivant le chemin constitué par les cailloux qu'il a préalablement déposés. La conjonction de ces deux règles très simples permet à un agent de prendre un chemin qu'il a emprunté sans qu'il n'ait besoin de mémoriser ce chemin et d'en avoir une représentation interne. L'environnement fait alors office de mémoire pour l'agent et permet à celui-ci d'émettre des comportements pouvant être qualifiés de complexes à partir de règles comportementales élémentaires.

Comme nous l'avons précisé en introduction, notre objectif est de construire de manière automatique des systèmes multi-agents intelligents. Pour atteindre cet objectif, nous avons souhaité nous intéresser à la construction d'agents réactifs pour deux raisons :

- une raison **idéologique** consistant à voir dans quelle mesure des agents simples peuvent parvenir à s'adapter pour répondre à des problèmes posés à la collectivité : Comme Brooks

l'a montré, de tels agents répondent à un certain nombre de problèmes, peuvent fournir des réponses complexes malgré la simplicité de leur architecture comportementale et constituent selon lui la première étape pour la construction d'artefacts intelligents.
- une raison **technique** : les comportements des agents réactifs sont facilement représentables et manipulables dans des cadres mathématiques.

Mémoire d'un agent Les agents que nous souhaitons concevoir doivent être autonomes selon la définition de Russel et Norvig, c'est-à-dire que leur comportement doit être déterminé et modifié par leurs expériences. Pour pouvoir garder une trace de leur expérience et adapter son comportement en conséquence, un agent a donc besoin de mémoire.

Pour un agent **réactif**, on distinguera deux types de mémoire :
- *la mémoire à court terme* qui est destinée à stocker des événements précis afin de modifier les décisions de l'agent en fonction d'une (ou plusieurs) observation(s) passée(s). Ce type de mémoire intervient directement comme entrée des règles comportementales Stimulus-Réponse de l'agent. Une question reste cependant posée : celle consistant à déterminer quel élément il peut être utile de mémoriser.
- *la mémoire à long terme* qui a pour objectif d'adapter le comportement de l'agent. Cette mémoire est constituée d'une synthèse de l'ensemble de l'expérience de l'agent. Ce type de mémoire n'est pas une entrée directe des règles comportementales mais un moyen de remettre en cause les règles stimulus-réponse internes à l'agent et d'en produire éventuellement de nouvelles. La question posée dans ce cadre consiste à trouver comment faire la synthèse des expériences passées, comment la stocker et comment adapter le comportement de l'agent à partir de cette synthèse.

Pour les mêmes raisons que précédemment, comme nous souhaitons construire de manière automatique des systèmes multi-agents, nous avons du faire des choix sur l'architecture interne des agents que nous considérerons par la suite. Comme nous souhaitons disposer des agents les plus simples, nous avons décidé de nous concentrer sur des agents sans mémoire à court terme. En effet, l'utilisation de mémoire à court terme pose un autre problème souvent relégué au concepteur et auquel nous ne souhaitons pas répondre pour le moment : Quel élément l'agent doit-il mémoriser et comment le représenter ?

Une des questions que nous nous poserons dans la suite du document consistera donc à déterminer s'il existe des processus permettant à des agents d'effectuer une synthèse de leurs expériences pour mettre à jour leur comportement. Il existe de nombreuses réponses à cette problématique dans un cadre mono-agent comme les techniques d'apprentissage par renforcement que nous présenterons dans la partie 3.2.4.

2.1.4 Problématique de conception d'un agent rationnel

Maintenant que nous avons décrit les composants d'un agent, il est possible de décrire le problème de l' *"agent designing"* (comme l'appelle Wooldridge dans cf [Woo01]). Ce problème consiste à construire des agents capables d'agir de manière autonome et indépendante afin de résoudre la tâche qui leur a été confiée.

Dans le cadre de construction de système intelligent que nous nous sommes fixé, il s'agit pour une fonction de performance donnée, de fournir l'architecture interne et externe de l'agent, ainsi que les règles comportementales de type stimulus-réponse régissant son comportement pour

que l'exécution du système construise la solution au problème posé par agencement des règles comportementales de l'agent couplées à la dynamique de l'environnement.

2.1.5 Bilan sur le concept d'agent isolé

Dans ce chapitre, nous avons présenté la manière dont la problématique que nous abordons peut s'exprimer en terme de concepts pour des systèmes constitués d'un seul agent. La manière dont nous considérons les systèmes et le fait qu'un agent ne dispose pas de l'ensemble des informations du système à chaque instant nous conduit à chercher à construire des agents à **rationalité limitée** et **autonomes** au sens de Russel et Norvig c'est-à-dire, capables de s'adapter à partir de leurs expériences passées.

Comme dans la suite du document, nous avons cherché à construire de manière automatique des systèmes intelligents, nous avons dû faire des choix concernant les agents que nous construirons. Ces agents seront :

- régis par des règles de type stimulus-réponses pour pouvoir représenter simplement leur comportement et les construire de manière automatique
- sans mémoire à court terme pour ne pas avoir à modifier la structure des règles comportementales d'un agent mais dotés d'une mémoire à long terme pour permettre à l'agent de s'adapter
- respectueux des principes de localité puisque nous cherchons des contraintes réalistes
- dans un environnement discret, non accessible et non déterministe du point de vue de l'agent pour rendre compte des capacités limitées des agents que nous cherchons à construire.

2.2 Système multi-agents

Dans la partie précédente nous avons présenté le concept d'agent isolé comme une entité située dans un environnement qu'elle perçoit et sur lequel elle peut agir.

Cette partie se concentre sur des systèmes composés de plusieurs agents. Elle met l'accent sur le concept d'interaction, élément central pour ces systèmes puisqu'il représente ce qui permet de construire une réponse collective à partir de réponses individuelles. Une fois cette spécificité mise en avant, nous pourrons présenter la problématique de *"society designing"* (comme l'appelle Wooldridge dans [Woo01]).

Nous présenterons tout d'abord la définition d'un SMA, la manière dont les propriétés d'intelligence et d'autonomie peuvent s'exprimer dans ce cadre puis les différentes facettes d'un SMA selon l'approche voyelle.

2.2.1 Définition d'un système multi-agents

Selon Jennings et al. [JSW98], le terme système multi-agents correspondait à la définition donnée par Lesser [DLC89] comme étant *"un réseau faiblement couplé de solveurs de problèmes qui travaillent ensemble pour résoudre des problèmes qui sont au delà des capacités individuelles ou des connaissances de chaque solveur de problème."*. Toutefois, il a pris un sens plus général

pour désigner les types de systèmes constitués de composants autonomes (au sens de Jennings).

La principale caractéristique d'un SMA réside dans son exécution par essence décentralisée : chaque agent agit de manière autonome mais aucun ne contrôle totalement la dynamique du système. Le comportement collectif d'un tel système provient du couplage à l'exécution des différents comportements des entités mises en présence. Ce sont ces couplages entre les comportements des agents qui permettent de construire des réponses plus complexes que les comportements des agents mis en présence.

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

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

2.2.2 Propriétés attendues

Le fait d'avoir plusieurs entités dans le système nécessite de redéfinir les termes de rationalité et d'autonomie.

2.2.2.1 Rationalité

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

- Pour certains systèmes, chaque agent dispose d'une rationalité propre et d'une mesure de performance individuelle. Cette vision des choses est principalement compétitive : chaque agent cherche à maximiser son critère de performance éventuellement au détriment des autres agents du système.
- D'autres systèmes s'intéressent à une mesure de performance globale : le système est caractérisé par une fonction globale de performance (éventuellement calculée à partir d'une combinaison de mesures de performance locales), il n'existe pas de fonction de performance individuelle à proprement parler mais chaque agent a pour objectif d'émettre les actions permettant, en fonction des connaissances dont il dispose, de maximiser la performance globale du système. Cette rationalité est forcément limitée puisque l'agent ne connaît pas le monde ni surtout le comportement des autres agents. La rationalité d'un agent implique donc la présence de capacité sociale permettant d'estimer les comportements des autres agents pour agir au mieux par la suite.

Comme les comportements que nous construisons répondent à des problèmes collectifs, nous nous intéressons à cette deuxième classe de systèmes. Les systèmes seront intelligents, collectifs et coopératifs dans la mesure où une mesure de performance globale est définie et caractérise les performances du système.

2.2.2.2 Autonomie

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

- La définition de Russel et Norvig correspond à la capacité de l'agent à s'adapter à partir de ses expériences passées. Dans un système multi-agents, cette capacité inclut la capacité à s'adapter à l'environnement global du système mais aussi aux autres agents présents dont le comportement est inconnu et peut évoluer au cours du temps.
- Celle plus faible de Jennings, implique que chaque agent décide de manière autonome de son action sans intervention extérieure. Cette définition implique que toute action émise dans le système est uniquement à l'initiative d'un agent.

2.2.3 Caractéristiques d'un système multi-agents

2.2.3.1 Approche voyelle

Afin de présenter les différentes facettes d'un système multi-agents, nous allons considérer un système multi-agents selon l'approche 'voyelle' proposée par Demazeau [Dem97].

Selon cette approche, un système multi-agents est constitué par :

- les agents (A) autonomes, chacun étant doté de ses propres senseurs et effecteurs
- l'environnement (E) qui correspond à l'ensemble des éléments décrivant le monde dans lequel évolue les agents
- les interactions (I) qui correspond aux couplages qui peuvent s'exercer entre les agents du fait d'actions réciproques et à la manière dont ceux-ci s'exercent. Ce concept central dans les SMAs sera détaillé dans la partie 2.2.3.4
- l'organisation (O) qui peut être perçue comme la résultante globale des comportements des agents.

Dans les parties suivantes, nous déclinerons chacun de ces éléments selon notre point de vue.

2.2.3.2 Environnement et principe de localité

Les systèmes multi-agents sur lesquels nos travaux vont se concentrer sont, comme les agents isolés, régis par un principe de localité. Pour un agent d'un système multi-agents, ce principe de localité s'applique

- aux relations entre cet agent et l'environnement global
- aux relations entre cet agent et les autres agents présents dans le système qui constituent son environnement social

Comme les agents sont situés dans un environnement et dotés de capacités limitées, les relations entre un agent et son environnement social doivent respecter un certain nombre de contraintes de localités :

- Un agent n'a pas accès à l'ensemble des perceptions des autres agents.
- Un agent n'a pas accès directement aux comportements des autres agents ou, en d'autres termes, à leurs fonctions de prise de décision.
- Les agents ne perçoivent pas directement l'ensemble des agents du système.
- Les échanges d'informations sont limités par la topologie du système (pas de mémoire globale partagée)

Il est néanmoins possible de construire des systèmes multi-agents s'affranchissant de ces contraintes en permettant des communications globales entres tous les agents du système, mais, nous souhaitons néanmoins nous intéresser à des systèmes dotés de ces contraintes plus réalistes à notre goût[2].

De plus, un agent perçoit son environnement comme inaccessible et stochastique. Il s'agit d'une conséquence directe de la présence d'autres agents comme faisant partie intégrante de cet environnement. Les lois d'évolution du point de vue de l'agent correspondent au couplage entre les lois d'évolution de l'environnement global du système et les comportements des autres agents. Ainsi même si les lois du monde sont déterministes, les lois de l'environnement subjectifs d'un agent évoluent avec les comportements des autres agents et ne sont pas a priori prédictibles par l'agent.

2.2.3.3 Agent

Complément à l'agent La présence d'autres agents dans le système introduit de nouveaux problèmes par rapport à des agents isolés et doit se fonder sur de nouvelles capacités. Il devient nécessaire de proposer quelques légères modifications à la définition d'un agent pour en avoir une vision plus centrée multi-agents.

Ferber propose dans [Fer97] comme définition d'un agent "une entité informatique qui :

1. se trouve dans un système informatique ouvert comprenant un ensemble d'applications, de réseaux et de systèmes hétérogènes
2. peut communiquer avec d'autres agents
3. est mue par un ensemble d'objectifs propres
4. possède des ressources propres
5. ne dispose que d'une représentation partielle des autres agents
6. possède des compétences (services) qu'elle peut proposer aux autres agents
7. a un comportement tendant à satisfaire ses objectifs, en tenant compte d'une part des ressources et des compétences dont elle dispose et d'autre part de ses propres représentations et des communications qu'elle reçoit"

Cette définition met l'accent sur les capacités sociales de l'agent (comme cela pouvait déjà être le cas avec la définition de Wooldridge). C'est en effet grâce à ses capacités sociales que l'agent va pouvoir considérer la présence d'autres agents dans le système et pouvoir prendre en compte leur comportement. *Cet aspect, bien évidemment absent des systèmes constitués d'un seul agent, constitue une des facettes de la problématique de construction des systèmes multi-agents que nous aborderons dans ce manuscrit. Nous chercherons ainsi à construire des mécanismes permettant à un agent de considérer la présence d'autres agents dans le système au cours de ses prises de décision.*

Typologie des systèmes multi-agents De la même manière qu'il existait deux types principaux d'agents, on distingue les systèmes multi-agents en fonction du type d'agents qui les composent. On compte ainsi deux grandes classes de systèmes :

[2] D'autant plus, qu'un échange global d'information introduit une explosion combinatoire du nombre d'états et du nombre d'actions à considérer dans les processus de résolution comme cela apparaîtra dans les approches centralisées présentées dans la partie 3.

- les systèmes multi-agents cognitifs
- les systèmes multi-agents réactifs

Les **systèmes multi-agents cognitifs** sont constitués d'agents cognitifs généralement en faible nombre. Leur étude est fondée sur l'idée que la construction de systèmes multi-agents "intelligents" peut (voire doit) se faire à partir d'agents aux capacités de représentation complexes et pouvant disposer de processus de communication élaborés.

- Avantages : De tels systèmes peuvent mettre en oeuvre et tirer parti de mécanismes complexes de représentation des autres, de négociation et d'échanges d'information. Le concepteur peut alors adopter un point de vue anthropomorphique et s'inspirer de comportements humains pour guider la construction de système (comme par l'utilisation d'émotions par exemple cf [BP04]).
- Inconvénients : Cette approche se heurte au problème de la représentation des connaissances dèjà présent pour des agents isolés. En outre, la représentation complexe des autres agents du système limite le nombre d'agents et peut mener à des raisonnements coûteux. A cela, s'ajoute enfin la complexité des processus de communications entre agents.

Les **systèmes multi-agents réactifs** sont composés d'agents réactifs habituellement en grand nombre. De tels systèmes se basent sur l'hypothèse qu'il est possible de produire des comportements collectifs 'intelligents' complexes au regard de la simplicité des comportements individuels. Cette hypothèse reste proche du postulat de Brooks [Bro91] selon lequel la complexité d'un agent peut résulter d'un environnement complexe. Dans un système multi-agent, un comportement global complexe peut apparaître comme résultant des interactions nombreuses entre des composants simples en grande quantité.

De nombreux phénomènes naturels peuvent s'expliquer de cette manière. Les phénomènes de construction collective de nids chez les colonies d'insectes ne peuvent pas s'expliquer par la connaissance d'un plan de construction préétabli dans chaque individu. En effet, les insectes ne disposent pas des capacités cognitives nécessaires pour mémoriser un plan aussi complexe (cf [CFS+01]). Par contre, des systèmes multi-agents inspirés de ces phénomènes ont pu montrer que le chaînage de comportements individuels très simples peut parvenir à produire des structures complexes (cf [BDT99]). Ainsi, certains agents inspirés des comportements de fourmis déposent des phéromones sur le sol en fonction de la configuration locale de leur environnement. En déposant ces éléments, un agent modifie l'environnement perçu par les autres agents ce qui déclenchera d'autres règles comportementales de leur part. Dans certaines circonstances, la résultante globale de ces comportements constitue une trace chimique unique permettant à la colonie de construire des chemins entre son nid et les sources de nourriture avoisinantes (cf [BDT99] et [DMC96]).

- Avantages : les systèmes multi-agents réactifs présentent habituellement des intérêts en terme de fiabilité (assurée par le grand nombre d'agents du système et leur simplicité) et de passage à l'échelle (adaptation à l'augmentation du nombre d'agents) (cf [Par97])
- Inconvénients : Par contre, la création de tels systèmes doit faire face aux difficultés de prédiction du comportement global émergent non représenté explicitement dans le système et à la difficulté de contrôler les comportements individuels par rapport à un objectif non représenté explicitement. Ces difficultés proviennent du fait que la tâche à résoudre et les moyens pour résoudre cette tâche sont exprimés à deux niveaux de complexité distincts.

2.2.3.4 Interaction

La notion d'interaction constitue l'essence d'un système multi-agents puisque c'est grâce à elle que les agents vont pouvoir produire des comportements collectifs complexes et dépendants les uns des autres. Cependant, le terme d'interaction recouvre plusieurs réalités. Dans [Fer97], ce terme désigne les couplages entre agents mais désigne aussi la notion de situations d'interaction. Enfin, le terme interaction peut aussi correspondre aux manières selon lesquelles ces couplages entre entités s'effectuent au sein du système [KG03].

L'interaction comme couplage La notion d'interaction peut être comprise comme les influences à long terme que le comportement d'un agent peut avoir sur les prises de décision et la dynamique des autres agents. Cette vision de l'interaction que l'on peut trouver dans [Fer97] est définie comme "une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques". Ce sont ces interactions qui permettent de passer de comportements individuels indépendants à un comportement collectif complexe résultant des décisions et des actions émises par l'ensemble des agents du système.

Analyser un système selon ce point de vue permet d'isoler les ensemble d'éléments totalement indépendants. S'il n'y a aucune interaction en ce sens entre deux agents, leurs comportements n'ont aucun influence mutuelle et les deux agents peuvent évoluer de manière totalement indépendante.

Cette notion d'interaction par contre ne fournit aucune information concernant la nature des couplages entre les agents et les moyens employés pour mettre en relation les entités du système. Les parties suivantes se chargent de développer ces deux facettes de l'interaction.

L'interaction comme situation Dans [Fer97], Ferber définit ce qu'il entend par situation d'interaction :

Définition : "On appellera situation d'interaction un ensemble de comportements résultant du regroupement d'agents qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent et de leurs compétences individuelles."

Il fournit une classification des situations d'interaction selon plusieurs critères :
- la présence d'objectifs communs ou compatibles
- l'accès à des ressources communes
- la répartition des compétences au sein des agents.

En fonction de ces critères et de l'objectif du système, la notion de situation d'interaction peut s'exprimer comme les attitudes adoptées par les agents vis-à-vis des autres agents. Ferber distingue trois grandes catégories d'interactions :
- l'antagonisme entre agents : les agents ont des objectifs conflictuels (compétition) ou ont besoin de ressources communes (conflit sur les ressources).
- l'indifférence entre agents : les agents n'ont pas besoin des autres pour atteindre leurs objectifs et ne sont pas gênés par ceux-ci.
- la coopération entre agents : les agents doivent s'entraider pour atteindre leurs objectifs (éventuellement communs).

Cette notion est identique à celle de **situation interactive** décrite dans les travaux de Simonin [Sim01] : "La situation interactive d'un agent A par rapport à un agent B est la perception de l'incidence des actions de B sur la tâche en cours de A. Celle-ci appartient à l'une des trois catégories suivantes :
- incidence nulle
- incidence positive : l'agent perçoit l'interaction comme une aide
- incidence négative : l'agent perçoit l'interaction comme une gêne"

Comme les agents doivent résoudre ensemble un problème, nos travaux se concentrent sur des systèmes purement coopératifs. Ferber définit la coopération comme (cf [Fer95])

Définition : on dira que plusieurs agents coopèrent ou encore qu'ils sont dans une situation de coopération si l'une des conditions suivantes 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.**

Comme nous l'avons fait remarqué dans la partie 2.2.2.1, Cette définition implique la nécessité de se doter d'une fonction de performance de groupe qui, du fait des contraintes de localité que nous nous sommes fixées, n'est pas accessible à tout instant à l'ensemble des agents. Le problème qui se pose alors et qui sera abordé plus précisément dans la suite du document est de pouvoir faire un lien entre le comportement émis par un agent et cette fonction de performance globale qui caractérise l'objectif que l'on cherche à atteindre (la résolution de la tâche commune) mais qui n'est pas perceptible par l'agent. Ce schisme entre les deux niveaux de granularité individuel/collectif constituera une problématique récurrente de ce mémoire.

[Fer97] propose un certain nombre de méthodes permettant de mettre en oeuvre cette attitude coopérative entre agents comme
- la communication pour échanger des informations
- le regroupement physique des agents
- la spécialisation pour rendre certains agents plus adaptés à leur tâche
- la répartition des tâches, des informations et des ressources
- l'arbitrage et la négociation pour résoudre les conflits entre agents (ressources ou objectifs conflictuels) et réduire les désaccords entre individus.
- la coordination d'actions qui correspond à l'exécution de tâches supplémentaires permettant d'exécuter d'autres actions critiques dans les meilleures conditions

Ces différentes méthodes possèdent de nombreuses intersections et nous semblent pertinentes dans des architectures d'agents cognitifs car elles constituent alors des sources d'inspiration pour proposer des mécanismes permettant de résoudre une tâche.

Comme nous nous basons sur des agents réactifs pour lesquels ces notions ne peuvent être représentées explicitement au sein de l'agent, cette classification nécessite un niveau d'interprétation et sera plutôt vue comme une grille de lecture qu'un observateur extérieur pourra utiliser pour analyser le comportement du système multi-agents obtenu. Nous aborderons par exemple dans la partie 4 des systèmes présentant des phénomènes collectifs de spécialisation. Cependant, la spécialisation ne constitue qu'une conséquence de règles locales données a priori et la notion de spécialisation, une interprétation émise par un observateur extérieur de la manière par laquelle ces règles locales parviennent à répondre au problème.

L'interaction comme moyen Le dernier point de vue sur l'interaction consiste à considérer celle-ci comme le moyen permettant de mettre en relation dynamique plusieurs agents dans le système et comme la manière dont cette mise en relation est faite.

On distinguera plusieurs manières de réaliser ce couplage :
- par interaction indirecte non dirigée et médiée par l'environnement
- par interaction directe, dirigée, ponctuelle et instantanée

Les parties suivants explicitent ces moyens d'interaction.

Interaction indirecte

Une interaction indirecte est définie dans [KG03] comme "une interaction via des changements d'états persistants observables".

Une interaction sera qualifiée d'indirecte si :
- Elle est médiée par l'environnement, qui conserve une trace de cette interaction
- Elle n'est pas dirigée explicitement vers un autre agent

Ainsi, les agents participant à cette interaction sont les agents qui vont percevoir ces changements de l'environnement. L'identité de ces agents dépend donc de la dynamique du système et de leurs comportements [KG03]. Ainsi, un agent en effectuant une interaction indirecte, n'est pas certain de savoir avec quel(s) autre(s) agent(s) il est en train d'interagir puisqu'il ne sait pas quel agent sera amené à modifier son comportement en observant ces changement dans l'environnement. Cette dernière remarque explique dans quelle mesure l'évaluation locale d'une action émise par agent est difficile.

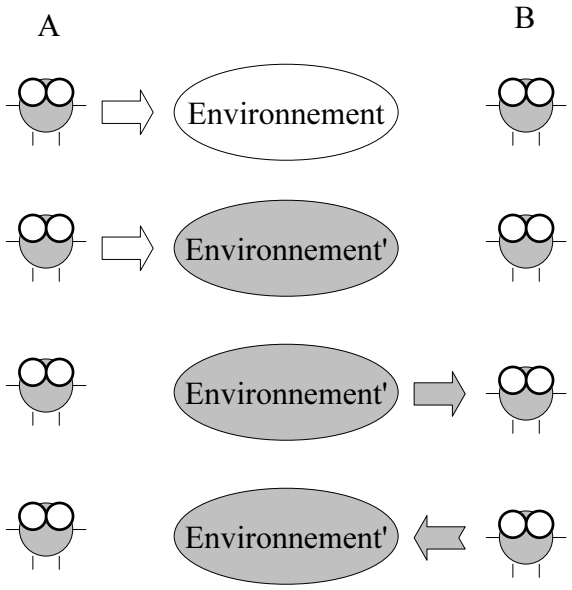


FIG. 2.4 – Interaction indirecte

Le schéma général d'une interaction indirecte est le suivant (cf figure 2.4) :
- un agent A émet une action
- cette action modifie l'environnement

– cette modification de l'environnement a pour conséquence une modification des perceptions des agents voisins comme cela peut être le cas pour B (éventuellement à partir d'un certain temps du fait de la localité des perceptions)
– B émet une action à partir de ses nouvelles perceptions

Il y a interaction (au sens de couplage entre les agents) puisque le comportement de l'agent B est fonction de l'action passée effectuée par l'agent A. Ce type d'interaction constitue le fondement du mécanisme de stigmergie [Gra59] observé dans des systèmes naturels selon lequel *'le travail guide l'ouvrier'*. Dans de tels systèmes, la tâche à résoudre est représentée au sein de l'environnement. En participant à la résolution de la tâche, les agents modifient l'environnement qui constitue une mémoire du groupe et permet de guider les agents qui s'y trouvent.

Les situations de conflit résolues par l'environnement constituent un cas particulier d'interaction indirecte. Par exemple, si deux agents décident simultanément de traverser un couloir et que ce couloir ne peut en laisser passer qu'un, les réactions de l'environnement aux tentatives des agents fournira une réponse adaptée (les deux agents se trouvent bloqués dans le couloir, un des agents a réussi à passer ou toute autre possibilité). Dans ce cas, la résolution de l'interaction est instantanée mais résulte des lois de l'environnement. Ce type d'interaction sera inclus dans notre notion d'interaction indirecte.

De nombreux systèmes multi-agents sont fondés sur la notion d'interaction indirecte (cf [DF93b], [BDT99], [BC01], [Rey87]). En effet, selon [KG03] l'interaction directe repose sur des 'communications anonymes' et permet de construire des systèmes ouverts qui ne souffrent pas de l'entrée ou de la sortie de plusieurs agents du système. Ces considérations sont aussi émises par certains éthologues au sujet des systèmes naturels (cf [CFS+01]).

Le modèle influence-réaction présenté dans la partie 2.2.4.1 permet de modéliser de manière générale ce type d'interaction indirecte.

Interaction directe

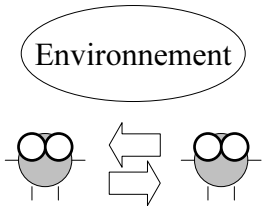


FIG. 2.5 – Interaction directe

Une interaction directe
– est dirigée explicitement vers un destinataire (un agent ou un groupe d'agents) dans le but de modifier son comportement (ou état interne) sans que l'état de l'environnement ne soit affecté directement
– elle est ponctuelle et ne se fait pas dans le temps même si les conséquences d'une interaction directe peuvent être à long terme.
– ne fait pas intervenir directement l'environnement.

L'interaction directe est fondée sur des envois de messages entre agents et des échanges d'information (cf [KG03]). Alors que les conséquences d'interaction indirecte sont décidées par les lois de l'environnement, les conséquences d'une interaction directe sont décidées par les lois comportementales des agents.

Parmi quelques utilisations d'interactions directes dans des systèmes multi-agents, on peut citer :
- l'envoi de messages permettant aux agents d'échanger des informations numériques sur leur environnement et l'avancement de la tâche (comme par exemple des notifications sur l'avancement de la tâche [Sze04] ou l'échange des fonctions de satisfaction des agents [SWMR99] présenté dans la partie 3.4.3.8)
- la négociation entre agents permettant à des agents de décider à plusieurs d'une répartition des tâches comme le réseau contractuel ([Smi88]) fondé sur des processus de négociation pour lesquels les agents envisagent leurs actions après négociation de contrats. Ces processus peuvent eux aussi prendre de nombreuses formes et certains travaux en proposent des modèles génériques (comme [MV03])

En fonction des types des agents impliqués, l'interaction directe peut aussi prendre de nombreuses formes :
- pour des agents cognitifs, elle peut s'exprimer à l'aide de langage et de protocoles de communication élaborés (comme KQML [FFMM94])
- pour des agents réactifs à l'aide d'échanges de signaux simples (comme dans le cas de l'éco-résolution [FJ91])

Comme nous nous focalisons sur des agents réactifs, les interactions directes que nous seront amenés à considérer seront fondées sur des échanges de signaux simples. Nous nous placerons en outre à un niveau d'abstraction tel que nous ne nous intéresserons pas aux problèmes de protocoles d'interaction.

Sur la notion d'interaction directe et indirecte

La différence entre une interaction directe et une interaction indirecte est avant tout une question de représentation. Tout échange d'information nécessitant un support, l'interaction directe dans un cadre réel s'accompagne forcément d'une perturbation de l'environnement. Cependant, si cette perturbation n'est pas représentée au sein des agents comme faisant partie de l'environnement et si elle est traitée indépendamment des lois environnementales, on considérera que cette interaction est directe.

2.2.3.5 Organisation

Il reste à définir la dernière composante d'un SMA : l'organisation. Il s'agit aussi d'un concept qui recouvre beaucoup d'acceptions et est utilisé dans de nombreux domaines (sociologie, éthologie).

Le mot organisation revêt deux acceptions (cf figure 2.6) :
- La première désigne la structure des relations entres les agents. Elle peut être comprise comme le résultat de l'analyse du système d'un point de vue extérieur. L'organisation est alors l'ensemble des corrélations qu'il est possible d'observer entre les comportements des composants du système à l'exécution. On parlera d'un système organisé lorsqu'il est possible d'exprimer le comportement global du système par des lois globales moins complexes que

l'ensemble des lois locales des agents(cf [Sha01]). L'organisation peut aussi être vue comme une entité à part entière qui exerce des influences et des contraintes sur les agents qui la supportent.
– La seconde désigne le processus dynamique à l'origine de cette structure. On préférera au terme d'organisation le terme plus spécifique d'auto-organisation, caractérisée par une apparition autonome d'organisation (en tant que structure) au sein du système ([CFS+01]). Cette autonomie est à prendre au sens de Jennings : l'organisation apparaît sans qu'aucune aide extérieure aux agents ne soit nécessaire.

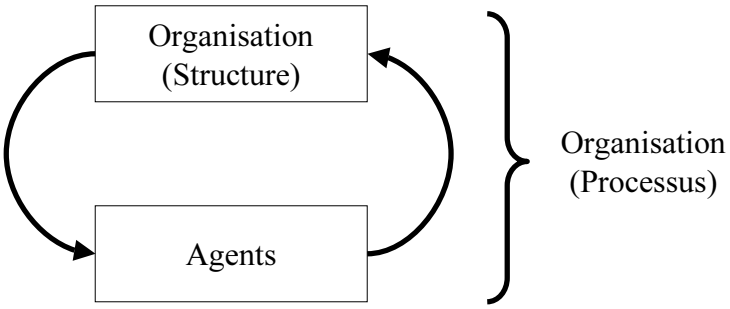


FIG. 2.6 – Organisation et processus d'organisation

Il est à noter que l'organisation est souvent considérée comme bénéfique pour le système : c'est elle qui permet à la communauté d'agencer les activités des agents dans le but de réaliser une tâche commune. Cependant, dans certains cas, la formation d'organisation et l'influence qu'elle peut avoir sur les agents peut s'opposer à la résolution du problème. Par exemple, la construction de chemins de phéromones persistants chez les fourmis peut empêcher celles-ci de trouver de la nourriture si la nourriture est déplacée rapidement d'un endroit à un autre. A peine un chemin apparaît-il que la quantité de nourriture censée se trouver à son extrémité se trouve à un autre endroit. (cf [Joh02])

Cette remarque pose la question de la construction d'une "bonne" organisation, c'est-à-dire une organisation utile à la tâche en cours par rapport au critère de performance associé au système. Ce processus de construction est lui aussi complexe puisque l'organisation est la conséquence des comportements locaux des agents alors qu'elle s'exprime au niveau collectif.

2.2.3.6 Bilan des caractéristiques d'un SMA

Dans cette partie, nous avons présenté les conséquences de la présence de plusieurs agents autonomes dans un système.

La multiplicité des prises de décision s'exprime sous la forme du concept d'interaction central dans les systèmes multi-agents car il concentre ce qui crée la relation entre les comportements individuels et le comportement collectif issu de ces comportements individuels.

Ce concept revêt trois significations. Chacune d'entre elles est liée à la présence de plusieurs agents dans le système :

- la situation d'interaction provenant du fait que les agents peuvent adopter des attitudes différentes les uns vis à vis des autres
- l'interaction à long terme qui représente les couplages possibles entre les comportements des agents du système.
- les moyens d'interaction, qui peuvent être classés en deux catégories : les interactions directes et les interactions indirectes.

Le concept d'interaction implique que les conséquences d'une action ou d'une interaction envisagée par un agent dépendent des autres agents du système et de leurs comportements. Ce concept implique en outre, qu'un agent doit considérer les autres agents du système s'il souhaite évaluer les conséquences à long terme des actions qu'il envisage.

2.2.4 Exemples de systèmes multi-agents

Afin d'illustrer les différents types d'interactions que nous avons décrits et la manière dont ceux-ci peuvent être utilisés dans des systèmes multi-agents, nous présentons plusieurs modèles génériques de systèmes multi-agents :

- le modèle influence-réaction
- le modèle d'eco-résolution
- le modèle satisfaction altruisme

2.2.4.1 Modèle influence-réaction

Le modèle influence-réaction est un modèle proposé par Ferber et Muller [FM96] et [Fer95] destiné à décrire le fonctionnement général d'un système multi-agent. Il n'émet aucune hypothèse quant à l'architecture interne des agents.

Ce modèle est basé sur la distinction entre les *influences* qui sont les tentatives mises en oeuvre par les agents pour modifier la dynamique du système et les *réactions* de l'environnement qui sont les réponses à ces influences. Ce modèle cherche à séparer "ce qui est produit par les agents (influence) de ce qui se produit effectivement (réaction)" résultant du couplage des influences. Cette distinction permet de rendre compte de l'émission simultanée d'influences. Les conflits possibles entre influences sont alors résolus par l'environnement qui centralise l'ensemble des influences émises.

Un système dynamique est défini dans ce modèle par :

- un ensemble d'états Σ
- un ensemble d'influences Γ
- un ensemble de lois élémentaires $laws : \Sigma \times \Gamma \rightarrow \Sigma$ qui décrivent les réactions élémentaires du monde aux influences.
- un ensemble d'opérateurs élémentaires générateur d'influences $op : \Sigma \rightarrow \Gamma$.
- une fonction $Exec : op^n \times \Sigma \rightarrow \Gamma$ chargée de concaténer les influences exercées sur l'environnement par les opérateurs utilisés.
- une fonction $React : laws^n \times \Gamma^n \times \Sigma \rightarrow \Sigma$ chargée de mettre à jour l'état du système à partir des lois du monde $laws$ et des influences exercés $\{\gamma_i\}, \gamma_i \in \Gamma$ sur l'environnement

Pour un SMA réactif, chaque agent est caractérisé par une fonction $Reflec_a : P_a \rightarrow \Gamma$ où P_a désigne l'ensemble des perceptions possibles de l'agent a. Un agent est alors défini par un tuple $< P_a, Perception_a, Reflex_a >$:

- P_a désigne l'ensemble des perceptions possibles de l'agent
- $Perception_a$ désigne la fonction de perception de l'agent, $perception_a : \Sigma \rightarrow P_a$
- $Reflex_a$ désigne la fonction de prise de décision de l'agent.

Exécuter un système se décompose en plusieurs étapes :
- le système est dans un état donné
- chaque agent calcule les influences qu'il émet
- l'état dynamique du système est modifié par l'ajout des influences produites par les agents
- le système met à jour son nouvel état

Les comportements des agents sont effectivement en interaction, les conséquences d'une influence émise par un agent peuvent dépendre des influences émises simultanément par les autres agents. Le modèle influence-réaction utilise systématiquement les lois de l'environnement pour régler les conflits entre agents et les couplages des comportements. Le modèle influence réaction permet de représenter des systèmes fondés sur des interactions indirectes.

2.2.4.2 L'éco-résolution

Le principe d'éco-résolution proposé par Ferber et Jacopin (cf [FJ91]) constitue une approche pour la résolution distribuée de problèmes d'optimisation. Il s'agit de construire une solution à un problème donné à l'aide d'une population d'agents.

Chaque agent dispose d'informations locales sur son environnement et cherche à atteindre un état de satisfaction qui lui est propre (plusieurs agents peuvent néanmoins chercher à atteindre le même état). Pour essayer de maximiser sa satisfaction, chaque agent dispose de comportements élémentaires. Lorsqu'un agent est gêné dans sa quête de satisfaction, il peut agresser les agents gêneurs qui se voient obligés de fuir et d'agresser éventuellement d'autres agents à leur tour. L'objectif de l'éco-résolution est de construire dynamiquement une solution globale à partir des maximisations locales effectués par les agents lors des agressions.

Dans [BDC98], la résolution des agressions utilise une fonction de force caractéristique de la satisfaction d'un groupe. Plus la satisfaction du groupe est importante, plus sa force pour conserver son état sera élevée.

Ce principe est fondé uniquement sur la notion d'interaction directe telle que nous l'avons présentée :
- la résolution des conflits ne fait pas intervenir la dynamique de l'environnement mais se fonde sur les caractéristiques comportementales des agents (caractérisées par leur fonction de satisfaction)
- enfin, même si elle peut avoir des conséquences à long terme, l'interaction est ponctuelle, aucune trace de l'interaction (en dehors de ses conséquences) n'est laissée dans l'environnement.

2.2.4.3 Modèle satisfaction altruisme

Dans [Sim01], Simonin présente un modèle fondé sur la notion de coopération directe et de coopération indirecte.

Ces deux notions sont similaires aux notions d'interactions directes et indirectes que nous présentons. Simonin définit la notion d'action coopérative comme "On appellera action coopérative d'un agent A envers un autre agent B toute interaction qui améliore qualitativement ou quantitativement la tâche de B ou leur tâche commune"

Le modèle satisfaction altruisme permet de représenter des agents capables d'agir de manière autonome pour participer localement à l'avancement de la tâche globale et d'interagir avec ses voisins en émettant des signaux d'attraction et de répulsion.

Simonin distingue alors la

- coopération indirecte qui est dûe aux actions individuelles émises par les agents faisant évoluer l'environnement. Dans l'exemple des robots fourrageurs, il s'agit de l'exploration de l'environnement qui introduit des gênes entre agents.
- la coopération directe résultante des signaux d'attraction et de répulsion émis par les agents.

Il propose un mécanisme pour intégrer au niveau des agents des actions altruistes en réponse aux signaux échangés entre les agents.

Le modèle satisfaction altruisme propose donc les deux catégories d'interaction énoncées précédemment :

- des interactions indirectes dues aux contraintes topologiques de l'environnement (robots situés qui sont amenés à se gêner dans leurs déplacements)
- des interactions directes issues d'échanges de signaux entre agents permettant de résoudre directement et localement les conflits.

2.2.4.4 Bilan sur ces exemples

Ces systèmes utilisent des interactions directes et / ou des interactions indirectes.

Si on souhaite disposer de tous les systèmes et tirer parti des différents types d'interactions, nous souhaitons disposer de systèmes capables de représenter ces différents types d'interactions :

- des interactions indirectes dûes à la présence d'un environnement commun dans lequel les agents évoluent
- des interactions directes permettant des influences mutuelles entre agents, dirigées et résolues localement par les agents impliqués

Ces systèmes devraient également disposer de processus permettant de manipuler ces deux types d'interactions dans un cadre commun pour résoudre les conflits dus à des agents rationnels dans un SMA.

2.2.5 La problématique de conception

On suppose disposer d'une fonction caractéristique du problème. On cherche alors à construire un système multi-agents réactif pour qu'à l'exécution, le chaînage du comportement des agents du fait des interactions parvienne à minimiser une fonction de coût.

Comme évoqué en introduction, concevoir un système multi-agents consiste donc à résoudre un certain nombre de problèmes (cf [Fer97]) :

- " Quelle est l'architecture de l'agent, sachant que le comportement de l'agent dépend de cette architecture ?
- Quelles sont les formes d'interactions permettant à plusieurs agents de maximiser leur satisfaction ? (dans notre cas la fonction caractéristique du problème)

- Comment faire évoluer les comportements des agents pour qu'ils puissent tirer parti des expériences passées et quelles en sont les conséquences sur le comportement collectif ?
- Comment implémenter et réaliser de tels systèmes ?"

De nombreux travaux, comme les travaux de Cost (cf [CCF+99]), D'inverno (cf [dKL98]), Dury (cf [Dur00]), Finin (cf [FFMM94]), Foisel (cf [Foi98]), Mathieu et Verrons (cf [MV03]) ou Ribeiro (cf [RD98]), se sont intéressés à la formalisation de l'interaction et aux protocoles d'interaction. Ces travaux en proposant des modèles génériques d'interaction et des schémas d'exécution se sont focalisés sur la description et l'opérationalisation de la notion d'interaction. Ils ont ainsi cherché à répondre aux deux premières questions de Ferber : quelle est l'architecture de l'agent et quelles sont les formes d'interaction. Cependant, ils se sont peu concentrés sur la question à laquelle nous cherchons à répondre dans cette thèse : à savoir comment construire de manière automatique les interactions entre les agents.

2.2.6 Bilan sur les systèmes multi-agents

Dans cette partie, nous avons présenté les systèmes multi-agents. Nous avons insisté sur la notion centrale d'interaction qui permet de faire le lien entre des comportements individuels et le comportement collectif global qui en est la conséquence. Le concept d'interaction pose alors la question de l'intégration de l'environnement social au sein de l'agent.

Nous avons aussi spécifié les différents aspects des SMA et présenté les caractéristiques sur lesquelles nous allons nous concentrer. Nous nous focaliserons par la suite sur des systèmes multi-agents

- réactifs
- coopératifs dans le sens que les performances du système sont évaluées de manière globale
- fondés sur un principe de localité
- dotés de moyen d'interaction directs et indirects

La construction de systèmes multi-agents pose le problème de la construction d'une société. Il s'agit d'un problème différent à l'agent designing, parce que cela nécessite

- d'avoir une fonction d'évaluation globale
- d'introduire des interactions
- d'ajouter un aspect social aux agents

C'est sur ce problème que nous allons nous concentrer.

2.3 Synthèse du chapitre

La première partie de ce chapitre s'est intéressée au concept d'agent. Elle a mis en évidence la notion de rationalité qui permet de spécifier ce qu'on entend par système intelligent et qui permet de formaliser plus précisément le problème de conception d'agent pour en tirer des algorithmes génériques.

La seconde partie de ce chapitre s'est focalisée sur les systèmes multi-agents que l'on cherche à construire. Elle s'est concentrée sur le concept d'interaction fondamental dans ces systèmes puisque qu'il s'agit du concept qui assure le lien entre des comportements individuels des agents et le comportement global qui en résulte.

La question qui se pose alors naturellement est de voir

- s'il est possible de doter les systèmes multi-agents d'une rationalité individuelle afin de pouvoir guider la construction des comportements des agents et disposer d'algorithmes génériques
- s'il est possible de doter les agents autonomes rationnels de socialité pour pouvoir construire de manière décentralisée le comportement d'un agent tout en prenant en compte la présence d'autres agents dans le système.

La problématique du passage de comportements individuels à des comportements collectifs s'exprime donc par la nécessité de doter les agents de compétences sociales pour

- comprendre ou analyser les situations d'interactions possibles entre agents
- évaluer la manière dont les comportements des agents sont couplés afin de prendre la meilleure décision individuelle
- considérer les moyens à sa disposition pour produire une réponse collective (interaction directe ou indirecte)

Pour mettre en œuvre cette idée, nous allons dans la partie suivante nous concentrer sur une catégorie de formalismes existant qui constituera notre point de départ : les processus de décision markoviens. Nous présenterons les problèmes rencontrés dans l'utilisation de tels formalismes et nous nous demanderons dans quelle mesure ces cadres formels fondés sur la notion de rationalité représentent l'interaction et la prennent en compte au niveau individuel.

Chapitre 3

Cadres formels et algorithmes issus des modèles markoviens

Les modèles markoviens issus de la théorie de la décision permettent de représenter des problèmes de prise de décision dans l'incertain. Bien qu'initialement ces formalismes s'intéressent à des problèmes n'impliquant qu'un seul agent, une extension récente, les DEC-POMDPs (cf [BGIZ02]), permet de représenter des problèmes de prise de décision multi-agents. Ces formalismes présentent plusieurs intérêts pour la construction de systèmes multi-agents :

- Ils intègrent explicitement la notion d'incertitude qui se révèle fondamentale lorsque les agents n'ont pas accès à l'ensemble de l'information concernant le système.
- Certaines extensions permettent de représenter des problèmes de prises de décision multi-agents correspondant aux problèmes auxquels nous cherchons à fournir une réponse
- Il existe des algorithmes permettant de résoudre en partie la problématique de construction automatique de SMAs réactifs.

Cette partie a pour objectif de présenter ces cadres formels, et la manière dont ils répondent aux questions posées par Ferber concernant la conception de systèmes multi-agents.

Dans un premier temps, nous nous intéresserons aux deux premières questions posées par Ferber, à savoir quelles sont **les architectures des agents et quelles sont les formes d'interaction autorisées par ces formalismes**. Nous décrirons tout d'abord les Processus de Décision Markoviens (ou Markov Decision Process (MDP)) qui permettent de représenter des problèmes de prise de décision impliquant un seul agent. Nous décrirons ensuite les Processus de Décision Markoviens Décentralisés Partiellement Observés (ou Decentralized Partially Observable Markov Decision Process (DEC-POMDP)) qui permettent de représenter des problèmes multi-agents et nous focaliserons sur la manière dont est représenté le concept d'interaction, central dans les systèmes multi-agents.

Dans un second temps, nous nous intéresserons à la troisième question posée par Ferber à savoir comment **construire** et faire évoluer les comportements des agents pour que la collectivité puisse répondre à un problème donné. Cette problématique correspond aux problèmes formalisés dans les cadres formels markoviens et nous décrirons les algorithmes qui permettent d'y répondre.

Comme les cadres DEC-POMDPs cherchent à représenter des SMAs à partir d'un cadre formel fondé sur la notion de rationalité, nous nous demanderons dans ce chapitre comment la notion d'interaction est représentée. Nous nous demanderons aussi dans quelle mesure cette

représentation conditionne les algorithmes de construction des comportements des agents et si les contraintes de localité sont effectivement respectées.

3.1 Formalismes inspirés des MDP

Dans un premier temps, nous présenterons les problèmes de prises de décision mono-agent et les Processus de Décision Markovien qui permettent de les formaliser. Ensuite, afin de pouvoir représenter les systèmes qui nous intéressent, nous nous concentrerons sur :

- des agents dotés de perceptions partielles, ce qui nous amènera à considérer les POMDP
- des systèmes constitués de plusieurs agents, ce qui nous amènera à considérer les DEC-POMDP.

Pour le formalisme DEC-POMDP, nous nous focaliserons particulièrement sur la manière dont les systèmes multi-agents sont représentés et dont le concept d'interaction est instancié.

3.1.1 Problème de conception mono-agent

Wooldridge propose une architecture abstraite comme modèle générique d'agent réactif [Woo01]. Cette architecture permet de formaliser notre vision de l'agent et nous permettra de faire un lien avec le cadre formel que nous utiliserons dans la section 3.1.2.

Dans [Woo01], l'environnement est supposé être décrit par un ensemble fini d'états discrets

$$E = \{e, e', e'', ...\}$$

Les actions possibles de l'agent sont définies par un ensemble d'actions discrètes

$$A = \{a, a', ...\}$$

L'exécution du comportement d'un agent dans son environnement construit une suite constituée de couples état-action :

$$e_0 \overset{a_0}{\rightarrow} e_1 \overset{a_1}{\rightarrow} e_2 \overset{a_2}{\rightarrow} e_3.... \overset{a_{u-1}}{\rightarrow} e_u$$

R désigne l'ensemble des résultats d'exécution du système à pas de temps fini. R^A est le sous-ensemble des résultats d'exécution qui s'achèvent par une action. R^E est le sous ensemble des résultats qui s'achèvent par un état.

Afin de représenter les effets des actions des agents sur leur environnement, Wooldridge introduit une fonction "state transformer" :

$$\tau : R^A \rightarrow E$$

Il s'agit d'une fonction de l'ensemble de l'historique du système (jusqu'à la dernière action entreprise) vers une distribution sur les états possibles de l'environnement.

De plus, on suppose l'environnement inaccessible : un agent ne peut prendre sa décision qu'à partir de ses perceptions et non pas à partir de l'état réel du système. On définit un ensemble P discret correspondant à l'ensemble des perceptions de l'agent et une fonction

$$see : E \rightarrow P$$

qui associe à chaque état de l'environnement la perception qu'en a l'agent. Dans le cas d'un environnement accessible, la fonction *see* est égale à la fonction identité.

Un agent est caractérisé par une fonction de décision qui lui permet de choisir une action à entreprendre parmi les actions possibles. Comme les agents sur lesquels nous nous concentrons sont des agents sans mémoire à court terme, leur fonction de décision dépend donc uniquement de la perception courante de l'agent et possède le profil suivant :

$$A_g : P \rightarrow A$$

La boucle perception-action de l'agent peut alors s'exprimer à l'aide des fonctions précédentes :

- L'environnement se trouve dans l'état e
- L'agent perçoit son environnement : $p \leftarrow see(e)$
- Il décide de son action $a \leftarrow A_g(p)$
- Cette action est ajoutée à l'historique du système $r \leftarrow r \cup a$
- Il effectue cette action et l'environnement est modifié en conséquence $e \leftarrow \tau(r)$
- L'historique du système est mise à jour $r \leftarrow r \cup e$

Conformément au chapitre 2, poser un problème de construction d'un agent réactif rationnel consiste à fixer une fonction de performance individuelle à optimiser, des actions et des perceptions possibles de l'agent et les lois d'évolution du système.

Construire le comportement d'un agent rationnel consiste alors à trouver une fonction de décision maximisant la fonction de performance individuelle. Le cadre formel des Processus de Décision Markoviens permet justement de formaliser cette problématique de conception et plusieurs algorithmes permettent d'y trouver une réponse exacte.

3.1.2 Processus Décisionnels de Markov

Formalisme Un MDP (Markov Decision Process) est défini par un tuple $< S, A, T, R >$.

- S désigne un ensemble d'états discrets
- A désigne un ensemble d'actions discrètes
- $T : S \times A \times S \rightarrow [0, 1]$ est une fonction appelée matrice de transition
- $R : S \times A \rightarrow \mathbb{R}$ est une fonction appelée fonction de récompense

État et action L'ensemble S désigne les configurations possibles du monde. A désigne l'ensemble des actions que l'agent peut choisir d'accomplir en vue de modifier l'état de son environnement.

Matrice de transition La matrice de transition T représente la dynamique du système. Elle caractérise les réactions de l'environnement aux actions émises par l'agent et correspond à la fonction "state transformer" de Wooldridge.

Un processus de décision markovien vérifie la **propriété de Markov**. Celle-ci stipule que la dynamique de l'état du système à un instant t ne dépend pas de son historique, à savoir la suite des couples actions-états visités $h_t = (s_0, a_0, s_1, a_1, ..., s_t)$: la probabilité d'atteindre un état s à la date $t + 1$ dépend uniquement de l'état s du système à la date t et de l'action a exécutée à la date t.

$$P(s_{t+1}|h_t, a_t) = P(s_{t+1}|s_0, a_0, s_1, a_1, ..., s_t, a_t) = P(s_{t+1}|s_t, a_t) = T(s_t, a_t, s_{t+1})$$

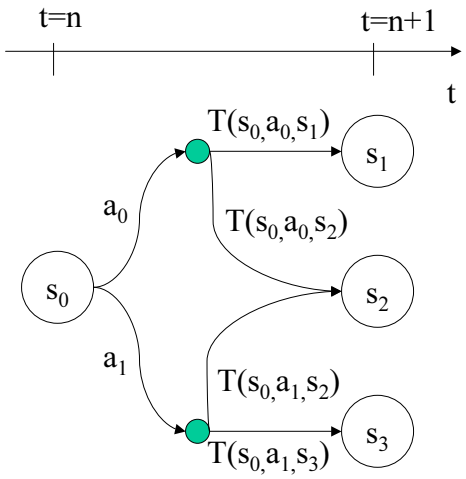


FIG. 3.1 – Représentation d'un exemple de matrice de transition. Lorsque l'agent se trouve dans l'état s_0, il peut accomplir deux actions a_0 ou a_1. Lorsqu'il effectue l'action a_0, il a une probabilité $T(s_0, a_0, s_1)$ d'arriver dans l'état s_1 et une probabilité $T(s_0, a_0, s_2)$ d'arriver dans l'état s_2. Lorsqu'il effectue l'action a_1, il a une probabilité $T(s_0, a_1, s_2)$ d'arriver dans l'état s_2 et une probabilité $T(s_0, a_1, s_3)$ d'arriver dans l'état s_3

Pour un état de départ s donné et une action a donnée, $T(s, a, s')$ fournit la probabilité d'atteindre l'état d'arrivée s'. En tout généralité, la matrice de transition permet de représenter des environnements stochastiques pour lesquels "les mêmes causes n'entraînent pas forcément les mêmes effets" : pour un état donné et une action donnée, l'état d'arrivée n'est pas forcément le même (cf figure 3.1). Cette représentation des lois d'évolution du monde permet de modéliser des incertitudes quant au bon déroulement des actions.

De telles incertitudes dans un MDP peuvent provenir d'un monde intrinsèquement stochastique ou du niveau de modélisation de cet environnement. Par exemple, on peut modéliser par un MDP le problème de navigation d'un robot mobile autonome à roues (comme [CKK96]). On ne souhaite cependant pas modéliser finement les problèmes d'adhérence des roues au sol et les glissements lors des déplacements du robot. Il est possible de simuler ces aléas par une matrice de transition stochastique qui ne rendra pas intégralement compte de la complexité des lois de la mécanique mais pourra constituer une première approche pour le problème de navigation permettant de considérer la possibilité d'aléas dans l'évolution du système.

On supposera en outre l'environnement perçu comme statique au sens de Jennings (cf partie 2.1.2.3) : l'état de l'environnement n'évolue pas entre deux prises de décision de l'agent.

Rappelons qu'un environnement peut aussi être
- un environnement stationnaire pour lequel les lois d'évolution n'évoluent pas au cours du temps : $\forall t, T_t = T$
- un environnement déterministe

$$\forall s \in S, \forall a \in A, \exists s' \in S \text{ tel que } T(s, a, s') = 1 \text{ et } \forall s'' \neq s', T(s, a, s'') = 0$$

Comme il est de toutes façons possible de modéliser un environnement non stationnaire par un

environnement stationnaire en intégrant le temps dans l'espace d'état, nous nous intéresserons par la suite à ce type d'environnements. Nous nous focaliserons sur des environnements non-déterministes puisque dans un cadre multi-agent, la présence d'autres agents fait que même si les lois du monde sont déterministes, l'évolution du monde perçue par un agent pourra être stochastique car celle-ci est dépendante des prises de décision des autres agents.

Fonction de récompense La fonction de récompense R caractérise les motivations individuelles de l'agent. Lorsque l'agent se trouve dans un état s et qu'il effectue une action a, il reçoit une récompense immédiate $R(s, a)$.

La rationalité d'un agent s'exprime sous la forme d'un critère de performance individuel lié à cette fonction de récompense. Dans un MDP, à partir d'une fonction de récompense immédiate donnée, plusieurs critères de performance individuelle peuvent être définis.

Ces critères de performance dépendent de la catégorie de problème que l'on cherche à résoudre. S'il s'agit d'un problème en horizon fini, c'est à dire pour lequel l'agent évolue pendant un nombre de pas de temps fixe n connu a priori, un critère de performance possible est la somme des récompenses reçues le long de la trajectoire de l'agent $E[\sum_{t=0}^{n} r_t]$

S'il s'agit d'un problème en horizon infini, c'est à dire pour lequel le processus évolue indéfiniment, d'autres critères de performance doivent être définis. Parmi les critères habituellement utilisés, on peut citer :

- le critère moyen : $\frac{1}{n}.E[\sum_{t=0}^{n} r_t]$
- le critère gamma pondéré : $E[\sum_{t=0}^{\infty} \gamma^t r_t]$ avec $\gamma \in [0, 1[$

Nous nous concentrerons sur des horizons infinis, et opterons pour le critère γ-pondéré qui est le plus simple à manipuler. Le paramètre γ appelé 'facteur de décompte' (*'discount factor'*) peut être compris comme

- un paramètre permettant d'assurer la convergence de la suite de récompenses reçues
- un paramètre représentant le risque que le processus de décision s'arrête à chaque pas de temps. (probabilité correspondante $1 - \gamma$)
- un paramètre permettant de déterminer l'importance du futur dans la prise de décision.

3.1.2.1 Politique

Caractérisation Le comportement de l'agent est modélisé par une politique π. La politique correspond à la fonction de prise de décision de l'agent. Une politique est une fonction qui peut

- dépendre uniquement de l'état courant du système (agent réactif sans mémoire) ou dépendre de l'historique des états-actions du système (agent qui utilise toute son expérience pour décider de son action)
- être déterministe (pour des entrées identiques, une seule action est choisie) ou stochastique (pour des entrées identiques, plusieurs actions sont possibles, le choix de l'action effective est déterminée par une densité de probabilité sur les actions)

On distingue donc 4 types de politiques (H désigne l'ensemble des historiques du système) :

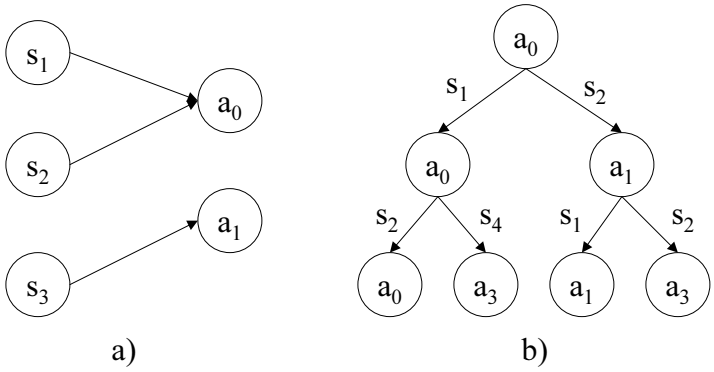


FIG. 3.2 – Politiques déterministes a) markoviennes (l'action choisie ne dépend que de l'état courant) et b) histoire-dépendantes (l'action choisie dépend des états et des actions antérieures).

- les politiques markoviennes déterministes (cf figure 3.2 a)) $\pi : S \rightarrow A$. Leur ensemble est noté Π^{MD}
- les politiques histoire-dépendantes déterministes (cf figure 3.2 b)) : $\pi : H \rightarrow A$. Leur ensemble est noté Π^{HD}
- les politiques markoviennes stochastiques $\pi : S \times A \rightarrow [0, 1]$. Leur ensemble est noté Π^{MS}
- les politiques histoire-dépendantes stochastiques $\pi : H \times A \rightarrow [0, 1]$ Leur ensemble est noté Π^{HS}

Enfin, une politique peut être stationnaire (auquel cas, $\forall t, \pi_t = \pi$). Comme nous nous concentrons sur des environnements stationnaires en horizon infini, nous nous limiterons à des politiques stationnaires dans un cadre mono-agent.

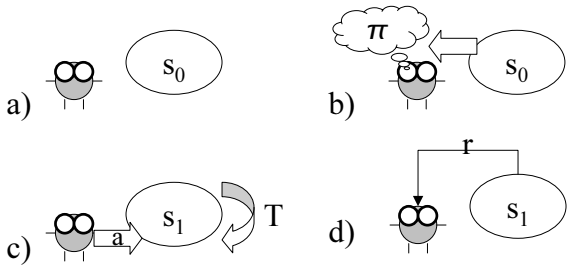


FIG. 3.3 – Exécution d'un MDP constitué de plusieurs étapes : a) état initial, b) étape de décision, c) étape d'exécution de l'action, d) étape de modification de l'état

Exécution d'une politique Pour un tuple $< S, A, T, R >$ donné, l'exécution d'une politique π correspond à la boucle perception-action d'un agent et est constituée de cycles de 4 étapes (cf fig 3.3) :

- l'agent observe l'état s_0 de l'environnement (cf fig 3.3 a)
- il détermine l'action a qu'il souhaite entreprendre en fonction de sa politique π, $a \leftarrow \pi(s_0)$ (cf fig 3.3 b)
- il effectue cette action et l'état du monde s'en trouve modifié $s_1 \leftarrow T(s_0, a, s')$ (cette

dernière expression cache en réalité une réalisation selon la densité de probabilité $T(s_0, a)$) (cf fig 3.3 c))
- il reçoit une récompense $r \leftarrow R(s_0, a)$ (cf fig 3.3 d)

Résoudre un MDP Un MDP définit un problème de prise de décision séquentiel. Résoudre un MDP consiste à calculer le comportement d'un agent étant donné un ensemble d'actions possibles, des lois du mondes (pas forcément connues), une fonction de performance individuelle et un état de départ. En d'autres termes, pour un tuple $< S, A, T, R >$ donné, résoudre un MDP consiste à trouver parmi une famille de politiques une politique optimisant un critère de performance (γ pondéré dans notre cas) défini à partir de la fonction de récompense.

L'adjectif séquentiel provient du fait que la performance individuelle est évaluée le long de la trajectoire du système et non pas de manière instantanée. Ainsi dans certains cas, l'action déterminante pour obtenir une récompense positive peut avoir lieu plusieurs pas de temps avant l'obtention effective de cette récompense. Par exemple, un agent devant atteindre la sortie d'un labyrinthe est confronté à un problème de décision à chaque intersection. Parfois (comme le montre l'exemple de la figure 3.4), l'action déterminante qui conditionne la réussite de l'agent peut être le choix du couloir à emprunter à la première intersection rencontrée.

Les techniques permettant de calculer une politique optimale seront présentées ultérieurement dans la partie 3.2.

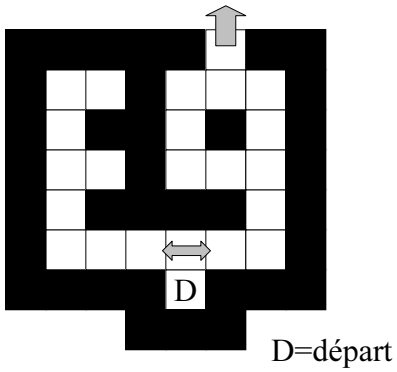


FIG. 3.4 – Exemple : labyrinthe et décision séquentielle. Le choix de l'agent à la première intersection conditionne ses récompenses futures

Nous avons présenté le cadre formel des MDP qui permet de représenter des problèmes de prises de décision séquentielles pour un système constitué d'un agent ayant une perception totale de son environnement.

Comme nous souhaitons aborder des problèmes multi-agents pour lesquels plusieurs agents dotés de perceptions partielles évoluent dans un même environnement, il nous faut considérer deux extensions des MDPs :
- la première, la classe des POMDPs, décrite dans la partie 3.1.3 qui traite de la problématique d'observabilité partielle.
- la seconde, la classe des DEC-POMDPs, décrite dans la partie 3.1.4 qui propose un cadre

formel permettant de représenter des systèmes constitués de plusieurs agents.

3.1.3 Processus Décisionnel de Markov Partiellement Observé

Du fait des contraintes de localité, les problèmes posés dans un cadre multi-agents se heurtent tout d'abord à des problématiques liées aux observabilités partielles. Afin de séparer cette problématique d'observabilité partielle des considérations liées à l'aspect multi-agents, nous présentons dans un premier temps les problèmes qui apparaissent dans un cadre mono-agent et qui pourront réapparaître dans un cadre collectif.

Le problème de prise de décision séquentielle pour un agent capable de percevoir localement son environnement peut être représenté par les Processus Décisionnels de Markov Partiellement Observables (ou POMDP : Partially Observable Markov Decision Process).

3.1.3.1 Processus de décision markovien partiellement observable

Un Processus de décision markovien partiellement observable (POMDP) est défini par un tuple $< S, A, T, R, \Omega, O >$:

- un ensemble d'états S (supposé fini)
- un ensemble d'actions A (supposé fini)
- une matrice de transition $T : S \times A \times S \rightarrow [0, 1]$
- une fonction de récompense $R : S \times A \rightarrow \mathbb{R}$
- un ensemble d'observations Ω (supposé fini)
- une fonction d'observation $O : S \times \Omega \rightarrow [0, 1]$

Contrairement à un MDP, l'environnement est supposé inaccessible. Désormais, l'agent n'a plus accès directement à son état s_t mais à une observation $o_t \in \Omega$ de cet état. De plus, le processus d'observation peut être stochastique. Un agent peut alors avoir deux observations différentes d'un même état à deux instants différents. La fonction d'observation O associe à chaque état s, la probabilité $O(s, o)$ d'observer o. L'aspect stochastique de cette fonction peut ainsi permettre de représenter des capteurs potentiellement défectueux.

3.1.3.2 La problématique de l'observabilité partielle

Ambiguïté sur l'état Dans un POMDP, un agent ne peut plus distinguer avec certitude l'état dans lequel il se trouve. Il peut néanmoins tirer parti de son historique des couples observations-actions pour tenter de l'inférer.

L'exemple du tigre présenté dans la première partie peut être formalisé par un POMDP : le système est constitué de deux états, l'état s_1 pour lequel le tigre est situé derrière la porte de droite, l'état s_2 pour lequel le tigre est situé derrière la porte gauche. L'agent peut faire trois actions : il peut ouvrir la porte gauche, ouvrir la porte droite ou essayer d'écouter pour tenter de détecter la position du tigre. Enfin, l'agent peut avoir accès à trois observations : une signifiant "vide", une "le tigre est a gauche" et une "le tigre est à droite". Lorsque l'agent écoute, il reçoit une récompense négative mais perçoit une observation différente de l'observation "vide". Cette observation n'est pas forcément sûre puisqu'il a une probabilité non nulle d'obtenir une perception ne correspondant pas à la réalité.

Ainsi, l'agent ne peut distinguer avec certitude les deux états s_1 et s_2. Le problème vient du fait que dans ces deux états, les actions à entreprendre sont opposées : une action pouvant être bénéfique dans un cas sera néfaste dans l'autre cas.

Observabilité partielle et mémoire Un moyen pour réduire le risque lié à l'ouverture d'une porte consiste alors à essayer d'écouter plusieurs fois pour acquérir plus d'informations sur son environnement et pouvoir décider de manière adéquate de l'action à exécuter. L'agent a cependant besoin d'une mémoire à court terme pour se souvenir des résultats d'une expérience passée.

Dans un POMDP, de manière générale, la politique optimale est à chercher parmi les politiques histoire-dépendante. Chaque politique représente un arbre dont les branches sont des observations et les noeuds des actions à entreprendre.

Résoudre un POMDP revient alors à construire un tel arbre (ou en horizon infini un automate) permettant de maximiser la somme des récompenses lorsqu'on exécute la politique associée.

Bilan du formalisme POMDP Les problèmes posés dans le cadre POMDP sont beaucoup plus complexes que les problèmes posés dans un cadre MDP en raison de la recherche dans l'espace des politiques histoire-dependante. La problématique de l'observabilité partielle remet par exemple totalement en cause l'architecture interne des agents puisqu'elle nécessite une mémoire à court terme pour pouvoir être abordée correctement.

Cette problématique se pose encore dans le formalisme DEC-POMDP que nous présentons dans la partie suivante.

3.1.4 Processus Décisionnel de Markov Décentralisé Partiellement Observé

Un système multi-agents est décrit par un ensemble d'agents, les actions possibles pour chacun d'entre eux, les interactions que les agents peuvent entretenir entre eux et les lois d'évolution du monde.

La caractéristique d'un tel système est qu'aucun agent ne contrôle entièrement la dynamique du système. Par contre, en effectuant une action, chaque agent tente d'influer sur le processus décrit par les lois du monde.

Un problème multi-agents coopératif consiste alors à construire les différentes politiques des agents pour que le système maximise un critère de performance globale défini par rapport à la dynamique du système.

3.1.4.1 Le modèle DEC-POMDP

Le modèle DEC-POMDP (Decentralized Partially Observable Markov Decision Process) a été proposé par Bernstein [BGIZ02] pour représenter des problèmes de décisions multi-agents et constitue le modèle de référence à l'heure actuelle.

Ce modèle consiste à représenter l'environnement comme un processus et les agents comme des entités pouvant influencer l'évolution du processus et la trajectoire empruntée par le système. C'est alors l'ensemble des actions émises simultanément par les agents qui décide de l'évolution effective du système. En cela, le modèle DEC-POMDP a beaucoup de points communs avec le modèle influence-réaction proposé par Ferber et Muller dans [FM96] et décrit dans la partie 2.2.4.1.

Un DEC-POMDP est un tuple $< S, A_i, T, \Gamma_i, O, R >$:
- S désigne l'ensemble des états possibles du monde (supposé fini)
- A_i l'ensemble des actions possibles pour l'agent i. Une *action jointe* $a = (a_0, a_1, ..., a_n)$ est un tuple constitué de l'ensemble des actions émises par les n agents. L'espace des actions jointes est constitué par l'ensemble de ces tuples $A = \times A_i$
- $T : S \times A \times S \rightarrow [0, 1]$ définit la matrice de transition du système
- Γ_i est l'ensemble des observations possibles pour l'agent i (supposé fini)
- $O : S \times A \times \Gamma \rightarrow [0, 1]$ définit les probabilités d'observation (sur les actions et observations jointes)
- $R : S \times A \rightarrow \mathbb{R}$ désigne la fonction de récompense globale du système

Il est à retenir que le terme décentralisé (*"decentralized"*) fait référence à l'aspect décentralisé de l'exécution mais ne concerne en rien les méthodes de construction des comportements mises en oeuvre comme on le verra par la suite.

Exemple Afin d'illustrer par la suite les différents éléments d'un DEC-POMDP, considérons un exemple très simple de système multi-agents constitué d'un cube pouvant coulisser avec frottements selon un axe et d'agents pouvant exercer des forces (de valeur discrète) sur ce cube. Pour ne pas considérer des problèmes continus, nous supposerons que l'espace d'état a été discrétisé et que les lois du monde sont exprimées en conséquence. La dynamique du cube est définie par les lois de la mécanique classique (l'accélération est égale à la somme des forces appliquées en tenant compte des forces de frottement). Chaque agent en exerçant une force sur le cube tente d'influencer la trajectoire suivie par l'objet (dans l'espace d'état constitué par le couple (position, vitesse)). Les agents décident individuellement de la force qu'ils souhaitent exercer. La dynamique effective du système dépendra de l'ensemble des forces appliquées par tous les agents et sera résolue par les lois du monde (à savoir les lois de la mécanique classique). La fonction de récompense est définie par rapport à l'état global du système. Un exemple possible serait de générer une récompense lorsque le cube arrive dans une certaine position à une certaine vitesse avec une accélération nulle. La position et la vitesse du système n'est pas contrôlée par un seul agent mais chacun participe à son évolution. Les agents doivent alors synchroniser leurs actions pour répondre correctement au problème.

Etat, Action S désigne l'état global du système (contenant éventuellement les variables décrivant les agents). Pour notre exemple, cet état correspond au couple (position, vitesse) du cube. A_i désigne l'ensemble des actions possibles pour un agent i. Dans notre exemple, A_i correspond aux différentes forces que peut exercer un agent i sur le cube.

Transition La fonction T correspond aux lois du monde et aux réponses de l'environnement aux influences exercées par les agents. Le résultat de cette fonction $T : S \times A \times S \rightarrow [0, 1]$ dépend directement de l'action jointe émise par l'ensemble des agents.

Dans notre cas, il s'agit des lois de la mécanique qui modifient l'état du cube (position, vitesse) en fonction de l'ensemble des forces exercées par les agents.

C'est la matrice de transition qui se charge d'intégrer la résultante des actions émises par les agents pour faire évoluer le système. Dans notre exemple, si deux agents émettent des forces opposées sur le cube, la vitesse de celui-ci restera constante (modulo les forces de frottement) comme cela a pu être défini au sein de la matrice de transition globale.

Récompense Notre objectif est de construire des systèmes **coopératifs** dans lesquels les agents doivent collaborer en vue d'atteindre un objectif commun. Comme nous l'avons décrit dans le paragraphe 2.2.2.1 présentant notre point de vue sur la rationalité dans les SMA, l'évaluation du système dépend de la réalisation de l'objectif commun et non pas directement des comportements individuels qui ne constituent qu'un moyen de réaliser cet objectif.

La fonction de récompense rend compte de l'aspect global du problème posé à la collectivité. Elle est définie sur l'espace S et fournit la récompense immédiate à l'ensemble des agents en fonction de l'action jointe émise dans le système : $R : S \times A \rightarrow \mathbb{R}$. Le critère de performance sera un critère de performance global construit à partir des récompenses reçues par le système. Nous conserverons le critère γ pondéré identique au critère présenté dans la partie 3.1.2.

Dans notre exemple, la fonction de récompense dépend de l'état du monde (vitesse et position du cube) et de l'action jointe émise par les agents (l'accélération résultant des forces appliquées doit être nulle). Il s'agit bien d'une récompense globale donnée à l'ensemble des agents en fonction de l'état global du système et de l'action jointe.

Observabilité partielle Chaque agent ne perçoit que partiellement l'état global du système. La fonction d'observabilité O associe une probabilité $o = (o_0, o_1, ..., o_n)$ pour tout état global s donné, toute action jointe $a = (a_0, a_1, ..., a_n)$ et toute observation jointe .

$O(s, a_0, a_1, ..., a_n, o_0, o_1, ..., o_n)$ donne la probabilité que chaque agent i observe o_i en fonction de l'état global du système s et de l'action jointe a.

Pour notre exemple, on peut imaginer que les agents ne peuvent percevoir que la vitesse du cube dans un seul sens (différent pour chacun).

Il existe un cas particulier de DEC-POMDP dont on parlera succinctement par la suite : il s'agit du DEC-MDP. Un DEC-MDP est un DEC-POMDP pour lequel l'ensemble des observations reçues par les agents permet de reconstituer l'état global du système. En d'autres termes, s'il est possible d'accéder à l'ensemble des observations des agents du système, l'état global du système est accessible.

Politique Chaque agent est caractérisé par une politique individuelle π_i qui est de manière générique histoire-dépendante. Cette politique π_i associe à chaque séquence de perceptions de l'agent l'action qu'il effectue pour influencer la dynamique du système.

La présence de politiques locales associées à chaque agent permet de construire des systèmes autonomes au sens de Jennings (présentés dans le chapitre 2.2.2.1). Chaque agent peut prendre ses décisions de manière autonome sans nécessiter l'intervention des autres agents et toute les

actions émises par le système sont à l'initiative des agents.

Maintenant que nous avons défini les différents constituants d'un DEC-POMDP, il est possible de s'intéresser au problème d'optimisation qui y est lié. Pour cela, nous présenterons tout d'abord la manière dont un DEC-POMDP est exécuté et comment les récompenses sont reçues par le système.

Exécution d'un DEC-POMDP L'exécution d'un DEC-POMDP est caractérisée par :

- l'exécution (contrôle) décentralisé,
- la simultanéité de l'émission des actions,
- la résolution par l'environnement (matrice de transition T) des actions et des conflits entre leurs conséquences.

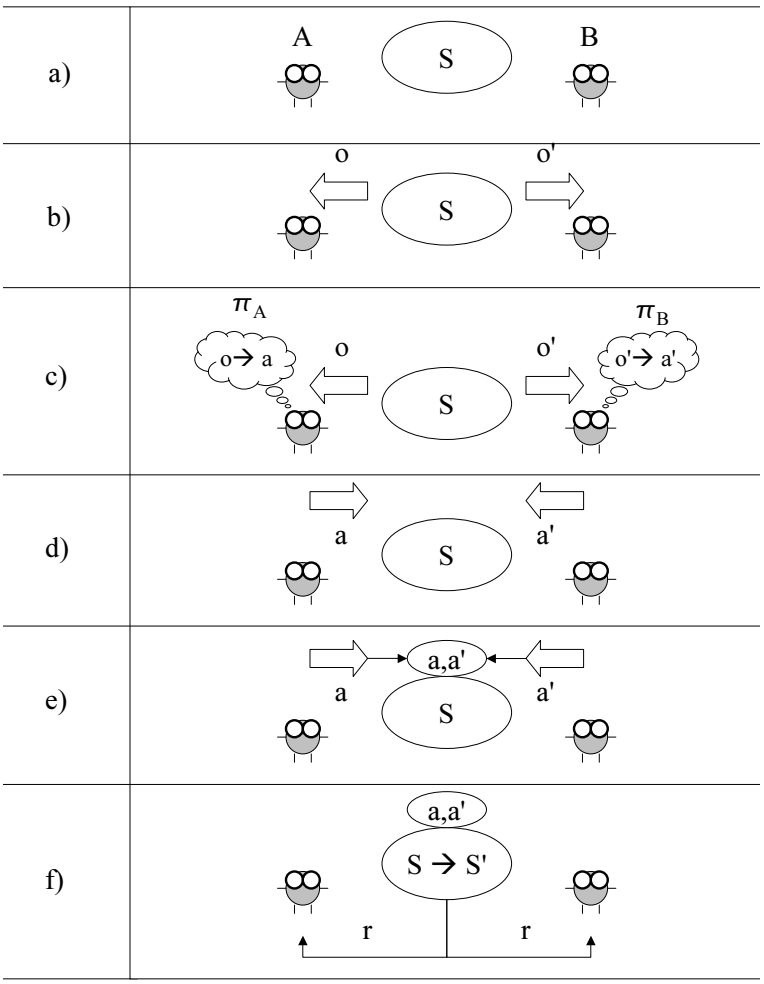


FIG. 3.5 – Exécution d'un DEC-POMDP

Ainsi, l'exécution d'un DEC-POMDP se décompose en cycles de plusieurs phases (cf fig 3.5)

1. Le système se trouve dans un état global s_t (cf fig 3.5 a))

2. Chaque agent i perçoit son environnement et reçoit une observation $o_i = O(s_t, a_t)$. (cf fig 3.5 b))
3. Chaque agent i décide individuellement de l'action à choisir à partir de sa politique locale π_i (cf fig 3.5 c))
4. Chaque agent émet simultanément aux autres agents du système une action a_i. L'ensemble des a_i définit l'action jointe a (cf fig 3.5 d))
5. Le système évolue en fonction de son état global s, de l'action jointe a et de la matrice de transition T. (cf fig 3.5 e))
6. Le système délivre une récompense collective R perçue par tous les agents. (cf fig 3.5 f))

3.1.4.2 Problème associé au DEC-POMDP

Chaque agent i est décrit par une politique locale π_i. Une politique jointe $(\pi_1, \pi_2, ..., \pi_n)$ est un tuple constitué de l'ensemble des politiques locales des agents du système. Résoudre un DEC-POMDP consiste à trouver une politique jointe maximisant un critère de performance global défini à partir des récompenses globales. On conservera le critère γ pondéré.

3.1.4.3 Interaction dans les DEC-POMDP

Les DEC-POMDPs permettent ainsi de représenter des systèmes multi-agents et de formaliser la problématique de construction des comportements d'agents réactifs. Comme la notion d'interaction est centrale dans ces systèmes, il est naturel de s'intéresser à la manière dont elle est instanciée dans ce cadre formel chargé initialement de représenter des systèmes constitués d'un seul agent.

Les comportements des agents sont couplés par l'intermédiaire de la matrice de transition. C'est elle qui décrit l'évolution du système en fonction de l'action jointe. Les conflits entre les agents sont résolus lors de l'évolution du système par cette même matrice. De ce fait, seules des interactions indirectes médiées par l'environnement sont possibles.

Par conséquent, un agent n'a pas de vision locale des conséquences à long terme de ses actions pour deux raisons

- Tout d'abord, il ne peut pas savoir a priori avec quel agent il va interagir lorsqu'il modifie l'environnement puisque ces modifications seront perçues par les autres agents en fonction de leurs comportements et de l'état global du système.
- De plus, les conséquences réelles et ponctuelles d'une action émise par un agent dépendent des actions émises par les autres agents. Ce qui fait que les lois du monde perçues par un agent ne sont plus stationnaires et peuvent évoluer lorsque les autres agents modifient leurs comportements. Il est donc nécessaire pour un agent de prendre en compte d'une manière ou d'une autre la présence d'autres entités dans le système afin de choisir au mieux ses actions.

L'absence de représentation explicite au niveau individuel de la notion d'interaction pose alors problème puisque les agents n'ont pas accès à la matrice de transition dans laquelle les couplages entre agents sont représentés et qu'ils ne disposent donc pas d'une structure nécessaire pour appréhender la présence d'autres agents dans le système. Ce manque du formalisme reporte

ainsi de nombreux problèmes au niveau des processus de résolution.

Ces problèmes ainsi que leurs conséquences seront décrits de manière plus précise dans la partie 3.4 qui traite des approches permettant de construire des systèmes multi-agents dans un cadre DEC-POMDP.

3.1.4.4 Bilan du formalisme DEC-POMDP

Dans cette partie nous avons présenté le cadre formel des DEC-POMDP introduit il y a quelques années par Bernstein ([BGIZ02]).

Ce cadre formel permet de modéliser des agents réactifs caractérisés par un ensemble d'actions A_i, un ensemble de perceptions Γ_i et une fonction de prise de décision individuelle nommée politique π, un monde et ses lois d'évolution caractérisées par l'ensemble S et la matrice T ainsi qu'un problème posé à la collectivité sous la forme d'une fonction de récompense globale R.

Il pose le problème de construction de comportements collectifs consistant à trouver les politiques réactives π_i des agents étant données leur architecture interne, leur architecture externe, les lois d'évolution du monde et la tâche à résoudre. En cela, le cadre formel DEC-POMDP constitue une première brique permettant de répondre à notre problématique : un formalisme permettant de représenter les problèmes de construction des comportements de SMA.

Cependant, les systèmes qu'il est possible de représenter dans ce cadre sont limités : les interactions entre les agents sont définies de manière globale. La notion d'interaction directe résolue selon les lois comportementales des agents et permettant à un agent de communiquer avec d'autres agents n'est pas directement envisageable puisque les comportements des agents sont uniquement couplés par l'intermédiaire des lois environnementales représentées dans la matrice de transition T.

Il reste à répondre à la seconde problématique consistant à produire les comportements des agents en tirant parti du formalisme DEC-POMDP. De nombreuses méthodes de résolution ont été proposées, un certain nombre d'entre elles vont être décrites dans la suite de ce manuscrit (cf partie 3.4). Nous montrerons que le fait que les interactions soient décrites au niveau global nécessite de se placer à ce niveau pour construire des politiques jointes et réduit fortement l'utilisation d'un tel formalisme.

3.1.5 Bilan des formalismes

A l'issue de cette partie, nous avons présenté des formalismes issus des processus décisionnels de markov. Ces formalismes peuvent s'organiser selon le tableau 3.6

	mono-agent	multi-agents
observabilité totale	MDP	MMDP (cf partie A.1.2)
observabilité partielle	POMDP	DEC-PODMP

FIG. 3.6 – Différents formalismes markoviens

Ces formalismes permettent de représenter des problèmes de prise de décision individuelle ou collective avec des observabilités totales ou partielles. Ils posent le problème de construction des comportements d'agents évalués par rapport à un critère de performance numérique.

Dans les parties suivantes, nous allons nous attarder sur les algorithmes qui manipulent ces formalismes pour construire de manière automatique le comportement d'agents réactifs.

3.2 Résolution mono-agent en observabilité totale

Cette partie traite de la résolution des modèles markoviens représentant des problèmes de prise de décision mono-agent (à savoir les MDPs). Cette partie nous semble importante pour deux raisons. D'une part, ces techniques constituent le fondement des approches utilisées pour résoudre des DEC-POMDPs comme nous le verrons dans la partie 3.4. D'autre part, nous souhaitons utiliser des techniques de construction de comportement individuel pour construire des comportements collectifs. Nous serons amenés à réutiliser ces techniques dans notre proposition (cf chapitre 6).

Le problème sur lequel nous allons nous concentrer tout d'abord consiste à construire à partir d'un MDP $< S, A, T, R >$, une politique $\pi^* : S \rightarrow A$ qui maximise à l'exécution le critère de performance choisi à savoir le critère γ-pondéré.

3.2.1 Fonction de valeur

Description Un moyen d'aborder le problème de construction du comportement de l'agent consiste à mesurer les performances associées à chaque politique π.

Pour cela, à chaque π est associée une fonction de valeur v^π. Il s'agit d'une fonction $v^\pi : S \rightarrow \mathbb{R}$ qui, pour la politique π, associe à tout état s la valeur espérée du critère de performance obtenue si un agent dans l'état s suit la politique π.

En considérant le critère gamma-pondéré, la fonction de valeur pour une politique donnée π s'écrit sous la forme :

$$v^\pi(s) = E^\pi[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s]$$

Elle représente la quantité de récompense décomptée à long terme que peut espérer recevoir un agent s'il se trouve dans s et suit la politique π.

Évaluation d'une politique L'algorithme *policy evaluation* permet d'évaluer de manière itérative la fonction de valeur v^π d'une politique π donnée lorsque la fonction de récompense R et la matrice de transition T sont connues (cf algorithme 1).

Il a été prouvé que cet algorithme convergeait vers v^π (cf [Put94]). Ainsi, il permet de trouver les valeurs de performances associées à deux politiques π_1 et π_2 et de comparer ces politiques entre elles. Il constitue une première étape pour construire des politiques maximisant le critère de performance individuelle.

Algorithme 1 Policy evaluation

entree $\leftarrow \pi, MDP, \gamma, \epsilon$
initialiser v_0, $t \leftarrow 0$
répéter
  $t \leftarrow t+1$
  pour tout $s \in S$ **faire**
    $v_t(s) \leftarrow R(s,\pi(s)) + \gamma \Sigma_{s' \in S} T(s,\pi(s),s') v_{t-1}(s')$
  fin pour
jusqu'à $max_s |v_t(s) - v_{t-1}(s)| < \epsilon$
retour $\leftarrow v_t$, l'évaluation de π

3.2.2 Politiques optimales

3.2.2.1 Résultat préliminaire

Soit une politique $\pi' \in \Pi^{MD}$. S'il y a une politique $\pi \in \Pi^{MD}$ pour laquelle il existe un s_0 tel que $v^{\pi'}(s_0) > v^{\pi}(s_0)$ alors il existe une politique markovienne $\pi'' \in \Pi^{MD}$ telle que $\forall s, v^{\pi''}(s) \geq v^{\pi}(s)$. (preuve dans [Put94])

$$\exists \pi' \in \Pi^{MD}, \exists s_0, v^{\pi'}(s_0) > v^{\pi}(s_0) \Rightarrow \exists \pi'' \in \Pi^{MD}, \forall s, v^{\pi''}(s) > v^{\pi}(s)$$

Cette propriété permet de définir la notion de politique optimale sans avoir besoin de faire référence à un état particulier, puisque si l'on découvre une politique meilleure pour un état, cette propriété prouve qu'il est possible de trouver une politique meilleure pour tous les états.

3.2.2.2 Définition

Une politique π^* est dite optimale si :

$$v^{\pi^*}(s) \geq v^{\pi}(s) \text{pour chaque } s \in S \text{ et chaque } \pi \in \Pi^{MS} \tag{3.1}$$

On appelle $v^*(s) = v^{\pi^*}(s) = max_{\pi \in \Pi} v^{\pi}(s)$ la fonction de valeur du MDP (et non plus d'une politique).

3.2.2.3 Politique markovienne déterministe

Soit un MDP $< S, A, T, R >$. Supposons une politique $\pi : H \rightarrow A$ histoire-dépendante stochastique ($\pi \in \Pi^{HS}$). Il existe une politique markovienne stochastique au moins aussi bonne que cette politique histoire-dépendante (preuve dans [Put94]).

De la même manière, parmi les politiques optimales, il existe une politique déterministe optimale.

En conséquence, on peut restreindre notre espace de recherche des politiques aux politiques markoviennes déterministes.

3.2.2.4 Lien vers l'architecture interne d'un agent

La propriété de Markov se révèle très intéressante. Dans des systèmes la vérifiant, les résultats précédents prouvent qu'il existe une politique markovienne optimale. L'architecture interne que

l'on s'est fixée initialement (agent réactif sans mémoire à court terme) suffit donc pour représenter la politique optimale.

3.2.2.5 Équation de Bellman

Supposons une politique markovienne stationnaire déterministe π. Cette politique a une fonction de valeur $v^{\pi}(s)$.

Cette fonction est solution unique du système d'équations linéaires [Put94] :

$$\forall s, v(s) = r(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') v(s')$$

La fonction de valeur de la politique optimale π^*, v^* est la solution unique du système.

$$v^*(s) = max_a \left(r(s,a) + \gamma \sum_{s' \in S} T(s,a,s') v^*(s') \right)$$

Inversement, à partir de cette fonction v^* il est possible de construire la politique optimale π^* :

$$\pi^*(s) = argmax_a \left(r(s,a) + \gamma \sum_{s' \in S} T(s,a,s') v^*(s') \right)$$

Il est à noter que la donnée seule de v^* ne suffit pas à déterminer la politique optimale. Il est en effet nécessaire de disposer, en plus de v^*, de la matrice de transition et de la fonction de récompense. Une fonction plus pratique est la fonction action-valeur. Comme elle constitue le fondement de l'algorithme Q-learning permettant de répondre au problème de l'apprentissage, nous la décrirons dans la partie 3.2.4.3 traitant spécifiquement de cette question.

3.2.2.6 Recherche de la politique optimale d'un MDP

Sans chercher à être exhaustif, plusieurs approches sont possibles pour résoudre un MDP en fonction des connaissances dont dispose l'agent :

- Si l'agent connaît un modèle du monde, il peut en tirer parti et planifier ses actions pour construire une politique optimale π^*.
- Si l'agent ne connaît pas de modèle du monde, il peut interagir avec son environnement pour acquérir de l'expérience et en inférer soit une politique optimale directement, soit un modèle du monde qu'il pourra réutiliser par la suite.

Les parties suivantes présentent différentes approches permettant de résoudre un MDP en fonction des données à la disposition de l'agent.

3.2.3 Résolution par Planification

On suppose que l'agent dispose d'un modèle du monde. En d'autres termes, l'agent connaît ou dispose d'une estimation de :

- la matrice de transition T
- la fonction de récompense R

A partir de ce modèle du monde, l'agent peut construire la politique optimale résolvant le MDP. Dans un environnement déterministe en horizon fini, lorsqu'on connaît l'état de départ, construire un plan consiste à déterminer la suite d'actions permettant de maximiser son critère

de performance.

Comme on se situe en horizon infini et dans un environnement stochastique, il est nécessaire de savoir quelle action entreprendre dans tous les états atteignables. La notion de politique répond effectivement à cette demande puisqu'elle associe à chaque état une action. La politique optimale obtenue permet donc de répondre de manière optimale à tous les événements et aléas qui peuvent se produire dans le système et ne nécessite donc pas de re-planification.

3.2.3.1 Value iteration

L'algorithme *'value itération'* (cf algorithme 2) est un algorithme de planification cherchant à calculer par itérations successives la fonction de valeur de la politique optimale v^* [Put94]. Une fois cette fonction de valeur calculée, il est possible d'en déduire une politique markovienne déterministe optimale (cf partie 3.2.2.5).

Algorithme 2 Value iteration

entree $\leftarrow MDP, \gamma, \epsilon$
Initialiser arbitrairement $V_0(s), \forall s \in S$
$t \leftarrow 0$
répéter
  $t \leftarrow t+1$
  pour tout $s \in S$ **faire**
    $V_t(s) \leftarrow max_a \left[R(s,a) + \gamma \Sigma_{s' \in S} T(s,a,s') V_{t-1}(s')\right]$
  fin pour
jusqu'à $max_s |V_t(s) - V_{t-1}(s)| < \epsilon$
$\forall s, \pi(s) = argmax_a R(s,a) + \gamma \Sigma_{s' \in S} T(s,a,s') V_{t-1}(s')$
retour $\leftarrow \pi$

3.2.3.2 Policy iteration

L'algorithme *'policy iteration'* (cf algorithme 3) consiste à construire une suite de politiques π convergeant vers la politique optimale π^* [Put94].

L'algorithme *'policy iteration'* se déroule en cycles de deux phases :

- la première phase propose d'évaluer la politique π à l'instant t avec l'algorithme *'policy evaluation'*
- la seconde phase construit une nouvelle politique plus performante en fonction de la fonction de valeur de la politique précédente

3.2.3.3 Bilan sur les techniques de planification

L'utilisation de techniques de planification se décompose en deux phases :

- la phase de construction de la politique nécessite une représentation du monde, de son environnement et de ses lois d'évolutions
- l'utilisation de la politique construite précédemment consistant à exécuter π et à émettre pour chaque état s rencontré l'action $a = \pi(s)$ donnée par la politique

Ainsi, même si la phase d'utilisation correspond à un agent réactif, la phase de construction des comportements nécessaire pour produire des comportements adaptatifs nécessite un modèle

Algorithme 3 Policy iteration

entree $\leftarrow MDP, \gamma$
Initialiser π et π'
répéter
  $\pi \leftarrow \pi'$
  pour tout $s \in S$ **faire**
    Calculer $V^{\pi}(s)$
  fin pour
  pour tout $s \in S$ **faire**
    $\pi'(s) \leftarrow argmax_a[R(s,a) + \gamma \Sigma_{s' \in nS} T(s,a,s') V_{\pi}(s')]$
  fin pour
jusqu'à $(V^{\pi'} = V^{\pi})$
retour $\leftarrow \pi$

du monde et s'oppose aux principes de simplicité que nous souhaitons mettre en œuvre. De plus, il n'est pas facile d'utiliser ces techniques dans des approches multi-agents puisque le modèle du monde perçu par un agent doit inclure les politiques des autres agents du système. Nous allons ainsi plutôt nous concentrer sur des techniques d'apprentissage qui permettent de s'affranchir de la connaissance d'un modèle du monde

3.2.4 Résolution par Apprentissage

3.2.4.1 L'apprentissage par renforcement

Problème La notion d'apprentissage par renforcement définit une classe de problèmes [SB98] : construire une politique maximisant le critère de performance à la suite d'un nombre répété d'expériences.

Ce problème peut être caractérisé de la manière suivante :
- L'agent ne connaît pas le modèle du monde (matrice de transition) et ne connaît pas la fonction de récompense
- Par contre, au cours de son existence, il interagit avec l'environnement : à l'instant t, le système se trouve dans l'état s_t. L'agent effectue l'action a_t, l'état de l'environnement est modifié (s_{t+1}) et l'agent reçoit la récompense r_t.
- Le tuple $xp_t = (s_t, a_t, r_t, s_{t+1})$ constitue une expérience élémentaire. La suite des tuples constitue l'ensemble des expériences de l'agent
- A partir de cette suite, l'agent doit pouvoir mettre à jour sa politique afin de maximiser son critère de performance individuel (dans ce cas la somme des récompenses pondérées)

On supposera en outre que l'agent évolue effectivement dans un MDP, c'est à dire que l'environnement est entièrement accessible et que ses lois d'évolution sont markoviennes.

L'apprentissage par renforcement est qualifié d'apprentissage semi-supervisé. Au cours d'un apprentissage supervisé, on donne au système un signal d'entrée et le signal de sortie que l'on souhaiterait obtenir et le système a pour objectif de faire correspondre au mieux la sortie qu'il fournit à la sortie exigée. Dans un apprentissage non-supervisé, l'agent ne reçoit pas de modèle des sorties attendues. Dans un apprentissage semi-supervisé, au cours de ses expériences, l'agent reçoit une note : la récompense lui donnant un aperçu de la pertinence de son action. Il peut

alors apprendre par "essai-erreur" à corriger son comportement pour adopter un comportement optimal sans aide exterieure.

Cette problématique est à rapprocher de la notion d'autonomie de Russel et Norvig. Pour eux, un agent est autonome s'il parvient à adapter ses règles comportementales à partir de son expérience. L'apprentissage par renforcement constitue une instanciation de ce problème générique d'adaptation, y répondre permet alors de proposer des agents dotés de capacités d'adaptation.

La question qui se pose désormais consiste alors à disposer d'un processus permettant de synthétiser les expériences de l'agent pour produire une politique π optimale au vu des connaissances de l'agent. La seconde question qui se pose est d'évaluer comment acquérir de l'expérience sur son environnement pour obtenir la politique optimale.

3.2.4.2 Origines des méthodes de résolution

Le problème d'apprentissage par renforcement est fortement lié aux travaux effectués en biologie expérimentale dans la première moitié du XX^{eme} siècle et consistant à analyser comment un organisme est capable de s'adapter à son environnement.

Deux approches expérimentales se sont intéressées à ce problème : le conditionnement pavlovien et le conditionnement opérant. Le conditionnement pavlovien s'intéresse à la manière dont un animal parvient à associer à des stimuli un autre stimulus générant des réponses physiologiques. Le conditionnement opérant porte sur la manière dont un animal parvient à modifier son comportement en fonction de stimuli extérieurs [Joz01].

[Joz01] présente les premiers travaux effectués par Sutton et Barto [SB98] comme fortement marqués par le modèle Rescorla-Wagner [RW72], issu du conditionnement pavlovien. Ce modèle d'apprentissage animal cherchait à expliquer comment il était possible d'expliquer l'apparition d'associations entre stimuli chez un animal. Des études du comportement animal ont en effet montré que lorsqu'un animal est soumis à un stimulus nommé stimulus conditionnel et que ce stimulus est suivi par un second stimulus (stimulus inconditionnel), l'animal parvient à émettre une réponse au stimulus inconditionnel dés qu'il perçoit le stimulus conditionnel. Le principe du modèle Rescorla-Wagner reposait sur le fait que l'animal essayait de prédire le stimulus inconditionnel sur la base du stimulus conditionnel et remettait à jour ses prédictions lorsque celles-ci n'étaient pas confirmées. Les extensions de ces modèles ont abouti à l'algorithme TD qui constitue un processus de calcul de fonction de valeur d'une politique par apprentissage dans les MDP [SB98].

[SB98] lie aussi a posteriori l'apprentissage par renforcement à l'apprentissage par essais-erreur d'E. Thorndike plus proche des expériences de conditionnement opérant. Lors d'un apprentissage essai-erreur, les actions ayant eu des conséquences bénéfiques vont avoir tendance a être ré-émises de manière plus fréquente dans le futur alors que les actions ayant eu des conséquences néfastes vont au contraire avoir tendance à être exprimées moins souvent. Cette tendance a été appelé par Thorndike d' 'effect law'.

Néanmoins, ces deux visions se rejoignent puisque de nombreux chercheurs ont tendance à croire que les processus d'apprentissage impliqués dans le conditionnement pavlovien et le conditionnement opérant sont identiques [Joz01], [Sig04].

Les algorithmes d'apprentissage par renforcement, sans être issus directement de modèles physiologiques précis, sont néanmoins inspirés par la psychologie animale. Cette remarque prouve la force des métaphores pour la construction d'algorithmes (idée que l'on retrouvera dans la partie suivante).

3.2.4.3 Q-learning

L'algorithme du Q-learning proposé par Watkins [WD92] constitue une partie de la réponse au problème de l'apprentissage par renforcement et se fonde sur la fonction état-action ou Q-valeur associée à chaque politique π.

La fonction Q-valeur d'une politique π est une fonction de $S \times A \rightarrow \mathbb{R}$. Pour un état s_0 et une action a_0 donnée, la Q-valeur $Q^\pi(s_0, a_0)$ représente l'espérance de gain de l'agent lorsqu'il se trouve dans l'état s_0, effectue l'action a_0 puis suit la politique π.

Cette fonction de Q-valeur est liée à la fonction de valeur v par l'équation :

$$Q^\pi(s,a) = R(s,a) + \gamma \sum_{s'} T(s,a,s') v^\pi(s')$$

La fonction de Q-valeur Q^* de la politique optimale π^* vérifie une forme particulière de l'équation de Bellman

$$\forall s \in S, \forall a \in A, Q^*(s,a) = E[r(s,a)] + \gamma \sum_{s'} p(s'|s,a).max_b[Q^*(s',b)]$$

Il est possible, à partir de la fonction Q^*, de construire très simplement la politique optimale π^* correspondante :

$$\pi^*(s) = argmax_a Q^*(s,a)$$

L'avantage des fonctions action-valeur (ou Q-valeurs) est qu'elles permettent de trouver la politique sans avoir recours au modèle du monde (contrairement à v).

Pour répondre au problème de l'apprentissage par renforcement, Watkins propose des méthodes pour calculer la fonction Q-valeur Q^* en interagissant avec l'environnement. L'algorithme Q-learning consiste à mettre à jour ces Q-valeurs à partir des expériences effectuées par l'agent. Pour l'expérience élémentaire n, $x_n = (s_n, a_n, s_{n+1}, r_n)$, les Q-valeurs sont mises à jour de la manière suivante :

$$Q(s_n, a_n) \leftarrow (1 - \alpha_n) Q(s_n, a_n) + \alpha_n (r_n + \gamma max_{b'} Q(s_{n+1}, b'))$$

où α_n désigne une suite de coefficients d'apprentissage caractérisant la vitesse d'apprentissage de l'agent. Plus α_n est important plus l'agent tire parti de sa dernière expérience et plus cette expérience (éventuellement au résultat biaisé) aura d'influence sur son comportement.

De plus, il a été prouvé que l'algorithme Q-learning converge presque sûrement vers Q^* à condition que [SB98] :

- chaque paire (état, action) (s,a) soit visitée un nombre infini de fois
- $\Sigma_n \alpha_n = \infty$ et $\Sigma_n \alpha_n^2 < \infty$
- $\gamma < 1$

Ainsi, l'avantage du Q-learning est de pouvoir apprendre une politique optimale π^* tout en suivant d'autres politiques et en s'assurant que chaque paire (état, action) peut être visitée un nombre infini de fois.

Algorithme 4 Q-learning

$\forall s, a$ Initialiser $Q(s, a)$
pour $n \leftarrow 0$ à $N - 1$ **faire**
  Effectuer une expérience ('choisir' a_n et s_n)
  (s_n, a_n, s_{n+1}, r_n) désigne le résultat de l'expérience
  $Q(s_n, a_n) \leftarrow (1 - \alpha_n)Q(s_n, a_n) + \alpha_n(r_n + \gamma max_{b'}Q(s_{n+1}, b'))$
fin pour
$retour \leftarrow Q$

3.2.4.4 Dilemme exploration/exploitation

Il reste une question à laquelle l'algorithme du Q-learning ne répond pas entièrement : il s'agit du dilemme exploration/exploitation. Les hypothèses de convergence de l'algorithme du Q-learning stipulent simplement que chaque couple (état,action) doit être visité un nombre infini de fois.

Problème posé La question qui se pose alors consiste à déterminer les trajectoires à suivre pour acquérir de l'expérience utile et à partir de quel moment, l'agent estime disposer d'assez d'informations pour arrêter d'explorer son environnement.

C'est un problème difficile car il faut s'assurer d'avoir effectué assez d'expériences pour
- avoir visité tous les états de l'environnement
- y avoir testé toutes les actions possibles
- mais surtout avoir une bonne connaissance de leurs conséquences.

En effet, le problème se pose lorsque l'environnement est stochastique. Dans ce cas, une expérience seule ne suffit pas pour déduire une loi générale. Par exemple, si un agent est face à un bandit manchot, ce n'est pas parce qu'il a gagné de l'argent en actionnant le bras que l'événement sera amené à se reproduire fréquemment. L'agent doit donc explorer son environnement pour acquérir le plus d'informations possibles et décider correctement des actions à entreprendre.

Cependant, une solution consistant à explorer intégralement l'environnement et à tester systématiquement toutes les possibilités un grand nombre de fois n'est pas satisfaisante. Une telle approche permettrait de déterminer la politique optimale mais cette phase d'exploration aura beaucoup coûtée à l'agent puisqu'il risque de subir un grand nombre d'échecs en cherchant à extraire des informations de ses expériences.

Exemple Prenons un agent dans un environnement très simple. Il est situé dans une cage et deux boutons sont à sa disposition. Un bouton sur la gauche et un bouton sur la droite.

L'état de l'environnement correspond à sa position (à gauche ou à droite de la cage) et l'agent peut effectuer plusieurs actions : il peut soit se déplacer dans l'autre direction de la pièce soit décider d'appuyer sur le bouton dont il est le plus proche. Lorsqu'il appuie sur un bouton, l'agent a une probabilité non nulle de recevoir une récompense positive.

Un agent peut avoir pris l'habitude d'appuyer sur le bouton gauche pour recevoir une récompense immédiate positive. Le dilemme exploration/exploitation se pose pour lui dans la mesure

où il doit choisir entre continuer à appuyer sur le bouton pour recevoir sa récompense ou aller explorer l'environnement (quitte à perdre du temps et à renoncer temporairement à une récompense positive facilement accessible) pour découvrir des récompenses plus importantes.

Politique stochastique Pour assurer le fait de visiter chaque paire (état,action) un nombre infini de fois, même si on cherche à construire une politique déterministe, la politique suivie pour générer des expériences se doit d'être stochastique. Pour chaque état s, la probabilité d'émettre une action a doit être non nulle. En outre, on souhaite suivre une politique permettant de se concentrer sur les zones intéressantes de l'espace d'état et permettant de recevoir des récompenses immédiates importantes au cours de l'exploration.

Une possibilité proposée est de suivre une politique ϵ-greedy [SB98]. Cette politique est stochastique, définie à partir des Q-valeurs de l'agent et dépend d'un facteur ϵ.

$$\pi(s) = \begin{cases} argmax_a(Q(s,a)) & \text{avec une probabilité de } 1-\epsilon \\ \text{aléatoire} & \text{avec une probabilité de } \epsilon \end{cases}$$

Il existe d'autres types de politiques permettant de répondre au dilemme exploration/exploitation. L'influence de leur utilisation sur les résultats obtenus par apprentissage ont été étudiés avec attention (cf [Thr92]). Pour des raisons de simplicité, nous nous limiterons à des politiques ϵ-greedy, faciles à mettre en œuvre et qui s'avéreront suffisantes pour mener nos expériences à bien.

3.2.4.5 Autres types d'apprentissage

On distingue plusieurs algorithmes d'apprentissage :

Les algorithmes d'apprentissage directs dont le Q-learning fait partie qui cherchent directement une politique et qui sont adaptés pour construire des comportements résolvant une tâche donnée a priori. Un certain nombre d'algorithmes d'apprentissage directs plus élaborés existent (TD, SARSA par exemple) mais nous ne les aborderons pas dans ce manuscrit, la technique du Q-learning étant suffisante pour valider notre approche (pour plus d'informations, voir [SB98]).

Un autre type d'algorithmes d'apprentissage existe : les algorithmes d'apprentissage indirects. Ils ont pour objectif d'apprendre un modèle du monde qu'il sera possible d'utiliser par la suite pour planifier ses actions ou pour accélérer l'apprentissage (comme Dyna-Q présenté dans [SB98]). Comme nous souhaitons des agents réactifs très simples, sans représentation du monde qui les entoure, nous n'aborderons pas ces techniques.

3.2.4.6 Bilan sur les techniques d'apprentissage

De la même manière que les techniques de planification, on peut séparer l'utilisation de techniques d'apprentissage en deux phases

- la phase d'apprentissage durant laquelle l'agent remet en cause certaines de ses règles comportementales à l'aide de ses expériences (et fait donc preuve d'autonomie). Durant cette phase l'agent modifie ses lois comportementales à partir d'une représentation simplifiée du monde (ses Q-valeurs) pour tenter d'atteindre un objectif.
- la phase d' exécution durant laquelle l'agent cesse d'apprendre et utilise les règles comportementales de type stimulus-réponse construites à la phase précédente. Dans cette dernière

phase, l'agent ne peut plus être qualifié d'autonome au sens de Russel et Norvig mais continue de l'être au sens de Wooldridge, c'est à dire une entité capable de prise de décision sans nécessiter l'intervention d'une aide extérieure. Durant cette phase, l'agent peut être considéré comme un agent réactif pur sans mémoire puisque ses actions sont entièrement déterminées par sa perception courante.

Afin de disposer d'agents dotés de capacités d'adaptation constante et d'autonomie au sens de Wooldridge, nous nous intéresserons dans notre approche à des agents en apprentissage constant pour lesquels la suite α_n ne converge pas vers 0.

3.2.5 Complexité

Les algorithmes présentés dans cette partie permettent de trouver les politiques optimales d'un MDP. L'utilisation de ces méthodes est par contre fortement limitée par la taille de l'espace d'états.

Un MDP constitue un modèle d'un problème donné et dans un monde discret, il est possible de se ramener à un tel cadre formel. Par exemple un problème discret dont les lois d'évolution dépendent du temps peut être représenté par un MDP (c'est à dire un système vérifiant la propriété de Markov) en définissant un nouvel espace d'états incluant le temps.

L'espace d'état peut souvent être très important et il n'est plus sûr
- qu'il soit possible de percevoir intégralement l'état du système
- qu'il soit possible de résoudre en un temps limité le MDP correspondant

[Par98] et [Gue03] proposent des méthodes permettant de réduire la complexité de ces calculs dans certains types de MDP appelé MDP faiblement couplés. Comme ces travaux sont très proches de ce que nous proposons dans ce manuscrit, la partie suivante se charge de les présenter.

3.2.6 MDP faiblement couplés

[Par98] s'intéresse à une classe de décomposition de problèmes relativement générale : celle où chaque sous-problème est 'faiblement' couplé avec les sous-problèmes voisins. En d'autres termes, les états reliant les sous-problèmes entre eux sont en nombre réduit. Par exemple, le problème d'un robot devant naviguer dans un bâtiment peut se décomposer en problèmes de navigation dans différentes salles.

Parr propose une approche constituée de deux étapes :
- la première étape consiste à construire pour chaque sous-problème un cache de politiques possibles. Pour notre exemple, il s'agira d'un ensemble de déplacements optimaux pour chaque salle en fonction des attentes possibles aux portes menant aux autres salles.
- la seconde étape consiste à agencer ces différentes politiques pour construire la politique finale optimale vis à vis du MDP initial. Il s'agit pour notre exemple de reconstituer le déplacement global à partir des déplacements optimaux dans chaque salle.

Un MDP faiblement couplé est défini comme un ensemble de sous-ensembles disjoints d'états $G_1, G_2, ..., G_n$. Chaque sous-ensemble définit un sous-problème (appelé région) et contient des états d'interface avec les autres sous-problèmes : les états accessibles à partir des autres sous-ensembles et des états permettant d'accéder aux autres sous-ensembles. Dans notre exemple, chaque pièce correspond à une région (cf figure 3.7) et les états d'interface sont représentés en

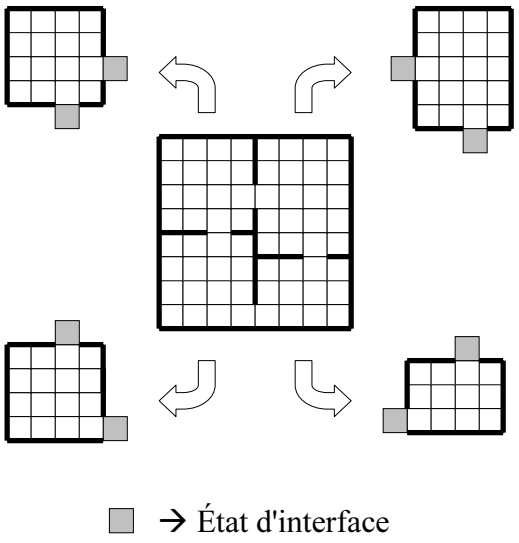


FIG. 3.7 – Décomposition d'un problème de navigation sous la forme d'un MDP faiblement couplés constitué de 4 régions

gris.

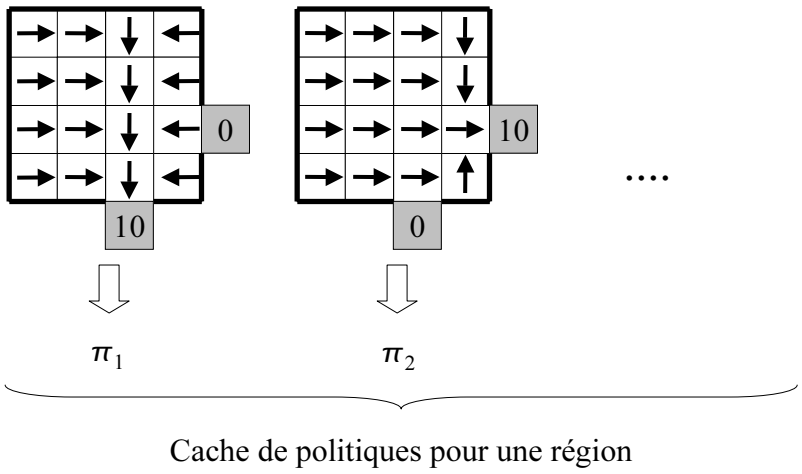


FIG. 3.8 – Construction d'un cache de politiques

Construire pour chaque région un cache de politiques consiste à déterminer pour toute valeur possible des états d'interface la politique optimale dans cette région. Il s'agit pour chaque valeur possible aux portes des salles de construire la politique optimale de déplacement dans cette salle. Si par exemple, la salle ne comporte que deux portes, les politiques optimales dépendront des fonctions de valeurs associées aux états des portes et consisteront à atteindre l'une, l'autre (cf figure 3.8) ou aucune d'entre elles (en supposant qu'il n'existe aucune autre source de récompense dans cette pièce). On sait que la politique optimale globale contiendra une de ces politiques

calculées. [Par98] propose ainsi des techniques efficaces pour construire ces politiques.

La seconde étape consiste à construire la politique optimale à partir des caches de politique. Parr définit ainsi un nouvel MDP. Dans cet MDP, chaque état correspond à une région et chaque action correspond à une des politiques du cache de la région. Il s'agit d'un MDP de plus haut niveau représentant le problème consistant à agencer les politiques des caches pour résoudre le problème global. La résolution de l'agencement des politiques potentiellement optimale peut alors se faire à l'aide des techniques classiques utilisées pour des MDP (value iteration, etc ...)

L'avantage d'une telle approche est de :

- limiter les recherches des actions pertinentes aux régions (de complexité moindre)
- effectuer un agencement pour construire la politique globale sans avoir besoin de reconsidérer l'ensemble des états.

Comme nous le verrons par la suite, de telles approches sont particulièrement intéressantes dans des cadres multi-agents car les systèmes collectifs ont souvent une structure particulière du fait de leur aspect distribué.

3.2.7 Bilan de la résolution d'un MDP

Dans cette partie, nous nous sommes concentrés sur les propriétés des politiques optimales définie dans les MDPs et les techniques permettant de les construire.

Le cadre formel des MDPs est particulièrement intéressant puisqu'il a été prouvé que dans ce cadre, la politique optimale peut s'exprimer sous la forme d'un ensemble de règles stimulus-réponse n'incluant aucune mémoire à court terme de l'agent ce qui correspond à l'architecture que l'on s'est fixée a priori.

Nous avons présenté différentes classes de techniques pour construire la politique optimale d'un MDP : les techniques de planification et les techniques d'apprentissage.

L'avantage de l'apprentissage réside dans le fait que l'agent n'a pas besoin de construire un modèle du monde mais simplement un modèle de ce qui est utile : les modèles de récompenses. Cette idée sera fondamentale dans les approches que nous envisagerons par la suite dans un cadre multi-agents (cf partie 6).

3.3 Résolution d'un POMDP

Un POMDP $< S, A, T, R, \Omega, O >$ permet de modéliser un problème de prise de décision dans lequel un agent n'a pas accès directement à l'état réel du système mais uniquement à une observation de celui. La problématique posée dans ce cadre consiste à construire un agent doté d'une rationalité limitée, c'est à dire agissant au mieux au vu des connaissances qu'il a à sa disposition.

3.3.1 Résolution exacte par planification

Lorsque l'agent dispose d'un modèle du monde et de la fonction d'observation, il peut tenter d'inférer l'état dans lequel il se trouve à partir de la suite des observations qu'il a pu percevoir.

Un moyen d'effectuer ceci est d'utiliser les états de croyance ('belief states').

Un état de croyance correspond à l'ensemble de l'information qu'un agent peut retirer des observations qu'il a pu recevoir. Il est défini par la densité de probabilité sur l'ensemble des états. Cette densité de probabilité donne pour chaque état la probabilité que l'agent soit effectivement dans cet état. L'évolution des états de croyance est markovienne ce qui permet d'utiliser des extensions des techniques précédentes basées sur la programmation dynamique appliquées au cas continu. Un algorithme possible est celui du witness qui fournit la politique optimale [Lit94c].

Mais l'algorithme est extrêmement complexe puisqu'il consiste à se placer dans l'espace continu des états de croyance et d'autres approches sont nécessaires.

3.3.2 Résolution par apprentissage

Lorsque l'agent ne dispose pas d'informations sur le monde (il n'a pas accès ni à T ni à O, ni à R), le problème consistant à agir optimalement dans un environnement partiellement observé est particulièrement difficile. En effet, l'agent ne peut savoir dans quel état il se trouve puisqu'une observation peut correspondre à plusieurs états et qu'inversement un état peut générer plusieurs observations.

On distingue ainsi deux approches :
- la première approche consiste à effectuer une recherche rapide tout en permettant d'obtenir une politique dont l'évaluation sera proche voire égale à la politique optimale parmi l'ensemble des politiques histoire-dépendant. Il s'agit d'un problème consistant à déterminer l'architecture interne d'un agent en lui ajoutant par exemple de la mémoire [DS03].
- la seconde consiste à chercher la politique parmi une sous-classe des politiques histoire-dépendantes comme les politiques markoviennes stochastiques qui associent à toute perception de l'agent une densité de probabilité sur les actions (cf [Lit94b]). Le problème d'optimisation doit alors être redéfini. De plus les performances des politiques sont fortement dépendantes du type de problème traité et des informations cachées par les perceptions courantes.

Ce dernier genre d'approche bien que basé sur une approximation est aussi utilisé dans des DEC-POMDPs comme nous le verrons dans la partie suivante (apprentissage incrémental cf [Buf03] par exemple) et parvient à construire des comportements intéressants si les apprentissages sont conduits avec précaution.

3.3.3 Bilan de la résolution d'un POMDP

Dans cette partie nous avons présenté la problématique de résolution d'un POMDP. L'objectif a été de montrer que :
- la complexité de la résolution exacte utilisant les états de croyance est grandement accrue
- la recherche de la politique optimale dans ce cadre est un problème complexe et nécessite de se placer dans l'espace des politiques histoire-dépendant
- que d'autres approches sont possibles mais nécessitent de se limiter à des problèmes particuliers ou de refuser à chercher la politique optimale dans l'absolu.

Ces considérations sont importantes, car il s'agit d'un des problèmes que nous rencontrerons par la suite puisque les systèmes multi-agents que nous voulons construire respectent un principe

de localité qui fait que les agents n'ont pas accès à la totalité de l'environnement.

3.4 Approches de résolution d'un DEC-POMDP

Notre objectif est de construire des systèmes multi-agents de manière automatique. Dans la partie 3.1 nous avons présenté les cadres formels DEC-POMDPs permettant de modéliser la problématique de construction d'un Système multi-agents réactif. Dans la partie précédente, nous nous sommes intéressés à la construction automatique du comportement d'un système constitué d'un agent réactif.

Dans cette partie, nous présentons quelques approches permettant de construire automatiquement les comportements des agents des systèmes multi-agents réactifs représentés dans le formalisme DEC-POMDP.

3.4.1 Organisation de cette partie

Cependant, avant d'entamer cette partie, il faut prendre quelques précautions.

Plusieurs approches sont envisageables pour construire les politiques individuelles. On distinguera :

- les approches **centralisées** que l'on présentera dans une première partie et pour lesquelles un processus central dispose de l'ensemble des informations et calcule la politique jointe afin de redistribuer cette politique parmi les agents.
- les **approches décentralisées** que l'on présentera dans une seconde partie pour lesquelles chaque agent, à partir de ses informations locales, cherche à construire ou mettre à jour sa politique individuelle.

Néanmoins, il faut garder à l'esprit que même si la résolution est centralisée, l'exécution reste décentralisée et implique des problèmes de coordination entre les comportements des agents.

Pour chacune de ces approches (centralisée/décentralisée) nous présentons les difficultés qui rendent la construction automatique des comportements des agents complexe. Nous nous contrerons sur les approches décentralisées qui sont les seules à être conformes au principe de localité que nous avons mis en avant dans la partie 2.

Enfin, les travaux traitant des DEC-POMDPs proposent divers formalismes. Chacun de ces formalismes décrit un point de vue sur les systèmes multi-agents en même temps qu'un ensemble de contraintes que doivent respecter ces systèmes. Il est difficile de faire la séparation nette entre modèle et processus de résolution puisque souvent ce processus s'appuie sur une spécificité du modèle. Par exemple, certains algorithmes se fondent sur le fait que la fonction de récompense peut se décomposer en somme de récompenses individuelles additives [BZLG03]. Ainsi, pour chaque travail, nous préciserons le modèle employé ainsi que les contraintes qui y sont liées.

On distinguera les propositions selon :

leur objectif , à savoir trouver une politique jointe optimale ou trouver une politique jointe sous-optimale

le modèle sur lequel s'applique le processus de résolution présenté

les contraintes respectées par le modèle utilisé

les moyens mis en oeuvre pour résoudre les problèmes posés : ces moyens sont très vastes et peuvent aller de la définition de sous-classes de problème à l'introduction de communication explicite.

Une liste plus complète des approches que l'on peut trouver dans la littérature est présentée en annexe à ce document (cf annexe A). Nous nous limiterons dans le coeur du manuscrit aux approches qui présentent un intérêt par rapport à la suite de nos travaux.

3.4.2 Approches centralisées

3.4.2.1 Description

Les approches de résolution centralisées se basent sur l'existence d'une entité chargée du contrôle de l'ensemble des agents. La politique jointe est calculée par cette entité qui redistribue ensuite les politiques individuelles aux agents. A l'exécution, chaque agent dispose alors de sa propre politique lui permettant de décider de manière autonome de l'action à effectuer.

Ces approches permettent de se ramener dans certaines conditions à un cadre mono-agent pour lequel il existe des algorithmes avec des preuves de convergence comme le 'value iteration' [SB98].

3.4.2.2 Avantages

Le fait de disposer d'un contrôleur global permet de répondre partiellement à plusieurs questions concernant le calcul de la politique jointe :

- comment évaluer la relation entre les comportements locaux et la fonction de récompense globale ? Cette question n'a plus lieu de se poser puisque le contrôleur dispose d'une vue globale de la récompense reçue par le système et peut lier cette récompense à l'action jointe qui a été émise par l'ensemble des agents.
- comment résoudre les problèmes de coordination entre agents ? Le contrôleur global ne s'intéresse pas à un agent particulier mais tente de construire directement la politique jointe. Il travaille alors sur l'espace des actions jointes et les problèmes de synchronisation entre agents sont résolus à ce niveau. Cependant, certains problèmes de coordination restent posés en raison de l'absence d'information complète au niveau de chaque agent à l'exécution du système.

L'intérêt d'une telle approche réside dans la possibilité de considérer les interactions entre agents. En effet, comme les interactions sont visibles au niveau de la matrice de transition définie de manière globale, disposer d'un contrôleur global permet de les considérer directement et de raisonner sur celles-ci implicitement en considérant les actions jointes.

3.4.2.3 Problèmes à résoudre

Complexité La première difficulté à laquelle la résolution de DEC-POMDP doit faire face réside dans la complexité du problème de construction d'une politique jointe.

Bernstein et al. se sont intéressés à la complexité de la résolution d'un DEC-POMDP constitué de deux agents dans [BGIZ02]. Le problème sur lequel ils se sont concentrés est le suivant ; "Étant donné un DEC-POMDP comprenant deux agents $< S, A_1, A_2, P, R, \Gamma_1, \Gamma_2, O, T, K >$, un horizon

fini T et un réel K, existe-t-il une politique jointe δ pour laquelle $V_\delta^T(s_0) > K$?" Ce problème a été prouvé NEXP.

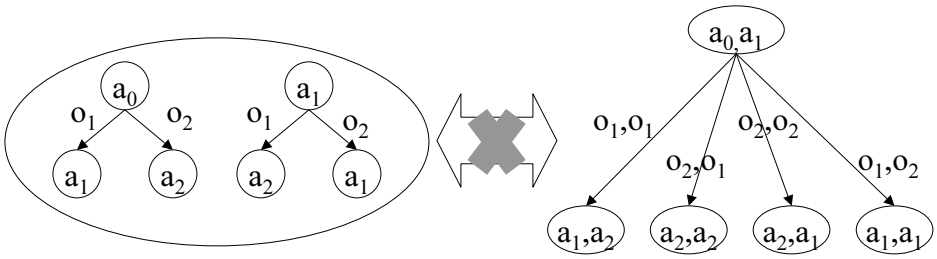


FIG. 3.9 – a)politique jointe, b)politique sur les actions et les observations jointes

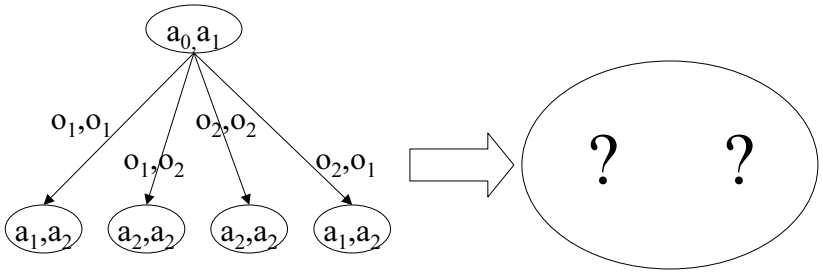


FIG. 3.10 – Problème de répartition d'une politique centralisée : Il n'y a pas de politique jointe correspondante : l'action choisie par l'agent 1 dépend uniquement de l'observation de l'agent 2

Origine de cette complexité Résoudre de manière centralisée un DEC-POMDP ou un DEC-MDP est un problème beaucoup plus complexe que la résolution d'un POMDP. En effet, dans un POMDP, la politique optimale est histoire-dépendante et associe à chaque historique une action. Dans un DEC-POMDP, la politique que l'on recherche est un tuple de politiques locales histoire-dépendantes, chaque politique locale se fonde sur les perceptions locales de l'agent concerné. Cette politique jointe n'est pas équivalente à une politique histoire-dépendante de l'historique joint (contenant observations jointes et actions jointes) vers les actions jointes (cf fig 3.9) telle qu'on pourrait la calculer en effectuant une centralisation complète [3]. En effet, une politique fondée sur les historiques joints serait inutilisable à l'exécution puisqu'elle nécessite que chaque agent dispose de l'ensemble des perceptions des autres agents à chaque pas de temps. Par exemple, la figure 3.10 présente une politique sur les observations jointes qu'il n'est pas possible de représenter sous la forme d'une politique jointe et qu'il n'est pas possible d'exécuter de manière décentralisée sans que les agents n'échangent de l'information.

Dans ces conditions, certains agents disposent à l'exécution d'informations que les autres agents n'ont pas. L'action qu'un agent va choisir n'est donc pas entièrement prédictible par un autre agent puisque même si les politiques sont construites de manière centralisée et partagées entre les agents, un agent ne peut savoir ce que l'autre agent a perçu et dans quelle branche de

[3] Ces politiques s'accompagnent en outre d'une explosion combinatoire du nombre d'états et d'actions en fonction du nombre d'agents

son arbre de politique histoire-dépendant il se situe.

Conséquences Du fait de cette complexité (déjà présente pour des systèmes constitués de deux agents), il est difficilement envisageable de chercher la politique jointe (qui n'est atteignable que pour des problèmes très simples). La problématique générale de construction d'un système multi-agents doit alors être reconsidérée sous un nouvel angle. Plusieurs approches peuvent alors être envisagées. Parmi celles-ci, on peut citer :

- celle consistant à se limiter à la recherche de solutions sous-optimales dans un temps raisonnable (en essayant de disposer de bornes concernant la qualité de la solution). Cela consiste à mettre en oeuvre des algorithmes permettant de trouver des politiques jointes sous-optimales en tirant éventuellement parti des caractéristiques du problème comme les approches proposées par Guestrin (cf [GKP01])
- celle consistant à caractériser des sous-classes de problèmes pour lesquelles il est possible de trouver la solution optimale relativement facilement en exploitant les propriétés de ces cas particuliers (comme les sous-classes des *transition-independant DEC-MDP* cf [BZL04])
- celle consistant à introduire des mécanismes ou de nouvelles capacités aux agents pour simplifier le problème de résolution. L'introduction de communication peut permettre l'échange d'informations entre agents et réduire dans certains cas la complexité du problème comme dans les COMMTDP (cf [PT02] et annexe A).

3.4.2.4 Approches abordées

Nous aborderons dans ce document trois approches de résolution centralisée :

- **les MMDPs** qui s'intéressent à des systèmes pour lesquels chaque agent perçoit l'état global du système. Ces approches s'opposent à notre principe de localité mais elles proposent des techniques permettant de synchroniser des politiques d'agents potentiellement optimales à partir d'apprentissage décentralisé. Elles laissent supposer que des techniques fondées sur ce type d'apprentissages peuvent être utiles.
- **les 'transition-independant DEC-MDPs'** et **les 'reward-independant DEC-MDP'** qui constituent des sous classes factorisées de DEC-MDP. Ces sous-classes sont basés sur une structuration particulière des lois d'évolution du système ou des fonctions de récompense afin de rendre compte des interactions entre les agents (au nombre de deux dans ces cadres). Cette structuration permet par la suite de construire des solutions optimales à moindre coût. Cette idée de représenter explicitement certaines interactions entre agents constitue un fondement de l'approche que nous proposerons par la suite.
- **les 'MDP factorisés'** qui utilisent une factorisation du MDP à l'aide de réseaux bayésiens pour construire des réponses approximatives à moindre coût. Cette approche sera plus détaillée dans la partie présentant les approches décentralisées car elle y trouve ses intérêts.

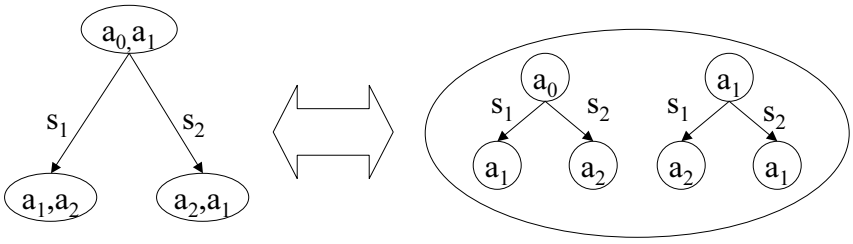


FIG. 3.11 – MMDP et politiques jointes

3.4.2.5 MMDP

Objectif	Construire une politique jointe optimale
Contraintes	Environnement accessible par chaque agent, récompense globale, lois d'évolution de l'environnement connues
Modèle	MMDP (Multi-agents Markov Decision Problem)
Moyen	Nouveau cadre formel, perception globale des agents, éventuellement capacités de communication pour se coordonner

Boutilier dans [Bou99] propose un modèle permettant de représenter des problèmes de décision multi-agents dans des systèmes entièrement coopératifs pour lesquels chaque agent perçoit l'état global du système : les MMDPs (Multiagent Markov Decision Problem).

Description Un MMDP est défini par un tuple $< \alpha, A_{ii\in\alpha}, S, Pr, R >$:
- α désigne le nombre d'agents du système
- chaque agent $i \in \alpha$ dispose d'actions individuelles décrites par A_i
- On définit $A = \times A_i$ l'espace des actions jointes.
- S désigne l'espace d'état
- Pr la matrice de transition. $Pr : S \times A \times S \rightarrow [0, 1]$
- R la fonction de récompense $R : S \times A \rightarrow \mathbb{R}$

Ce cadre peut être compris comme un DEC-POMDP pour lequel les agents ont une perception totale de leur environnement$(\forall i, s, O_i(s) = s)$. Cette propriété permet d'assimiler la recherche de la politique jointe à la recherche d'une politique sur les actions jointes. Une politique globale sur les actions jointes pourra en effet facilement être distribuée parmi les agents puisqu'ils connaissent l'état global du système (cf fig 3.11).

Résolution On suppose en outre que tous les agents connaissent le modèle du monde et des récompenses. Chaque agent peut alors résoudre individuellement ce problème par planification en supposant qu'il contrôle le comportement de l'ensemble des agents du système. Résoudre un MMDP est équivalent à résoudre un MDP $< S', A', T', R' >$ pour lequel :
- S' est égal à S
- A' correspond à l'espace des actions jointes
- T' correspond à Pr
- R' correspond à R

En utilisant les algorithmes présentés dans la partie 3.1.2, chaque agent peut donc construire une politique optimale $\pi^* : S \rightarrow A'$ comme s'il contrôlait intégralement les actions émises par tous les agents. Si tous les agents disposent du même modèle du monde, ils peuvent alors décider à chaque instant quelle action émettre en supposant que les autres agents aient effectués la même planification.

Néanmoins, puisque l'exécution est décentralisée, un problème de coordination se pose quand deux actions jointes ont la même valeur. Dans ce dernier cas, chaque agent hésite entre plusieurs actions jointes et la réponse individuelle qu'il doit fournir. Si les agents choisissent des actions jointes différentes, cela peut grandement nuire à la performance du système.

Par exemple, considérons deux agents qui doivent traverser un pont qui ne peut supporter que le poids d'un seul. On suppose en outre que pour ce problème, la politique jointe pour laquelle l'agent A avance avant l'agent B est autant avantageuse que celle pour laquelle l'agent B avance avant l'agent A. L'action que l'agent A doit émettre dépend de la politique jointe optimale que B va choisir de suivre et réciproquement. Il est donc nécessaire d'introduire des mécanismes de coordination pour répondre à la problématique de répartition de la politique jointe en politiques individuelles.

Boutilier avance plusieurs propositions pour effectuer le choix d'une action jointe commune quand il y a ambiguïté. Il propose :

d'utiliser des synchronisations par apprentissage A l'exécution, chaque agent évolue selon une certaine politique individuelle et apprend les réactions des autres agents lors des situations de conflit jusqu'à ce que l'ensemble des agents se synchronise sur une action jointe optimale qu'ils répéteront dans le futur.

d'utiliser des conventions Les agent connaissent des règles qui permettent de lever l'ambiguïté sur l'action jointe à choisir en cas de conflit.

d'utiliser de la communication Les agents peuvent s'échanger des messages pour se mettre d'accord sur l'action jointe à choisir dans les situations de conflit.

Avantages L'approche proposée par Boutilier permet effectivement de construire des politiques jointes optimales en s'appuyant sur les techniques développées dans le cadre mono-agent.

Inconvénients Néanmoins, utiliser de telles approches se heurte à l'explosion combinatoire de la taille de l'espace d'états et de la taille de l'espace des actions jointes par rapport au nombre d'agents, ainsi qu'à la nécessité de pouvoir observer l'état global de l'environnement (ce qui s'oppose au principe de localité présenté dans la partie 2)

Par rapport à nos travaux Une des propositions de Boutilier nous intéresse au plus haut point pour la suite de nos travaux : il montre que les approches centralisées ne sont pas suffisantes pour résoudre l'ensemble des problèmes de coordination même si chaque agent perçoit l'ensemble du système. Il propose surtout une approche consistant à résoudre ces problèmes en synchronisant les comportements des agents par apprentissage.

3.4.2.6 Sous-Classes de DEC-POMDP

Objectif	Recherche de politique optimale
Contraintes	Perceptions partielles, récompense globale
Modèle	'transition independent DEC-MDP', 'reward independent DEC-MDP'
Moyen	Se limiter à des classes particulières de DEC-MDP

Plusieurs travaux se sont intéressés à définir des sous-classes de DEC-POMDP à partir desquelles il devient possible de chercher des politiques optimales de manière efficace [GZ04]. Deux classes de DEC-POMDPs sont au coeur des travaux présentés dans [BZLG03] et [BZL04].

Ces travaux s'intéressent à des systèmes constitués de deux agents et supposent que l'espace d'état peut se factoriser à l'aide de deux sous-espaces $S = S_1 \times S_2$ et que chaque agent i peut observer s_i. Le système est donc un DEC-MDP, puisque les observations des agents permettent de reconstituer l'état global.

Un DEC-MDP factorisé $< S, A_i, T, \Gamma_i, O, R >$ à deux agents est dit *'transition indépendant'* s'il existe T_1 et T_2 vérifiant :

$$\forall s_2, a_2 : T(s'_1|(s_1, s_2), (a_1, a_2), s'_2) = T_1(s'_1|s_1, a_1)$$

$$\forall s_1, a_1 : T(s'_1|(s_1, s_2), (a_1, a_2), s'_2) = T_2(s'_1|s_1, a_1)$$

Un DEC-MDP factorisé $< S, A_i, T, \Gamma_i, O, R >$ est dit *'reward-independant'* s'il existe R_1 et R_2 vérifiant :

$$R((s_1, s_2), (a_1, a_2), (s'_1, s'_2)) = R_1(s_1, a_1, s'_1) + R_2(s_2, a_2, s'_2)$$

Exemple : Une flotte de robots devant explorer un environnement peut être modélisée par un DEC-POMDP 'transition-indépendant' : les actions entreprises par un agent n'ont pas d'influence sur le résultat des actions d'un autre agent et la matrice de transition T peut se décomposer en deux matrices T_1 et T_2. Un tel système n'est par contre pas *'reward-independant'* : si les robots effectuent les mêmes tâches et explorent la même partie de l'environnement, leurs récompenses seront moins importantes que s'ils partent explorer l'environnement dans des directions différentes.

[BZLG03] propose de résoudre des DEC-MDP transition-independant. La fonction de récompense est représentée comme la résultante de récompenses additives et d'un terme supplémentaire correspondant aux dépendances entre les actions des agents. Il propose un algorithme permettant de trouver la politique optimale jointe pour deux agents. Cet algorithme consiste à déterminer dans un premier temps, l'ensemble des politiques optimales π_A de l'agent A pour n'importe quel politique de l'agent B (nommé Coverage Set) en utilisant les propriétés de la fonction de valeur. Ensuite, à chaque politique π_A fixée du 'Coverage Set', est associée la politique optimale de B $\pi^*_B(\pi_A)$ qu'il est possible de calculer facilement. La politique jointe optimale se trouve alors parmi l'ensemble des couples $\pi^*_A, \pi_B(\pi^*_A)$.

[BZL04] propose d'autres types de résolution fondés sur le même genre d'approche : dans les *transition independant DEC-MDP*, la seule influence possible d'un agent envers un autre agent résidait dans les fonctions de récompense. Dans les DEC-MPD *'with event-driven interaction'*, les

transitions effectuées par un agent sont la résultante de transitions locales et de transitions dues aux événements effectués par les autres agents. Un agent peut ainsi faciliter les transitions des autres agents. L'algorithme repose sur le même principe que l'algorithme utilisé pour résoudre des DEC-MDP transition-independant : définir un ensemble de politiques optimales pour l'agent A pour toutes les politiques de l'agent B, associer à chacune de ces politiques, la politique optimale de B, chercher dans l'ensemble de ces politiques jointes, la politique jointe optimale.

Avantages En se focalisant sur des sous-problèmes particuliers des DEC-POMDP, ces travaux ont pu proposer les premiers algorithmes permettant de construire des politiques jointes de manière optimale à moindre coût.

Inconvénients Cette approche se heurte à l'explosion combinatoire du nombre d'états par rapport au nombre d'agents. Elle permet de construire des politiques optimales pour des groupes de deux agents en considérant des couples de politiques mais elle n'est plus envisageable dès que le nombre d'agents est supérieur à deux ce qui limite fortement son intérêt.

Par rapport à nos travaux Par contre l'idée sur laquelle se fonde cette approche est extrêmement intéressante. Cette approche utilise une structuration explicite du problème dû à la présence de plusieurs agents dans le système pour construire des résolutions à moindre coût. Elle isole ainsi les interactions possibles entre les agents et raisonne explicitement sur les couplages qui peuvent exister entre les comportements des agents.

3.4.2.7 MDP factorisés

Objectif	Recherche de politique sous-optimale
Contraintes	Observabilité partielle
Moyen	MDP factorisés
Modèle	Réseaux bayésiens

L'approche proposée par [Gue03] consiste à décrire le MDP sous la forme d'un réseau bayésien dynamique et à utiliser cette structure pour résoudre des MDP de grande taille de manière approchée en décomposant la fonction de valeur comme combinaison linéaire de fonctions de valeurs élémentaires définies sur cette structure. Cette approche permet de construire des systèmes de manière décentralisée et sera donc présentée de manière plus détaillée dans la partie suivante.

3.4.2.8 Bilan

En conclusion, l'approche centralisée permet de répondre en partie à notre problématique : elle parvient à construire les comportements d'agents réactifs dans certains cas particuliers mais se heurte à la complexité du problème de recherche en lui même.

Ces approches ont permis néanmoins de mettre en évidence un certain nombre de points intéressants :

- les approches centralisées ne résolvent pas tout et certaines approches fondées sur des apprentissages décentralisés permettent de résoudre des problèmes de coordination entre agents.
- il est possible d'exploiter la structure du problème et les interactions possibles entre agents pour faciliter la résolution comme dans [Gue03] [BZL04]. C'est cette dernière idée que nous souhaitons mettre en oeuvre dans le formalisme que nous avons développé et que nous présenterons par la suite (cf chapitre 5).

Dans notre cas, comme nous cherchons à faire des agents autonomes au sens de Russel et Norvig, c'est à dire capable de tirer parti individuellement de leurs expériences, les approches centralisées ne nous conviennent pas et il est donc nécessaire de se focaliser sur des approches de construction décentralisées.

3.4.3 Approches décentralisées

3.4.3.1 Description

Dans les approches décentralisées, chaque agent met à jour localement sa politique en fonction des informations dont il peut disposer. On cherche alors à savoir s'il est possible, à partir de mises à jour locales, de construire la politique jointe optimale ou tout au moins une politique jointe sous-optimale caractérisée par un critère de performance élevé.

La question fondamentale qui se pose dans le cadre d'approche décentralisée consiste à trouver des moyens permettant de considérer la présence d'autres agents dans le système afin de mettre à jour localement les politiques des agents ou en d'autres termes, comment intégrer une composante sociale à l'agent.

3.4.3.2 Avantages

Les approches décentralisées présentent un certain nombre d'intérêts :

- Les politiques sont plus facilement représentables dans ces systèmes puisqu'il s'agit de politiques locales. Une politique locale a besoin de considérer moins de variables, l'espace d'entrée de la fonction de prise de décision est plus réduit.
- De la même manière, elles permettent dans certains cas le passage à l'échelle et de construire des agents dans des systèmes ouverts : les agents sont définis par des politiques locales, et n'ont pas besoin de considérer l'ensemble des agents du système pour faire évoluer leur comportement. Ces politiques locales peuvent avoir un effet généralisateur permettant aux agents de se comporter de manière adéquate dans des systèmes composés de nombreux agents et dans des situations différentes mais localement semblables. Cet intérêt s'accompagne par contre d'une complexité algorithmique accrue quant à la recherche de la politique optimale ou d'une politique sous-optimale. Cependant, dans des applications réelles, il est possible, en sacrifiant la recherche de solution optimale, d'obtenir de bons résultats inaccessibles avec des approches centralisées.

3.4.3.3 Problèmes associés

Complexité La problématique de construction d'un système multi-agents dans un cadre centralisé a été prouvée NEXP-complet. Dans un cadre décentralisé, cette problématique est au moins aussi complexe puisqu'on dispose de moins d'informations lors de la construction des com-

portements individuels. Rechercher la politique optimale dans ce cadre reste alors inatteignable et nécessite de définir de nouveaux objectifs.

Observabilité partielle Une problématique à traiter est la problématique d'observabilité partielle déjà présente dans les POMDP et dans les approches centralisées. Un agent n'a pas accès instantanément à assez d'information sur son environnement pour pouvoir prédire avec certitude l'évolution future du système même s'il en contrôlait intégralement la dynamique. Cette problématique est encore plus présente ici. Dans une approche centralisée on disposait néanmoins des informations de l'ensemble des agents pour la construction des politiques locales alors que désormais, les mises à jour de ces politiques ne pourront se faire qu'à partir d'observations locales.

Co-évolution A ceci s'ajoute le problème de co-évolution des agents dû à la présence de plusieurs agents dans le système. Même si un agent a accès à l'état global S du système, il ne peut accéder aux comportements des autres agents. Les lois de l'environnement perçu par un agent qui inclut l'environnement du système et son environnement social sont alors non stationnaires puisque les autres agents sont libres de remettre eux aussi en cause leur comportement en s'adaptant à leur environnement et aux réactions des autres agents.

Exemple : Par exemple, si deux agents A et B se trouvent en face d'un pont qui ne peut supporter que le poids de l'un d'entre eux. Le comportement optimal de A (à savoir traverser le pont ou non) dépend du comportement de B et réciproquement. De plus, même si B a un comportement donné a priori, il peut changer celui-ci en fonction des réponses de A. Ainsi, si les deux agents sont des agents cherchant à plaire au mieux à l'autre, les mécanismes d'adaptation individuels peuvent se révéler contre-productifs.

Supposons par exemple que A ait un comportement consistant initialement à traverser le pont et que B ait initialement ce même comportement. Les deux agents A et B vont alors tenter de traverser simultanément le pont, celui-ci va s'écrouler et les agents vont recevoir une récompense négative. Les agents peuvent ensuite décider simultanément de changer leur comportement pour s'adapter au comportement de l'autre tel qu'ils ont pu l'observer. Ce changement simultané de politiques implique que les deux agents vont alors tous deux attendre que l'autre agent fasse le premier pas. S'ils modifient à nouveau simultanément leurs comportements, ils vont à nouveau tenter de traverser simultanément le pont. Ce processus de co-adaptation peut alors boucler.

Une autre conséquence de la co-évolution est la convergence possible vers des **Équilibres de Nash** pour lesquels augmenter la récompense globale nécessite de modifier simultanément les politiques de plusieurs agents. Supposons que deux agents A et B doivent appuyer simultanément sur deux boutons pour recevoir une récompense et qu'au départ aucun agent n'a ce comportement. L'agent A ne peut rien apprendre tant que l'agent B n'apprend rien et n'explore pas son environnement, puisqu'appuyer seul sur un bouton ne suffit pas à générer une récompense. Cette problématique est liée à l'exploration distribuée de l'espace des actions jointes.

Distribution des Récompenses Enfin, un dernier problème se pose vis à vis du lien entre les comportements locaux des agents et les récompenses associées au système. Deux cas sont à envisager :

1. soit la récompense est définie globalement et se pose le problème (souvent négligé) de sa perception par les agents et le problème du **crédit assignement**

2. soit la récompense s'exprime localement et se pose alors le problème de la **tragédie des communs**

Credit assignment Le problème du *credit assignment* se pose lorsque la récompense est définie au niveau global par une fonction $R : S \times A \rightarrow \mathbb{R}$. Un premier problème se pose vis à vis du principe de localité qui s'oppose à la perception d'une variable définie globalement par le système. Mais en outre, un second problème s'y ajoute : le problème du 'credit assignement'.

Le problème du 'credit assignement' provient du fait que les agents sont décrits à un niveau local et mettent leur comportement à jour à ce niveau alors que le problème à résoudre est décrit au niveau global par la notion de récompense.

Il devient alors nécessaire de répartir cette récompense pour savoir quel agent et quelle action est responsable de l'avancement de la résolution de la tâche. Ce problème nommé "credit assignmnent problem" [Wei99] peut se décomposer en deux sous-problèmes :

- un problème de 'credit assignement' inter-agent consistant à répartir les récompenses entre les agents
- un problème de 'credit assignment' intra-agent consistant à répartir les récompense entre les actions d'un agent.

Tragédie des communs Le problème de la tragédie des communs se pose lorsque la récompense est définie sous la forme d'une somme de récompenses individuelles additives. R peut alors s'écrire sous la forme $R = \sum_i r_i$ avec $r_i : S \times A \rightarrow \mathbb{R}$. Il est à noter que les récompenses individuelles r_i, bien que perçues localement par les agents, dépendent de l'action jointe et de l'état global du système. Ainsi, un agent qui effectue une action peut réduire ou augmenter les récompenses reçues par un autre agent.

Chaque agent i perçoit r_i mais ne parvient pas à percevoir les récompenses reçues par les autres agents. Ceci permet de répondre partiellement au problème du credit assignment : chaque agent est guidé par un signal de récompense local et peut mettre à jour son comportement en conséquence.

Néanmoins, il a été montré dans [Har68] que la maximisation individuelle des satisfactions locales peut conduire à la ruine du groupe. En d'autres termes, tenter de maximiser localement les récompenses reçues par chaque agent peut minimiser la somme des récompenses du système.

Exemple : Prenons trois agents A,B et C. Chaque agent correspond à un berger qui dispose d'un troupeau de moutons qu'il souhaite faire prospérer afin de vendre la laine récoltée. Ces agents ont décidé de mettre leur économies en commun pour acheter un champ dans lequel il feront paître simultanément leurs troupeaux. A chaque pas de temps, chaque agent a la possibilité d'augmenter la taille de son troupeau. Cette action a deux conséquences : une augmentation du gain de l'agent concerné (de 2) du fait de l'augmentation de la quantité de laine vendue et un coût partagé dans la communauté du fait que le mouton supplémentaire broute une partie de l'herbe du pré commun (chaque agent reçoit une récompense locale négative de -1). Ainsi, individuellement, chaque agent a intérêt à introduire un mouton dans le pré pour recevoir une récompense positive (de $r_i = 2 - 1 = 1$) mais si l'ensemble des agents agit de cette manière, chaque agent reçoit une récompense négative et le système est globalement déficitaire ($r_i = 2 - 1 - 1 - 1 = -1$ et $R = -3$). Des actions individuelles rationnelles peuvent ainsi conduire le système à la ruine en raison de la méconnaissance des agents sur les conséquences globales de leurs actions.

En outre, en plus du problème de la tragédie des communs, s'ajoute un problème proche du *credit assignment* : dans certaines circonstances, l'action d'un agent A ne génère pas de récompense immédiate mais peut permettre à un autre agent B d'atteindre un état particulier et de recevoir une récompense. Il est nécessaire de redistribuer une partie de cette récompense reçue par B à l'agent A responsable en partie de la récompense reçue par le système. Ceci peut se produire lorsque l'agent est doté de capacités particulières et effectue une opération pour l'agent B. Si A ne perçoit pas de bénéfice direct, il n'est pas incité à reproduire son action alors qu'en augmentant les récompenses reçues par B, il augmente la récompense globale reçue par le système.

3.4.3.4 Conséquences

Dans de telles circonstances, des planifications individuelles simultanées ne sont pas directement envisageables puisqu'un agent ne peut connaître le modèle du monde qu'il perçoit étant donné que celui-ci inclut les autres agents dont le comportement n'est pas donné a priori.

De la même manière, du fait de la tragédie des communs et de l'absence de possibilité de considérer les conséquences globales des actions d'un agent sur le système, des apprentissages simples effectués en parallèle sans précautions préalables ne peuvent construire des comportements collectifs pertinents et peuvent même conduire à la ruine du système.

Les approches qui suivent proposent alors des mécanismes permettant d'avoir des processus décentralisés conduisant à des comportements collectifs permettant de répondre en partie au problème.

3.4.3.5 Approches abordées

Au cours de cette partie, nous aborderons successivement :

- **l'utilisation de réseaux bayésiens** pour tirer parti de la structure du problème inhérente aux systèmes multi-agents et construire des réponses à moindre coût et de manière décentralisée à partir de communications et de techniques d'élimination de variables. Cette approche trouvera des échos dans notre proposition consistant à structurer le système à l'aide des interactions entre les agents basées sur des communications locales.
- **les techniques fondées sur l'empathie** qui cherchent à construire par planification itérative les politiques des agents. Bien que ces techniques permettent de construire des comportements à moindre coût, elles se heurtent à des équilibre de Nash dûs à la présence d'interactions entre les agents. Ces équilibres nécessitent de modifier simultanément les comportements plusieurs agents. Notre proposition consiste à proposer de formaliser explicitement certaines interactions pour pouvoir dépasser certains de ces minima locaux.
- **les fonctions de valeur distribuées** qui consistent à permettre aux agents d'échanger au cours de leur apprentissage leur Q-valeur et d'intégrer les Q-valeurs des agents voisins pour prendre en compte leur satisfaction dans les prises de décision. Bien que cette approche utilise des communications constantes entre agents, elle met en évidence des moyens de construire de comportements collectifs à moindre coût en considérant les satisfactions des autres agents du système. Notre approche sera basée sur une idée similaire.
- **certains MDP faiblement couplés** qui formalisent un problème d'allocation de ressource pouvant être interprété comme un problème multi-agent. L'intérêt de cette approche

réside dans la représentation explicite des interactions entre agents (consistant en des attributions de ressources communes) et l'utilisation d'heuristiques fondées sur des variables individuelles pour résoudre ces situations d'interaction. Notre approche proposera des heuristiques similaires.
- **Les techniques d'apprentissage incrémental** qui mettent l'accent sur la capacité de produire des systèmes collectifs à partir d'apprentissages décentralisés mais qui nécessitent du fait de l'absence de représentation des interactions entre agents de présenter aux agents de manière progressive des situations de plus en plus complexes afin qu'ils puissent apprendre progressivement des comportements joints utiles au système.

3.4.3.6 Utilisation de réseaux bayésiens

Objectif	Construire des politiques sous-optimales de manière peu coûteuse
Moyen	Exprimer la structure du MDP et en tirer parti pour construire des politiques sous-optimales
Contraintes	Observabilité partielle
Modèle	MDP factorisés et réseaux bayésiens dynamiques

Dans sa thèse, Guestrin [Gue03] propose de tirer parti d'une décomposition d'un MDP afin de construire à moindre coût une politique sous-optimale. Dans de nombreux problèmes, les états du systèmes correspondent à la concaténation de différentes caractéristiques définies dans le cadre du problème. De plus, ces caractéristiques n'ont pas forcément toutes des influences directes entre elles.

Afin de tirer parti de la structure du problème, Guestrin propose d'isoler ces caractéristiques pour obtenir une représentation factorisée de MDP inspirée des réseaux bayésiens dynamiques. Les états du système sont définis en fonction de variables X_i et la fonction de récompense et la matrice de transition peuvent s'exprimer de manière factorisée à partir des relations de dépendances entre ces variables.

[Gue03] propose d'utiliser des fonctions de valeur élémentaires définies sur des ensembles réduits de variables. Il s'intéresse particulièrement aux fonctions de valeurs $v_1, ...v_n$ définies sur les supports des relations de dépendance entre variables.

L'approche proposée par [Gue03] consiste alors à chercher une approximation de la fonction de valeur optimale définie sur l'espace d'état parmi les combinaisons linéaires de ces fonctions élémentaires $v(s) = \sum_i \alpha_i.v_i(s)$. En se plaçant dans ce sous-espace, la recherche de la fonction de valeur sous-optimale consiste à chercher les coefficients α_i.

Des techniques basées sur la programmation linéaire peuvent être utilisées pour effectuer cette recherche. En ayant choisi de décomposer la fonction de valeur dans l'espace défini par ces fonctions élémentaires, il est possible de représenter de manière compacte les équations d'optimalité de Bellman à résoudre. Enfin, en utilisant des techniques d'élimination de variables dans les réseaux bayésiens, la complexité de la résolution peut être réduite.

Cette approche parvient alors à trouver des solutions approchées (dont il est possible de borner l'erreur) dans des MDPs avec un nombre d'états très important [GKPV03]

Comme la résolution centralisée de systèmes multi-agents correspond à la résolution de MDP de grande taille et que les SMAs sont adaptés à une représentation factorisée, Guestrin propose d'utiliser ce même type de techniques pour effectuer de la planification [GKP01].

En tirant parti de la décomposition du problème et des techniques d'élimination de variables, il propose en outre des techniques d'apprentissage par renforcement distribuées [GLP02]. Ces techniques se fondent sur la présence du réseau de relation entre les variables modifiables par les agents. Ainsi il est possible, grâce aux techniques d'élimination de variables, de faire une maximisation globale à partir d'un transfert des fonctions de valeurs locales dans les réseau des relations inter-agents.

L'intérêt de ces approches résident dans le fait que la représentation factorisée du MDP correspondant explicite les relations entre les agents et les variables de l'environnement qu'ils peuvent modifier. De ce fait, les techniques d'élimination de variables peuvent être réalisées grâce à des communications locales entre agents.

Avantages Ces techniques permettent ainsi de limiter les communications entre agents aux communications utiles et de trouver la politique sous optimale à moindre coût à partir d'une structuration du problème tout en conservant des bornes concernant la qualité du comportement collectif construit.

Inconvénients Ces approches ne correspondent néanmoins pas entièrement à nos attentes :

- Les communications ne sont pas locales : bien que les canaux de communications utilisés soient toujours des canaux impliquant peu d'agents, il est supposé que les agents peuvent toujours échanger des informations et donc qu'il est toujours possible de reconstituer par communication l'approximation de la fonction de valeur globale
- Ces approches se heurtent au problème de l'expressivité des réseaux bayésiens dynamiques : le système est censé pouvoir se représenter sous la forme d'un réseau entre les différentes variables du système, en supposant cette représentation compacte. Cependant, si on s'intéresse à des systèmes complexes, une telle représentation est toujours possible mais n'est plus forcément compacte. Ainsi, si on s'intéresse à des système dans lesquels certaines actions (comme des échanges d'objets entre agents) nécessitent le respect de certaines distances entre les agents et des interactions impliquant des agents divers en fonction de leur position, ceci peut ajouter de nombreuses relations entre les noeuds du réseau bayésien puisque toutes les variables sont potentiellement en interaction les unes avec les autres.

Par rapport à nos travaux L'intérêt de ces travaux est de proposer une approche utilisant la structuration du système pour construire de manière distribuée les politiques des agents. Guestrin propose une approche consistant à partir d'un MDP factorisé pour lequel les interactions entre agents sont définies de manière explicite. Ceci permet alors de considérer les influences à long terme d'une action par un agent en évaluant l'ensemble des conséquences que peut avoir cette action sur le système, y compris sur les autres agents. Cependant, l'absence de localité dans les communications s'oppose au principe de localité que nous avons mis en avant.

3.4.3.7 Empathie

Objectif	Construire les politiques sous-optimales
contrainte	Récompense globale, Perceptions partielles
Moyen	Communication implicite permettant de connaître les politiques des autres agents, processus de synchronisation global
Modèle	DEC-POMDP

Un des problèmes rencontrés dans les systèmes multi-agents provient du fait qu'un agent doit connaître les comportements des autres agents et en être sûr pour pouvoir décider de la meilleure action à entreprendre. Une solution possible consiste à fixer les comportements des autres agents pendant qu'un agent planifie ses actions.

Les travaux présentés dans [Cha02] utilisent cette idée et se fondent sur la notion d'empathie. Chades propose de doter les agents de capacités de communication leur permettant de s'échanger leur politique. Chaque agent peut alors prendre en compte le comportement des autres agents dans sa prise de décision. De plus, si les comportements des autres agents restent fixes, un agent peut facilement planifier ses actions puisque les hypothèses de convergence des algorithmes classiques sont respectées étant donné que l'environnement perçu par un agent est alors stationnaire.

L'algorithme de co-evolution qui a été proposé dans [Cha02] consiste à choisir un sous-groupe d'agents et à fixer les politiques des autres agents. Il consiste ensuite à résoudre un MMDP pour trouver les politiques optimales du sous-groupe d'agents sélectionné. A l'itération suivante, un autre groupe d'agents est choisi et leur politique optimale calculée. Et ainsi de suite jusqu'à ce qu'aucun agent ne modifient leur politique au cours d'une itération. A chaque itération, les récompenses reçues par l'ensemble des agents augmentent puisque la phase de planification trouve au pire les politiques dèjà possédée par les agents au début de l'itération, ce qui prouve la convergence de l'algorithme.

Avantages L'avantage d'une telle approche est de disposer d'un algorithme de construction entièrement décentralisé et basé sur des planifications individuelles peu coûteuses pour construire des comportements permettant de résoudre un problème collectif.

Inconvénient Cependant, cet algorithme converge vers des équilibres de Nash. Il devient alors nécessaire de choisir le bon sous-groupe d'agents à faire évoluer pour résoudre cet équilibre, ce qui peut éventuellement en provoquer d'autres. De plus, ces approches nécessitent la communication des politiques entières entre agents et utilise un processus de synchronisation consistant à autoriser ou bloquer les phases de planification chez les agents, ce qui se heurte à notre principe de localité.

Par rapport à nos travaux Cette approche nous semble extrêmement intéressante car elle se fonde sur l'idée que des planifications individuelles peuvent permettre de construire des comportements collectifs. Néanmoins, le fait de synchroniser ces planifications, bien que permettant

d'avoir des preuves de convergence des politiques construites, ne conduit qu'à des sous-optimaux et se heurte aux systèmes qui doivent utiliser des interactions complexes entre agents, comme l'émission d'actions synchronisées.

3.4.3.8 Fonctions de valeurs distribuées

Objectif	Trouver une politique sous-optimale
Manière	Apprentissages décentralisés
Moyen	Introduire des communications entre agents pour échanger des fonctions de valeurs
Modèle	DEC-POMDP factorisé avec récompenses locales

Description [SWMR99] présente une approche pour effectuer de l'apprentissage par renforcement décentralisé (Decentralized Reinforcement Learning). Le point de départ de ces travaux consiste à s'intéresser tout d'abord à des apprentissages entièrement distribués. Chaque agent dispose de ses Q-valeurs et essaie d'apprendre une fonction de valeur comme s'il était seul dans l'environnement.

$$V_i(x) = max_{a \in A_i}(R_i(x,a) + \gamma \Sigma_{x' \in X} p(x'|a,x) V(x'))$$

Afin de prendre en compte les récompenses reçues par les autres agents, Schneider et al définissent une topologie et ajoutent un terme supplémentaire à cette équation : les récompenses pondérées (par un facteur $f(i,j)$) que transmettent les agents voisins à chaque agent.

$$V'_i(x) = max_{a \in A_i}(\Sigma_j f(i,j) R_j(x,a_j) + \gamma \Sigma_{x' \in X} p(x'|a,x) V'_i(x'))$$

Désormais, lorsqu'un agent apprend à maximiser cette nouvelle fonction V', il prend en compte de manière implicite les récompenses reçues par les autres agents et aura tendance à adopter un comportement utile pour les agents voisins. Simplement, la prise en compte des récompenses reste limitée aux voisins immédiats. L'approche proposée consiste alors à échanger non pas les récompenses mais les fonctions de valeurs.

$$V''_i(x) = max_{a \in A_i}(R_i(x,a_i) + \gamma \Sigma_j f(i,j) \Sigma_{x' \in X} p(x'|a,x) V''_i(x'))$$

Il est à noter que cette fonction V'' n'a plus le sens ni les propriétés des fonctions de valeur classique puisque elle ne vérifie plus l'équation de Bellman initiale.

Cette heuristique permet aux agents de prendre en compte la satisfaction des agents voisins qui intègrent à leur tour la satisfaction des voisins de rang supérieur. De cette manière, un agent considère de manière implicite l'ensemble des agents du système au moment de prendre sa décision. Appliquée à des problèmes de distribution, une telle approche fournit de bons résultats en se limitant à des communications entre agents voisins. Des perspectives sont envisagées concernant l'adaptation des paramètres $f(i,j)$ à l'exécution du système.

Avantages Cette approche présente l'avantage de construire des politiques collectives sous-optimales à partir d'apprentissages entièrement individuels et décentralisés. Elle se fonde sur l'idée qu'un simple échange des fonctions de valeur entre agents suffit pour qu'un agent puisse considérer la présence d'autres agents dans le système et leurs capacités par rapport à la tâche à résoudre. Cette approche constitue donc un moyen simple de doter les agents de compétences sociales au cours de leur apprentissage.

Inconvénients Les communications sont statiques et impliquent toujours les mêmes ensembles d'agents, or les interactions possibles entre les composants du systèmes doivent pouvoir se reconfigurer. De plus, il est supposé qu'il est toujours possible pour les agents de communiquer entre eux ce qui se heurte à nos contraintes de localité.

Par rapport à nos travaux L'approche employée par Schneider et al propose un moyen simple et économique d'intégrer une composante sociale au niveau de l'agent pour construire des comportements collectifs à partir d'apprentissages décentralisés. L'échange de fonctions de valeur est un moyen permettant aux agents d'adopter une attitude coopérative les uns avec les autres et de rendre compte des interactions à long terme entre les composants du système. Comme nous le verrons par la suite, l'approche que nous proposerons reposera sur le même postulat.

3.4.3.9 MDP faiblement couplés

L'article [MHK+98] porte sur la résolution de MDP faiblement couplé. Néanmoins, l'approche proposée peut être interprétée d'un point de vue multi-agents comme en témoigne l'exemple présenté et est trés proche du formalisme que nous développerons par la suite. Nous nous permettons donc de la développer dans cette partie.

Objectif	L'objectif est de construire une politique sous-optimale dans des problèmes à contraintes de ressources
Contrainte	Perception partielles, récompenses locales
Manière	Résolution décentralisée
Moyen	Communication implicite pour réajuster les politiques locales au sein du groupe.

[MHK+98] s'intéresse à un problème d'allocation de ressources entre plusieurs tâches. A chaque tâche est associée une fonction de performance indépendante et la fonction de performance globale du système est définie comme la somme des fonctions de performance locales. En outre, le MDP se décompose en états indépendants propres à chaque tâche. La résolution de chaque tâche peut ainsi être vue comme un sous-processus indépendant des autres (mais dépendant des ressources attribuées).

Interprétation multi-agents possible : Chaque tâche peut être vue comme assignée à un agent. Cet agent accède à la fonction de performance locale et cherche en fonction des ressources qui lui sont attribuées à maximiser cette fonction.

Résoudre un tel problème est non trivial puisque :

- les politiques locales dépendent des ressources attribuées
- les ressources attribuées dépendent des politiques locales qui en tirent parti et des perfomances associées.

L'exemple proposé est un problème de distribution d'armement dans une flotte aérienne pour laquelle chaque avion a une cible fixée a priori. Les contraintes sont de deux ordres : des contraintes globales concernant le stock global d'armement et des contraintes locales liées à la capacité de chaque avion.

[MHK+98] propose de calculer les politiques locales individuellement (comme pourrait le faire chaque agent de manière décentralisée si on garde à l'esprit notre interprétation) et d'utiliser des heuristiques pour distribuer les ressources entre les tâches :

- Tout d'abord en cherchant à maximiser la somme des performances locales attendues tout en respectant la contrainte globale. Chaque ressource est attribuée de manière itérative en fonction de son utilité marginale, c'est à dire du gain que peut apporter cette ressource pour la performance de la tâche considérée. Cette ressource est attribuée à la tâche pour laquelle le gain marginal est le plus important.
- En vérifiant ensuite si les contraintes locales sont respectées et en redistribuant les excès lorsqu'une contrainte locale n'est pas respectée.

Avantage Chaque agent détermine sa politique de manière individuelle entièrement décentralisée et une phase de centralisation fondée sur une heuristique permet d'ajuster les ressources attribuées entre agents.

Inconvénient Il y a échange d'information entre toutes les 'entités du système' pour déterminer comment distribuer les ressources.

Par rapport à nos travaux Cette approche se fonde sur l'explicitation des interactions entre les composants du système : le système est décrit sous la forme de processus locaux chacun associé à une tâche particulière et les interactions possibles entre ces processus résident dans l'attribution de ressources communes. En structurant le système de cette manière, il est possible de construire des réponses collectives (attribution des ressources et comportement associé à chaque tâche) à partir de résolutions locales et d'heuristiques permettant de décider de l'attribution des ressources en fonction des performances locales qui ont pu être calculées.

3.4.3.10 Apprentissage incrémental

Objectif	L'objectif est de construire les politiques sous-optimales
Contraintes	Perceptions partielles, Récompense globale
Manière	Résolution décentralisée, Exécution décentralisée
Moyen	Guider les agents et proposer des situations de plus en plus complexes aux agents.

Cette approche proposée par Buffet [Buf03] cherche à construire de manière décentralisée des politiques individuelles d'un ensemble d'agents dotés d'observations partielles. Il existe deux types d'agents et leur tâche consiste à fusionner des blocs de couleurs différentes (jaune et bleu).

Chaque agent perçoit localement le bloc jaune et le bloc bleu les plus proches ainsi que l'agent de l'autre type. La précision des perceptions dépend de la distance de l'agent aux autres objets : si l'agent est à une case de l'objet, il peut percevoir sa position exacte. Si celui-ci est plus éloigné, il en perçoit la direction. Chaque agent reçoit une récompense lorsqu'il participe à la fusion de bloc.

Bien qu'il ne soit pas possible de générer par apprentissage les politiques optimales des agents, l'apprentissage incrémental cherche à construire "de bonnes politiques". Il consiste à proposer à deux agents des situations de plus en plus complexes et à leur faire apprendre ces situations à partir de leurs apprentissages précédents. Les agents sont guidés par le concepteur de l'application et parviennent :

- à se coordonner à partir des signaux de récompenses reçus et des perceptions partielles
- à construire des politiques permettant de résoudre des situations élémentaires

Ces solutions possèdent en outre de propriétés de passage à l'échelle. Une fois les politiques individuelles apprises, on peut disposer dans l'environnement un certain nombre de cubes et d'agents et parvenir à des fusions de blocs. Le fait de disposer de politiques stochastiques et de perceptions partielles constitue un moyen de généralisation des comportements des agents.

Inconvénients Cependant, la présence d'une entité extérieure est nécessaire pour guider les apprentissages en donnant des situations de plus en plus complexes. En outre, la récompense du système est globale et observée par l'ensemble des agents.

Par rapport à nos travaux Cette approche se fonde sur un apprentissage des interactions possibles entre agents. Elle n'utilise pas de structuration a priori du système mais permet aux agents d'apprendre à résoudre les situations d'interactions en leurs présentant des situations de plus en plus complexes. Un apprentissage simultanée (contrairement à [Cha02]) permet de construire des solution collectives du fait de la présence d'un signal de récompense global qui renforce les actions jointes utiles qui ont pu être émises.

3.4.3.11 Bilan

De nombreuses approches basées sur des modèles et des contraintes diverses ont été proposées. Des approches décentralisées parviennent à construire des politiques individuelles générant un comportement collectif avec un critère de performance élevé malgré la complexité importante du problème.

Plusieurs aspects nous semblent intéressants et interviendront dans le formalisme que nous présenterons par la suite :

- [Buf03] a prouvé qu'il est possible d'apprendre à des agents à se synchroniser et à se coordonner à condition que ces apprentissages soient correctement guidés.
- [Gue03] a montré que l'utilisation de la structure d'un problème inhérente au système multi-agents peut permettre de construire des solutions approchées de manière décentralisée à moindre coût.
- [MHK+98] montre qu'il est possible de construire des systèmes à partir de construction de politiques individuelles et de prises de décision centralisées impliquant l'ensemble des agents (partage de ressources dans ce cas).

3.5 Bilan du chapitre

3.5.1 Synthèse du chapitre

Afin de disposer de méthodes génériques pour construire automatiquement des systèmes répondant à des problèmes collectifs, nous avons choisi de nous concentrer sur des approches formelles. De telles approches nécessitent tout d'abord un cadre formel permettant de représenter un système multi-agents, constitué de plusieurs entité en interaction, le problème qui y est lié ainsi que les comportements de ces entités afin de pouvoir les manipuler. L'objectif qui se pose ensuite est de fournir des algorithmes permettant de construire de manière automatique les comportements d'agents rationnels autonomes.

Nous avons ainsi été amené à considérer les modèles markoviens et leurs extensions. Nous avons donc présenté les MDPs qui formalisent un problème de prise de décision mono-agent puis nous nous sommes attardés sur les DEC-PODMPs. Ce cadre formel correspond effectivement à nos attentes, car il permet de représenter à la fois des systèmes constitués de plusieurs agents réactifs autonomes en interaction, les réponses de l'environnement aux actions multiples des agents ainsi qu'un problème collectif par l'intermédiaire d'une fonction de récompense globale.

Nous avons en outre analysé comment le concept d'interaction est instancié dans les DEC-POMDPs : seules les interactions indirectes entre agents sont possibles et ces interactions sont définies dans la matrice de transition au niveau global.

Afin de répondre à notre problématique initiale, nous nous sommes intéressés aux approches permettant de construire automatiquement des comportements dans ce formalisme. Alors qu'il existe des solutions exactes permettant de construire le comportement d'un agent à partir d'un MDP, le problème posé dans le formalisme DEC-POMDP est trop complexe pour pouvoir disposer d'algorithmes construisant une solution exacte utilisables en pratique.

Deux types d'approches sont néanmoins proposées : les approches centralisées et les approches décentralisées.

Les approches centralisées permettent de se placer à un niveau dans lequel il est possible de considérer la matrice de transition du système et les interactions qui y sont représentées. Cependant, ces approches se heurtent au problème d'explosion combinatoire et ne permettent pas de construire des agents autonomes au sens de Russel et Norvig, c'est-à-dire capables de tirer parti de leur expérience pour s'adapter.

Les approches décentralisées quant à elles doivent permettre à un agent de considérer la présence d'autres agents dans le système alors que les interactions sont définies au niveau global. Ces approches cherchent à introduire et à expliciter la notion d'interaction pour en tirer parti

- en guidant l'apprentissage des agents pour leur faire apprendre les interactions de manière incrémentale [Buf03], [Cha02]
- en structurant le système pour mettre à jour les relations entre agents et en tirer parti [Gue03], [MHK+98]

Les approches décentralisées sont néanmoins elles aussi basées sur une centralisation qui s'oppose à notre principe de localité mais qui semble nécessaire puisque l'interaction apparaît seulement à ce niveau. Cette centralisation peut avoir lieu au niveau des approches de construction (approche centralisée ou apprentissage guidé par le concepteur), au niveau des communications

(qui ont lieu à tout instant dans tout l'espace et entre tous les agents) ou au niveau des informations partagées (échanges des perceptions, connaissance des comportements complets des autres agents,etc ...)

3.5.2 Pour une formalisation de l'interaction

Parmi ces travaux, certains s'intéressent à utiliser la structure particulière du problème pour obtenir des solutions à moindre coût (comme les travaux de Guestrin [Gue03] ou ceux de Meuleau [MHK+98]). Nous pensons que dans beaucoup de problèmes, les agents ne sont pas constamment en interaction. Il n'est pas nécessaire de raisonner constamment au niveau global même si des décisions impliquant un nombre réduit d'agents sont nécessaires pour obtenir des solutions satisfaisantes. Pour un problème donné, dans certaines circonstances, il est possible de prendre une décision sans avoir besoin de considérer les autres agents du système, cette partie du problème peut alors s'approcher d'un problème mono-agent que l'on sait résoudre dans un cadre markovien.

Notre approche va donc chercher à utiliser une structure inhérente aux systèmes multi-agents pour trouver un compromis entre des approches centralisées qui doivent faire face à l'explosion combinatoire du nombre d'états, et les approches décentralisées qui nécessitent d'introduire des compétences sociales au sein de l'agent. Nous avons montré dans la partie 2, que c'est le concept d'interaction qui permet de produire un comportement collectif à partir de comportements individuels. Nous pensons donc que c'est l'interaction qui va permettre de structurer naturellement les problèmes que l'on souhaite résoudre.

Comme nous souhaitons construire des systèmes de manière décentralisée, il faut que l'agent puisse disposer d'une structure du problème puisque c'est à son niveau que les comportements vont être calculés et remis en cause. Nous proposons donc de formaliser le concept d'interaction **au niveau de l'agent** pour en faire une entité manipulable par celui-ci. L'interaction permettra alors de distinguer les décisions individuelles des décisions collectives. Cette formalisation de l'interaction disponible à l'agent sera alors la structure à partir de laquelle il pourra intégrer une composante sociale. Elle permettra ainsi d'élaborer des comportements collectifs à partir de processus d'adaptation individuelle.

Ce concept d'interaction recouvre de nombreuses acceptions, mais nous avons classé les moyens permettant d'effectuer les couplages des comportements entre agents en deux catégories : les interactions indirectes et les interactions directes (cf partie 2.2.3.4). L'interaction indirecte (qui apparaît dans le formalisme DEC-POMDP) est non dirigée, elle est émise dans l'environnement et ses conséquences dépendent de l'état global du système ainsi que de sa dynamique. Par exemple, un agent déposant un objet dans l'environnement n'est pas certain de pouvoir modifier le comportement d'un agent spécifique puisqu'il faut que ce dernier agent ait un comportement l'amenant à percevoir l'objet déposé. Cette caractéristique de l'interaction indirecte rend difficile l'évaluation de ses conséquences au niveau local.

Nous proposons donc plutôt de nous concentrer sur la formalisation du concept d'interaction directe et de l'intégrer dans un cadre markovien. Cette formalisation permettra alors de mettre en relation des agents donnés pour prendre des décisions collectives basées uniquement sur des communications locales et ne s'opposant pas à nos contraintes de localité.

Il reste à instancier ce concept d'interaction directe dans le système. Ceci consiste à :
- trouver la manière de représenter l'interaction directe dans un système markovien
- trouver des règles d'adaptation décentralisées pour en tirer parti
- trouver comment ces mécanismes peuvent permettre de construire des comportements collectifs à partir de décisions individuelles

Pour ce faire, nous avons choisi de reconsidérer à leurs sources les techniques d'apprentissage par renforcement, originellement inspirées de la psychologie animale (cf partie 3.2.4.2). Il semble alors légitime de voir s'il n'existe pas d'autres systèmes naturels qui présentent des mécanismes d'adaptation mais cette fois-ci collectifs et fondés sur la notion d'interaction directe. Ils pourraient alors constituer un point de départ permettant de nous guider pour de nouvelles techniques d'apprentissage collectif.

Cette démarche a pour objectif de comprendre comment la nature à pu répondre à un certain nombre de paradoxes apparents afin de trouver des réponses aux questions suivantes posées dans le cadre des SMAs :
- comment l'interaction directe peut elle être mise en œuvre dans un système multi-agents pour lequel toute initiative est d'origine individuelle ?
- comment intégrer une composante sociale et prendre en compte le comportement d'autres agents à partir d'agents réactifs disposant de capacités cognitives réduites et sans représentation complexe ?
- comment adapter localement les comportements pour permettre de construire une solution collective plus performante que la simple juxtaposition de comportements individuels ?

Cette démarche est présentée dans le chapitre suivant.

Chapitre 4

Inspiration biologique

Dans la partie précédente, nous avons présenté les modèles inspirés des processus décisionnels de Markov et nous avons mis en évidence la manière dont l'interaction y est représentée et les conséquences que l'absence de représentation explicite de l'interaction au niveau individuel pose quant à la construction des comportements de systèmes multi-agents.

Nous en sommes arrivés à la nécessité de formaliser le concept d'interaction dans un cadre markovien pour que les agents puissent raisonner sur la présence d'autres agents dans le système et puissent intégrer une composante sociale. Il reste ainsi à trouver une représentation possible du concept d'interaction et des mécanismes d'adaptation permettant d'en tirer parti. La question qui est posée est identique à une des questions posées par Ferber à savoir quelle forme d'interaction est-il intéressant d'introduire dans le système.

De la même manière que l'apprentissage par renforcement est issu de mécanismes d'adaptation individuels, notre objectif a été de considérer un mécanisme d'adaptation collectif et de comprendre dans quelle mesure il peut être une source d'inspiration pour la construction de processus d'apprentissage collectif. Nous nous sommes ainsi intéressés à des phénomènes collectifs naturels afin de nous guider selon une démarche ascendante pour instancier la notion d'interaction directe et pour proposer des processus adaptatifs permettant d'en tirer parti.

Pour cela, nous avons cherché un phénomène biologique pour lequel :
- les individus sont en interaction les uns avec les autres
- ces interactions permettent l'apparition de comportements collectifs qualitativement différents des comportements individuels
- les individus savent répondre individuellement au problème posé mais exhibent des comportements différents en fonction de leur contexte social
- le comportement collectif peut être compris comme ayant une fonction pour le groupe

Une expérience consistant à confronter des rats à des contraintes environnementales d'accès à la nourriture présente ces caractéristiques :
- Des mécanismes d'adaptation locale ont pu être mis en évidence
- Les comportements individuels adoptés par les individus dépendent de leur environnement social.
- Ces comportements conduisent à une réponse collective robuste qui s'exprime par l'apparition d'une spécialisation dans le système biologique.

Nous avons ainsi cherché à comprendre les mécanismes existants et les lois d'adaptation

permettant à ces goupes animaux de s'organiser alors que les individus ne disposent que de perceptions partielles et doivent faire face à des environnements incertains. Les éthologues se posent les mêmes questions et disposent en outre d'un certain nombre de réponses et d'hypothèses sur le fonctionnement de ces groupes issus d'expérimentations.

Les questions que nous nous sommes posés conjointement avec les éthologues consistaient alors à comprendre :

- comment un individu pouvait **intégrer son contexte social** afin de comprendre **quelles formes d'interactions** peuvent permettre aux agents de prendre des décisions sociales.
- quelles **capacités cognitives** pouvaient être impliquées dans ces interactions afin de choisir la meilleure **architecture interne** à fournir aux agents agents.
- comment les **règles d'adaptation locales** couplées aux interactions permettent de construire un comportement collectif robuste et dans quelle mesure un **mécanisme d'adaptation** peut permettre de construire des comportements collectifs à partir de lois de construction de comportements individuels.

Ces questions constituent le pendant biologique des questions que nous nous sommes posées à la fin du chapitre précédent.

Enfin, afin de valider les hypothèses émises par les éthologues et de construire une première implémentation de ces concepts, nous avons proposé un modèle original : le modèle Hamelin chargé de reproduire le phénomène de spécialisation biologique observé dans des groupes de rats. Il constituera notre source d'inspiration pour le cadre formel que nous proposerons dans la partie suivante (cf chapitre 5).

Dans un premier temps, nous décrirons le domaine de l'éthologie qui s'intéresse à l'étude des comportements animaux et les liens qui peuvent unir l'éthologie aux systèmes multi-agents. Nous présenterons ensuite les résultats des éthologues du laboratoire de neuro-sciences comportementales de UHP Nancy 1 sur les phénomènes de spécialisation observés dans des groupes de rats ainsi que leurs hypothèses. Enfin, nous détaillerons le modèle Hamelin qui simule ce phénomène et nous nous attarderons sur les mécanismes qui y sont mis en oeuvre et que nous réutiliserons par la suite.

Au cours de la description des modèles à l'origine du modèle Hamelin, un parallèle sera fait avec notre démarche consistant à introduire des interactions directes dans les modèles markoviens puisque nous serons amenés à faire de même dans ce chapitre.

4.1 La démarche d'inspiration biologique

4.1.1 Éthologie

4.1.1.1 Définition

L'objet d'étude de l'éthologie est le comportement animal. L'éthologie est définie comme l'étude comparée des comportements animaux" (selon Lorenz cité dans Picault [Pic01] page 17).

Cette étude se fonde sur l'observation des activités animales et la quantification d'ensembles de réactions objectivement observables. Ce domaine cherche à construire des schémas explicatifs

des comportements. Ces schémas explicatifs vérifient un axiome fondamental : le canon de Morgan ou principe de parcimonie.

Cet axiome stipule : "en aucun cas, nous ne pouvons interpréter une action comme la conséquence d'un exercice ou d'une faculté psychique plus haute, si elle peut être interprétés comme l'aboutissement d'une faculté située plus bas dans l'échelle psychologique" (cf [Pic01]). Selon cet axiome, parmi un ensemble d'explications possibles, il faut retenir les explications les plus simples en terme de faculté psychologique et les moins nombreuses. L'axiome de Morgan implique que ces dernières explications possèdent un pouvoir explicatif plus important jusqu'à ce qu'elles soient mises en défaut.

Il reste à préciser ce qu'on entend par l'explication de l'émission d'un comportement. Tinbergen un des fondateurs de l'éthologie insiste sur le fait qu'il existe plusieurs manières de répondre à la question 'pourquoi un comportement ?' en éthologie. Cette remarque est connue sous la dénomination "des 4 questions de Tinbergen" [KD93].

4.1.1.2 Questions de Tinbergen

Donner une explication à l'émission d'un comportement peut être envisagée selon quatre approches distinctes :

- On peut s'intéresser à l'aspect **fonctionnel** du comportement : fournir une réponse consiste alors à expliquer le comportement en terme de fonction adaptative pour l'animal, ou en d'autre terme quelle peut être l'utilité de ce comportement pour l'animal.
- On peut s'intéresser à l'aspect **causal** *et aux* **mécanismes** mis en oeuvre : fournir une explication consiste alors à analyser les causes des comportements, les stimuli déclencheurs et les réactions physiologiques mises en oeuvre.
- On peut s'intéresser à l'aspect **ontogénétique** : fournir une réponse consiste à expliquer dans quel mesure l'animal a pu acquérir ce comportement au cours de son développement.
- On peut s'intéresser à l'aspect **phylogénétique** : fournir une réponse consiste alors à s'intéresser à l'apparition de ce comportement au sein de l'espèce concernée (et non pas au sein de l'individu) et au rôle que ce comportement a pu avoir dans l'évolution de l'espèce.

Tinbergen précise qu'il est nécessaire de faire une distinction entre ces différentes questions sous peine d'engendrer des débats stériles sur la prééminence d'une explication par rapport à une autre. Ces questions se poseront dans nos travaux et méritent d'être explicitées dès maintenant.

Un courant de pensée de l'éthologie, la *'behavioural ecology'* [KD93] tente de faire le lien entre ces questions.

4.1.1.3 Behavioural ecology

La 'behavioural ecology' se focalise sur l'aspect fonctionnel du comportement. Un certain nombre d'expériences ont prouvé que des différences comportementales entre animaux peuvent être issues de différences génétiques. Berthold (cité dans [KD93]) a montré par exemple que les différences de comportement migratoire pour une même espèce pouvaient s'expliquer par des patrimoines génétiques différents. En effectuant un élevage sélectif, il a réussi à produire des groupes de fauvettes à tête noires constitués soit à 100% de migrateurs soit à 100% de résidents.

Ainsi, les comportements des individus dépendent en partie des gènes. Un patrimoine génétique est transmis en fonction des chances de reproduction de l'individu possédant ce patrimoine :

la théorie de la sélection naturelle implique que la nature va avoir tendance à favoriser les gènes qui permettent aux individus les possédant d'avoir le plus de chances de se reproduire et de transmettre ce patrimoine. Les comportements qui présentent un gain adaptatif (dépendant des conditions écologiques) vont avoir tendance à être plus sélectionnés. Ces remarques permettent alors de parler de sélection naturelle de comportement

Les tenants de la 'behavioural ecology' cherchent ainsi à comprendre pourquoi certains comportements ont pu être sélectionnés et pourquoi un comportement particulier contribue à augmenter les chances de reproduction d'un animal. Les tenants de ce courant analysent donc les comportements émis comme réponse à un problème d'optimisation posé à l'animal. Ils stipulent la présence d'un lien entre l'aspect fonctionnel du comportement et son aspect causal sans requérir la notion d'intentionnalité au niveau de l'individu. Ces considérations vont dans le sens de notre désir de production de comportement réactif.

Cette approche est très proche de nos préoccupations : les éthologues de ce courant cherchent à comprendre en quoi un comportement répond à un problème ; nos travaux cherchent à construire des comportements pour répondre à un problème donné. Ceci sous-entend qu'il peut être possible pour un problème donné de trouver un comportement permettant de résoudre ce problème dans la nature car celui-ci a pu être sélectionné pour ces raisons.

4.1.1.4 Groupes animaux

Comme nous cherchons à construire des systèmes au contrôle décentralisé nous allons nous focaliser sur les groupes d'animaux plutôt que sur des comportements solitaires. De plus, les collectivités animales respectent des contraintes que nous souhaitons imposer à nos systèmes :

- Chaque animal n'a accès qu'à des perceptions partielles
- Il est situé de ce fait dans un environnement incertain
- Dans un groupe animal, le comportement collectif est par essence la conséquence de comportements purement individuels

La sélection naturelle a pu choisir certains comportements individuels parce qu'ils permettent aux individus possédant les gènes correspondant d'avoir des chances de reproduction plus importantes à long terme. Dans cette optique, la sélection naturelle peut ainsi être considérée dans une certaine mesure comme un concepteur ayant testé tout au long de l'évolution un certain nombre de règles et de mécanismes permettant de construire des comportements de groupes en modifiant progressivement les comportements individuels. On peut espérer alors trouver dans la nature des solutions (ou tout au moins des sources d'inspiration) permettant à des sociétés de s'organiser pour répondre à un problème, tout en respectant les contraintes précisées précédemment.

En se focalisant sur les systèmes naturels collectifs, il est possible de développer les relations entre l'éthologie et les systèmes multi-agents.

4.1.2 Relations entre éthologie et systèmes multi-agents

4.1.2.1 Construction de modèles

Notre objectif dans ce chapitre va être de construire un modèle d'un phénomène collectif naturel pour disposer d'une première implémentation d'un concept d'interaction direct permettant de proposer un nouveau formalisme.

[Pic01] définit la notion de modèle en citant Soler :" Une représentation théorique qui ne prétend pas décrire fidèlement l'objet d'étude mais revendique au contraire son caractère délibérément schématique en même temps que la fécondité eu égard à un objet spécifié.". Cette définition correspond totalement à notre approche.

En effet, par la construction d'un modèle, nous n'envisageons pas de répondre à la question du fonctionnement du comportement animal. Nous ne prétendons pas chercher à reproduire fidèlement les processus physiologiques mis en oeuvre permettant de produire un comportement. Comme tout modèle, nous nous limiterons à un certain niveau d'abstraction.

La question qui se pose alors est de déterminer quand est ce qu'un modèle est considéré comme un bon modèle. [CFS+01] (dans le chapitre 7) fournit une réponse : un modèle a pour objectif de reproduire l'essence d'un phénomène. Nous nous intéresserons donc aux résultats qu'il est possible d'obtenir par exécution du modèle et vérifieront leur conformité par rapport aux résultats obtenus dans un cadre biologique. Cette conformité ne sera pas évaluée par rapport à des variables quantitatives mais par rapport à des types de comportements qualitatifs conformes aux observations réelles.

4.1.2.2 Approche 'animat'

L'approche animat proposée par Jean Arcady Meyer et Wilson [Wil91] constitue un projet de recherche dans cette direction. Cette approche préconise l'étude et la simulation du comportement animal pour pouvoir comprendre et reproduire la notion d'intelligence selon une approche ascendante [Sig03].

Cette approche se focalise sur les mécanismes adaptatifs des animaux à leur milieu et préconise une démarche incrémentale consistant à comprendre et modéliser tout d'abord les mécanismes d'organismes simples pour s'intéresser à des comportements adaptatifs plus complexes qui en sont le développement.

A plus court terme, l'approche animat cherche à mettre en évidence des mécanismes observables dans la nature permettant à des animaux de s'adapter à leur milieu et de réutiliser ces mécanismes dans d'autres cadres pour la résolution de problème (pour la construction de robots par exemple). C'est une démarche similaire que nous avons souhaité suivre dans ce chapitre : mettre en évidence des mécanismes d'adaptation collectifs qu'il est possible d'abstraire et de réutiliser dans des cadres formels.

4.1.2.3 Intérêts mutuels SMA-éthologie

Le travail consistant à produire un modèle a été réalisé en collaboration avec les éthologues du laboratoire de neurosciences comportementales de Nancy-1. Ils étaient eux aussi intéressé par la production d'un tel artefact mais pour des raisons différentes des nôtres.

Nous présentons ici de manière plus détaillée, l'intérêt que peut avoir la production d'artefacts aussi bien pour les éthologues que pour nos travaux.

Les SMA pour l'éthologie L'objectif des éthologues consiste avant tout à émettre des hypothèses sur les comportements animaux et de concevoir des outils leur permettant de valider ces hypothèses.

La démarche classique consiste à mener des expériences dans des environnements naturels pour extraire par induction des lois et des explications concernant les comportements observés. Ils effectuent pour cela des expériences scientifiques "en modifiant délibérément leurs conditions expérimentales afin d'établir des relations entre paramètres et réactions observées (Soler dans [Pic01])

Une fois ces relations établies, une démarche possible peut constituer à construire un modèle mathématique issu de ces hypothèses et à évaluer son caractère prédictif. Jusque récemment, les modèles traditionnels en éthologie se fondaient sur l'analyse de variables extensives comme les densités de population [Jud94].

En représentant explicitement l'individu, les systèmes multi-agents ont permis d'avoir des modèles plus expressifs. Les hypothèses concernant les comportements individuels et les interactions entre entités peuvent alors être représentées explicitement et il est possible dans ce cadre d'effectuer de l'"éthologie synthétique" [Pic01] : il s'agit de construire des laboratoires virtuels dans lequel les entités peuvent être étudiées.

Les systèmes multi-agents sont intéressants pour plusieurs raisons :

- Ils permettent de représenter des variables individuelles et des systèmes hétérogènes. Les systèmes multi-agents permettent par exemple de mesurer l'impact de variations inter-individuelles qui n'étaient pas présentes auparavant. Il est possible d'effectuer des expériences scientifiques comme défini précédemment sur un modèle disposant de plus de variables.
- Ils s'intéressent au passage du niveau local au niveau global et présentent le comportement collectif comme conséquence de comportements individuels. Cette hypothèse constitue une base de l'étude des groupes animaux. Ce passage entre comportement local et comportement global résultant peut être très complexe en raison de phénomènes émergents non représentés au niveau individuel. La simulation en intégrant ces deux niveaux de granularité constitue alors un bon outil pour valider la manière dont est réalisé ce passage.
- Ils intègrent explicitement l'environnement et les interactions comme faisant partie du système et ayant une influence sur la dynamique globale. Par exemple, ils sont particulièrement bien adaptés pour représenter des sociétés d'insectes dont le comportement collectif est souvent fondé sur la notion de stigmergie et pour lequel l'environnement est prépondérant.

Une fois un modèle construit, la question que se posent les éthologues consiste à déterminer quelle connaissance il est possible d'extraire d'un modèle et de sa simulation. Tout d'abord, les modèles sont des représentations simplifiées et sont construits par rapport à un objectif précis. Il est donc nécessaire de se prononcer dès la construction du modèle sur cet objectif afin de pouvoir extraire de l'information des expériences.

Ensuite, il faut être très vigilant quant aux conclusions qu'il est possible de tirer de simulations.

- Si les résultats qualitatifs obtenus par un modèle ne sont pas en conformité avec les attentes, cela signifie que la modélisation n'est pas assez fine ou que les hypothèses émises ne parviennent pas à capturer la réalité du phénomène. L'intérêt d'une simulation réside en effet dans sa capacité à séparer les éléments que l'on cherche à tester d'éléments perturbateurs. Il est fort possible qu'un élément jugé perturbateur fasse partie du phénomène. La simulation permet alors de mettre en évidence l'insuffisance des hypothèses à expliquer le

phénomène.

- Si les expériences fournissent les résultats attendus, cela ne signifie pas pour autant que le modèle correspond à une réalité quelconque. Simplement que les hypothèses permettent de reproduire qualitativement les phénomènes observés. Si ces hypothèses sont plus simples, du fait du canon de Morgan, elles seront considérées comme ayant un pouvoir explicatif plus important. Dans certaines conditions, ces considérations peuvent conduire à des remises en cause partielles d'explications établies. Ainsi, Hemelrijk [Hem00] a réussi à reproduire des phénomènes de réciprocité observés dans des groupes de macaques à partir d'hypothèses très simples et sans représentation explicite des autres. La simulation a donc remis en cause plusieurs capacités associées aux macaques sur la base de ces expériences en prouvant qu'elles n'étaient pas nécessaires pour voir apparaître le phénomène dans ses grandes lignes (même si, bien entendu, la simulation ne parvient pas à reproduire le comportement réel des groupes de macaques).

De nombreux travaux en collaboration avec des éthologues ont donné lieu à des simulations. Parmi ceux-ci, des travaux se sont intéressés à modéliser les comportements de construction collective de nid [BT94], les comportements de construction collective de toiles chez les araignées sociales [BC01] et [BCT03], les comportements coordonnés de capture de proies [DVB$^+$01], les phénomènes d'agrégation chez les cafards [JJD$^+$02], les comportements de fourragement dans des colonies de fourmis [DG89], les comportements de tric collectifs [LF94]. De plus en plus, les systèmes multi-agents constituent un outil de simulation utile pour l'éthologue et c'est pour cette raison que les éthologues du laboratoire de neuro-sciences comportementales de Nancy 1 ont été intéressés par nos travaux.

L'éthologie pour les SMA Réciproquement, l'éthologie des groupes animaux peut constituer une source d'inspiration très fructueuse pour la construction de systèmes multi-agents.

En effet, plusieurs raisons font qu'il y a un lien très fort entre les systèmes multi-agents que l'on souhaite construire et les groupes animaux qu'il est possible d'observer dans la nature :

- Dans un groupe animal, le comportement collectif est par essence la conséquence de comportements purement individuels. Cette considération est à rapprocher de la propriété fondamentale des systèmes que l'on souhaite construire à savoir le contrôle décentralisé.
- Les comportements observés dans des groupes animaux peuvent être compris comme issus d'un processus de sélection naturelle. C'est le point de vue de la *behavioural ecology* qui stipule que ces comportements collectifs ont une fonction adaptative et répondent à un problème d'optimisation. Nous cherchons à construire des systèmes distribués ayant pour objectif de résoudre des problèmes. S'il est possible de caractériser précisément le problème auquel le groupe animal répond et dans quelle mesure les comportements individuels permettent d'y répondre, il peut être possible de réutiliser les mécanismes mis en oeuvre dans des problèmes artificiels. Le phénomène de fourragement des colonies de fourmis est par exemple à l'origine de nouvelles techniques de résolution de problème d'optimisation combinatoire [DC99]
- Les animaux doivent pouvoir s'adapter à des environnements variables et fournir des réponses collectives pertinentes. Les comportements collectifs disposent de propriétés de robustesse qui font écho à notre volonté de disposer de systèmes fiables et adaptatifs.
- la méthodologie des éthologues se fonde sur le canon de Morgan. Ce principe est proche de nos préoccupations consistant à construire des comportements complexes à partir de briques les plus simples possibles.

Enfin, dans le cadre de construction de systèmes, la notion de métaphore peut être constructive. Dans sa thèse, Sébastien Picault (cf [Pic01]) insiste sur *'le statut ambivalent de l'informatique qui relève à la fois des sciences empiriques et des mathématiques en construisant ses objets d'études et en étudiant ces objets non pas sur leur propriétés intrinsèques mais sur leur comportement. Ainsi, lorsque l'informatique utilise des métaphores elle ne se contente pas d'importer une image au contenu vague. Elle construit un concept calculable qu'elle baptise par analogie avec un phénomène naturel'* .

Ce concept spécifié dans un cadre informatique peut être ensuite analysé indépendamment de son origine.

4.1.3 Bilan

L'objectif que nous nous sommes fixés au départ consiste à construire de manière automatique des systèmes multi-agents réactifs. Nous nous sommes intéressés aux approches basées sur des formalismes markoviens et avons montré leurs insuffisances. Nous avons donc cherché des sources d'inspiration permettant de proposer un nouveau formalisme et nous sommes tournés vers les phénomènes collectifs étudiés en éthologie.

Dans cette partie, nous avons présenté les liens qui peuvent exister entre les systèmes multi-agents et l'étude des comportements des sociétés animales, ainsi que les intérêts que peuvent avoir les systèmes multi-agents pour l'éthologie et réciproquement.

En conséquence, on gardera à l'esprit que notre objectif en construisant un modèle est de mettre en évidence un mécanisme réutilisable dans un cadre plus générique. Comme ce travail a été réalisé en collaboration avec des éthologues, on s'intéressera néanmoins aux aspects explicatifs des modèles que nous avons pu construire.

4.2 L'expérience de la piscine

L'objectif commun que nous avons avec les éthologues consiste à comprendre la manière dont certains animaux parviennent à s'organiser en société pour répondre à un problème collectif. Plus précisément, nous cherchons à savoir comment les individus parviennent à intégrer leur environnement social pour prendre une décision en considérant les autres individus avec lesquels il est mis en présence.

Afin de nous inspirer de mécanismes issus du vivant, nous avons cherché une situation dans laquelle les individus isolés ou en groupe sont capables de s'adapter pour fournir une réponse au problème et pour laquelle les comportements collectifs et individuels sont qualitativement différents. L'expérience de la piscine est une situation artificielle qui présente cette spécificité.

4.2.1 Description du dispositif expérimental

Plus précisément, le phénomène biologique sur lequel nous nous sommes focalisés est l'adoption de différents rôles au sein de groupes de rats. Le protocole expérimental développé par les éthologues du laboratoire de neurosciences comportementales de l'université UHP Nancy 1 consiste à mettre en présence un nombre limité de rats (généralement 6) dans un dispositif constitué d'une cage, d'une mangeoire et d'un couloir reliant ces deux lieux [DKTD91] . L'accès à la

nourriture est rendu de plus en plus difficile par l'immersion progressive du couloir, seul chemin permettant l'accès à la mangeoire. (cf figure 4.1).

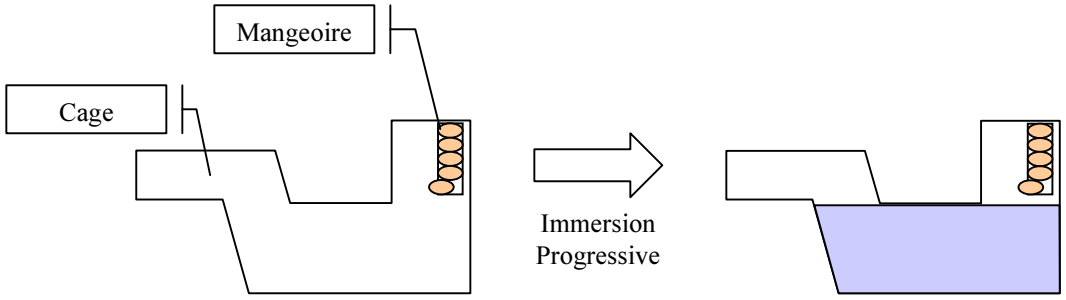


FIG. 4.1 – Dispositif expérimental

Une fois le couloir totalement immergé, le rat est obligé, pour se nourrir, de quitter la cage d'habitation et de nager en apnée le long d'un aquarium jusqu'à une mangeoire où il ne pourra obtenir qu'une croquette de nourriture à la fois. Par manque d'appui, le rat ne peut pas manger la nourriture sur place et est contraint, pour pouvoir s'alimenter, de rejoindre la cage d'habitation.

4.2.2 Expériences

4.2.2.1 Expérience principale

Après quelques jours, ce dispositif conduit à l'apparition d'une différenciation en deux profils comportementaux principaux : (a) les rats transporteurs qui plongent, qui vont effectivement chercher la nourriture et la ramènent dans la cage et (b) les rats non-transporteurs qui ne plongent jamais, restent dans la cage et parviennent à subvenir à leur besoin en volant la croquette des rats transporteurs (cf [DKTD91]). Il est important de noter que tous les rats parviennent à survivre dans ce dispositif : les rats transporteurs réussissent à s'alimenter quand tous les rats non-transporteurs sont repus ou en train de se nourrir.

Cette différenciation sociale apparaît régulièrement (cf [DT92], [DKTD91]) et ce quels que soient les individus introduits dans le dispositif. De plus, les tailles relatives des groupes des rats transporteurs et des rats non-transporteurs restent constantes (50 % des rats introduits dans le dispositif deviennent transporteurs). La différenciation reste stable pendant plusieurs mois et conduit le système dans un état viable. Enfin, cette différenciation a été observée pour diverses espèces de rats (dont Long-ewans, Wistar, ...) ce qui suppose la présence d'un mécanisme de régulation générique que l'on peut retrouver chez ces différentes espèces.

4.2.2.2 Expérience re-différenciation

Lorsqu'on met en présence dans le dispositif des rats différenciés dans des expériences préalables, l'expérience conduit à une nouvelle différenciation (cf figure 4.2) : certains ex-transporteurs deviennent non-transporteur et réciproquement. Il semble ainsi que le profil ne soit pas directement inscrit dans les rats mis en présence mais serait plutôt la résultante des interactions qui ont eu lieu entre les individus au cours de l'expérience.

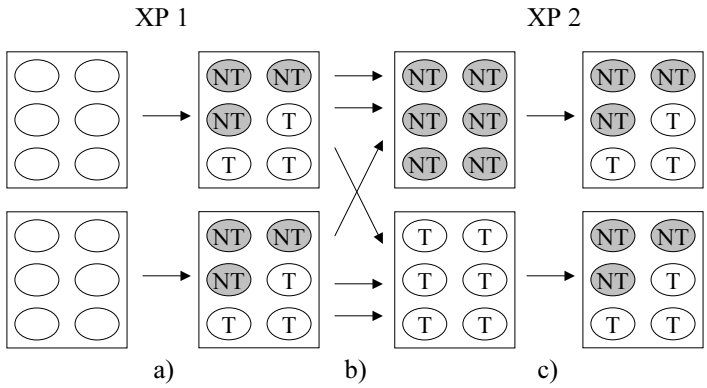


FIG. 4.2 – Expériences de redifferenciation : a) première différenciation, b) répartitions des rats dans de nouvelles cages, c) re-differenciation

4.2.2.3 Expérience de différenciation avec rats drogués

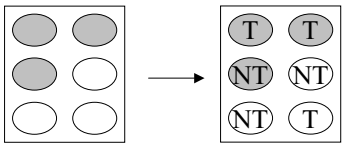


FIG. 4.3 – Expérience de différenciation avec certains rats drogués (en gris sur le schéma)

Lorsque la moitié des rats introduits dans le dispositif est droguée avec du diazépan un dérivé du valium (cf [STND98]), la spécialisation est de nouveau observée avec les mêmes proportions. Mais si on étudie de manière plus attentive les sous-groupes drogués et non drogués, on remarque une sur-représentation du profil transporteur (environ 80 % des rats) dans le groupe de rats drogués et une sur-représentation du profil non-transporteur dans le sous-groupe des rats non drogués (cf fig 4.3). On peut en tirer deux conclusions. Tout d'abord, l'adoption des profils semble fortement liée à l'anxiété des rats comme le confirme les apparitions de profils observées dans le sous-groupe de rats drogués. Ensuite, l'environnement social a une forte influence sur l'adoption du profil des rats. Ainsi, malgré le fait que le sous-groupe de rats non drogués soit un sous-groupe standard, la présence de rats drogués dans le même dispositif a conduit à une sur-représentation du profil non-transporteurs dans ces sous groupes. Les rats drogués se sont donc adaptés aux caractéristiques comportementales des autres rats avec lesquels ils ont été mis en présence.

Cette expérience suggère la présence de mécanisme permettant à un individu de considérer les autres individus avec lesquels il est mis en présence.

4.2.2.4 Influence de l'effectif du groupe

Dans une série expérimentale, des rats ont été introduits seuls dans le dispositif. Dans cette situation, tous les rats plongent et parviennent à atteindre la nourriture directement. Tous les rats possèdent donc les capacités physiques nécessaires leur permettant d'adopter un profil transporteur. L'adoption d'un profil non-transporteur peut être interprétée comme la conséquence de

l'influence de l'environnement social des rats.

De plus, les proportions des rats dans les sous-groupes dépendent fortement des effectifs des rats mis en présence. S'il s'agit des groupes de deux rats, un rat non transporteur apparaît dans 10% des cas. S'il s'agit de groupe de 3, cette proportion est de 50%. S'il s'agit de groupes de 4, cette proportion passe à 80%. Ces derniers résultats viennent renforcer l'hypothèse selon laquelle l'intégration du contexte social est responsable de l'apparition de profils spécifiques dans le dispositif.

4.2.2.5 Étude ontogénétique

D'autres études ont été menées concernant la présence d'éventuelles prédispositions chez un individu à acquérir un profil plutôt qu'un autre.

Une série expérimentale a ainsi été menée : des portées de rats ont été suivies depuis la naissance avant d'introduire ces rats dans le dispositif. Sur le plan individuel, l'évolution pondérale des individus ainsi que leurs capacités motrices ont été suivies. Sur le plan social, les animaux ont été confrontés à des situations de compétitions à deux et leurs réactions enregistrées.

Il ressort de ces expériences qu'il est possible de prédire très précisément le comportement adopté par chaque individu lorsque plusieurs individus sont mis en présence dans la situation expérimentale[4]. L'émergence de profils comportementaux peut alors être comprise comme l'amplification de légères différences inter-individuelles dues aux interactions entre les individus.

4.2.3 Conclusion

Ces expériences ont permis de mettre en évidence un certain nombre d'éléments.

Tout d'abord, le caractère systématique de cette spécialisation (indépendant des espèces, des individus, de la taille des effectifs) ainsi que le temps nécessaire pour son apparition suggèrent qu'un mécanisme de régulation collectif indépendant des individus est en oeuvre et permet à la collectivité de s'auto-organiser.

Ce mécanisme est constitué de deux types d'adaptations :
- un mécanisme d'adaptation individuel permettant à chaque rat dans le système de subvenir à ses besoins d'une certaine manière en collectivité (en volant de la nourriture ou en plongeant) ou en plongeant lorsqu'il se trouve seul dans la cage.
- un mécanisme d'adaptation collectif extrêmement robuste permettant une distribution des rôles au sein de la population.

Les expériences (comme celles consistant à inoculer du Diazepan à une partie des individus) laissent supposer en outre que chaque individu intègre son contexte social et que le profil adopté en est la conséquence. Nous nous sommes donc demandés avec les éthologues comment des mécanismes de régulation individuels simples pouvaient conduire à un comportement collectif permettant à l'ensemble des individus de la collectivité de satisfaire un besoin interne : la faim ressentie localement par chaque individu mais non perçue par les autres individus.

[4] Les facultés motrices semblent par exemple plus développées chez les futurs transporteurs.

En particulier, les éthologues ont cherché à savoir si des processus cognitifs complexes étaient impliqué dans cette différenciation et s'il était possible de la reproduire à partir de comportements très simples ne nécessitant pas de représentation complexe des autres individus. Pour répondre à cette question, nous avons cherché à construire un modèle de ce phénomène avec pour objectif de reproduire qualitativement les observations biologiques.

A long terme, comme nous le présenterons dans la chapitre 5, nous espérons en tirer des mécanismes permettant à un agent de considérer les autres agents du système et des lois d'adaptation permettant de construire des comportements collectifs à partir de comportements individuels.

4.3 Modèles de spécialisation existants

Afin de mettre en évidence l'originalité de notre modèle, nous présentons tout d'abord deux modèles de spécialisation existants : le modèle des réponses à seuil qui parvient à reproduire des spécialisations observés dans des sociétés d'insectes et le modèle des relations de dominance qui reproduit des structurations spatiales de colonies de macaques. Nous les expliciterons et mettrons en évidence leurs insuffisances à reproduire et à expliquer le phénomène collectif de spécialisation dans des groupes de rats.

Ces insuffisances nous semblent par ailleurs très intéressantes et nous effectuerons un parallèle avec les DEC-POMDPs. En effet, a l'issue de cette présentation, nous serons amenés à proposer un modèle original : le modèle Hamelin fondé sur un couplage de ces deux modèles pour essayer de reproduire le phénomène biologique présenté dans la partie précédente.

Cette dernière proposition constitue une reproduction à plus petite échelle de la démarche globale que nous avons suivi dans cette thèse consistant à intégrer la notion d'interaction directe pour tirer parti de mécanisme d'adaptation individuels insuffisant pour construire un système collectif fondé sur des perceptions locales.

4.3.1 Spécialisation dans des colonies d'insectes sociaux

Des modèles existent déjà pour expliquer l'apparition et la genèse de phénomènes de spécialisation dans des colonies d'insectes. Ils sont fondés sur des mécanismes d'adaptation individuels et exhibent une plasticité qui peut être observée dans les spécialisations de groupes de rats.

Ces modèles permettent d'expliquer comment il est possible d'obtenir des phénomènes de spécialisation à partir de mécanismes d'adaptation individuels comme cela semble être le cas pour l'expérience de la piscine, puisqu'un rat seul parvient à apprendre à accéder à la nourriture. De tels modèles peuvent donc constituer de bons prétendants pour modéliser le phénomène biologique observé dans l'expérience de la piscine.

4.3.1.1 Présentation générale

Il est possible d'observer une division du travail dans les colonies d'insectes sociaux. Cette division du travail peut prendre trois formes différentes (cf [BT99]) :

- le polyethisme temporel, pour lequel les individus du même âge ont tendance à effectuer les mêmes tâches.
- le polymorphisme, pour lequel les travailleurs ont des morphologies différentes et les travailleurs d'une même caste morphologique ont tendance à effectuer la même tâche
- la variabilité individuelle pour laquelle des individus extérieurement identiques adoptent des réactions comportementales différentes

La principale caractéristique de la division du travail réside dans sa capacité à s'adapter : le nombre d'individus accomplissant une tâche donnée évolue en fonction de la réponse collective à apporter à l'environnement extérieur.

Dans cette partie, nous nous concentrerons sur les variabilités inter-individuelles pour essayer de comprendre quelle peut être leur utilité et comment une variabilité inter-individuelle peut permettre d'adapter la répartition des tâches au sein d'une colonie.

4.3.1.2 Réponses à seuils

Afin de rendre compte de la division du travail dans les colonies d'insectes, Bonabeau et al [BT99] ont proposé le modèle des réponse à seuil. Selon ce modèle, chaque individu correspond à un agent i caractérisé par un seuil θ_i. Un stimulus s correspondant à la tâche à résoudre est émis dans l'environnement. Ce stimulus est global et potentiellement perceptible par l'ensemble des individus (comme illustré sur la figure 4.4).

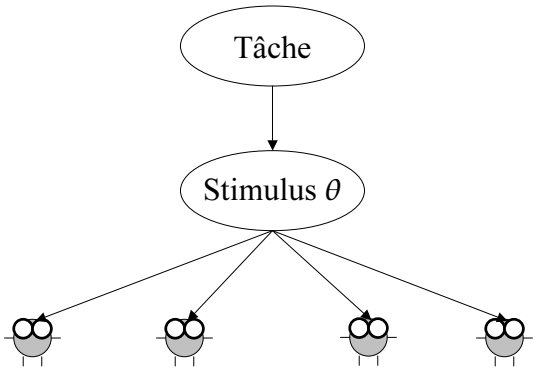


FIG. 4.4 – Stimulus global pour les réponses à seuil

Dès que le stimulus associé à la tâche est perçu par un agent i avec une intensité supérieure à seuil θ_i, l'agent concerné engage une réponse adaptée par des actions visant à l'accomplissement de la tâche et à la diminution de l'intensité du stimulus.

Une tâche Lorsqu'il n'y a qu'une seule tâche à résoudre, la fonction de réponse d'un agent dépend de son seuil θ et de l'intensité du stimulus s associé à la tâche. Cette réponse est modélisée par la fonction (n désigne un entier naturel) :

$$T_\theta(s) = \frac{s^n}{s^n + \theta^n}$$

qui associe au stimulus s la probabilité qu'a l'agent d'émettre une réponse adaptée à la tâche.

Pour un θ donné, la forme de la courbe de T_θ est représentée figure 4.5. Ainsi plus le stimulus est important, plus la probabilité de s'engager dans l' accomplissement de la tâche est grande. Inversement, pour un s donné, plus θ est important, plus la probabilité de s'engager est faible.

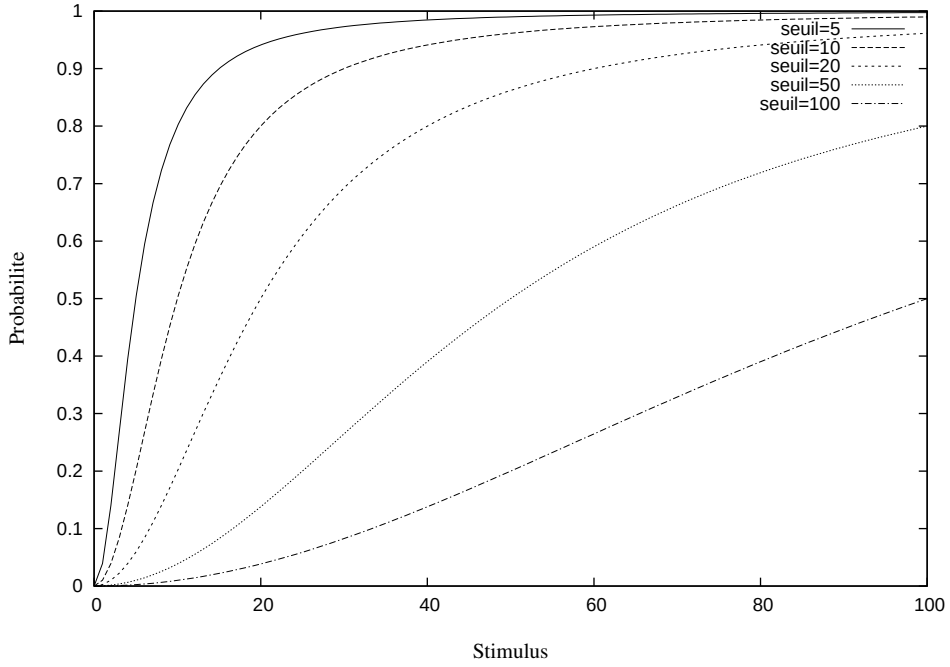


FIG. 4.5 – Courbe de probabilité en fonction du stimulus et du seuil

Ce mécanisme associé à une variabilité inter-individuelle du coefficient θ permet d'expliquer la régulation du nombre d'individus à une tâche donnée et le fait que ce soient toujours les mêmes individus qui vont prioritairement résoudre la tâche. Ainsi, lorsque le stimulus associé à la tâche à accomplir augmente, les individus dotés d'un seuil θ plus faible vont avoir tendance à réagir plus rapidement. Si à un instant donné, le nombre d'individus s'occupant de la tâche n'est pas assez élevé, le stimulus associé à la tâche augmente et déclenche les comportements d'autres individus au seuil légèrement plus important et ainsi de suite, jusqu'à ce qu'assez d'agents s'occupent de la tâche et que l'intensité du stimulus associé diminue.

Ce mécanisme peut être compris comme un mécanisme d'asservissement distribué consistant à réguler le nombre d'agents pour répondre à la tâche.

Plusieurs tâches Ce modèle peut être étendu à plusieurs tâches. Chaque agent a alors un seuil particulier par tâche. Il est possible d'observer à nouveau des spécialisations en fonction des populations d'agents.

Ce modèle permet aussi d'expliquer l'émergence de comportements plus complexes dûs à l'imbrication des exécutions des réponses et des apparitions de stimuli. Par exemple, les comportements d'agression sont déclenchés par la présence de proies dans l'environnement. Ces derniers

comportements conduisent à l'apparition de cadavres qui déclenchent à leur tour d'autres comportements.

4.3.1.3 Spécialisation

Le modèle précédent explique comment la variabilité inter-individuelle des seuils permet de réguler les activités de la colonie et de faire apparaître une spécialisation. Mais ils ne rendent pas compte de l'apparition d'allocation de tâche parmi des individus initialement identiques. D'autres expériences ont montré que la probabilité qu'a un individu de se consacrer à une autre tâche j que la tâche i à laquelle il a tendance à répondre est décroissante par rapport au temps passé à la résolution de la tâche i [BDT99].

Ces résultats suggèrent qu'il existe des processus d'apprentissage individuel qui permettraient de rentabiliser l'expérience acquise par un individu.

Pour cela, Theraulaz et Bonabeau ont proposé un modèle de réponse à seuil variable. De la même manière que précédemment, chaque agent est caractérisé par un seuil par tâche mais lorsqu'un agent s'engage vis à vis d'une tâche, le seuil de réponse au stimulus associé diminue.

$$\theta \leftarrow \theta - \delta$$

Inversement, tant que l'agent ne s'engage pas, son seuil continue à augmenter (jusqu'à un maximum).

$$\theta \leftarrow \theta + \epsilon$$

Cette approche fondée sur des renforcements locaux permet d'adapter les individus aux tâches rencontrées. Plus un individu s'engage dans une tâche i, plus son seuil sera faible et plus il sera rapide pour reproduire les actions liées à la résolution de la tâche i dans le futur.

Ces renforcement permettent d'adapter les distributions des seuils entre les individus et de répartir les différentes tâches à traiter entre les individus. Si la tâche n'est pas traitée par assez d'individus, le stimulus associé va croître et déclencher de nouvelles réponses. Les individus qui vont répondre à cette tâche seront plus sensibles dans le futur au stimulus associé et répondront plus rapidement à ce besoin.

Ce dernier modèle plus complexe parvient à expliquer la genèse d'une spécialisation à partir d'individus initialement indifférenciés et à lier la spécialisation aux tâches à résoudre.

4.3.1.4 Conclusion

Le modèle de réponse à seuil permet d'expliquer comment une spécialisation peut apparaître dans le système à partir d'adaptations individuelles des seuils de réactions et de stimuli globaux disposés dans l'environnement et perceptibles par les agents.

Si on analyse ce modèle en tant que système multi-agent, les agents sont effectivement en interaction (en terme de couplage) mais, comme les DEC-POMDPs, les seules interactions impliquées sont des interactions indirectes : les agents perçoivent des stimuli dans leur environnement,

émettent des actions qui modifient la valeur de ces stimuli et modifient en conséquence les perceptions des autres agents.

Or, le problème de spécialisation observé dans des groupes de rats se fonde non pas sur des stimuli extérieurs émis par l'environnement et observables par plusieurs individus, mais sur des stimuli individuels proprioceptifs que sont la faim des individus mis en présence. Le fait que ces besoins soient individuels et ne puissent pas être perçus par la collectivité nécessite d'autres mécanismes que des interactions indirectes. Les agents doivent intégrer les besoins des autres individus pour pouvoir fournir une réponse adaptée qui ne peut s'expliquer par la présence d'un environnement commun dans lequel ces besoins s'expriment.

Les modèles de réponse à seuil peuvent rendre compte de l'adaptation du comportement d'un individu capable de répondre de plus en plus rapidement à son besoin mais ne peut expliquer la structuration des échanges qui apparaît au sein de la société. Pour cela, nous nous sommes concentrés sur un autre modèle : les relations de dominance.

4.3.2 Relations de dominance

Les relations de dominance sont considérées comme d'une importance centrale dans les comportements sociaux de nombreux groupes animaux : bancs de poisson , poules , chevaux, insectes sociaux (cf [BTD96]). Les travaux d'Hemelrijk (cf [Hem99]) cherchent à expliquer l'apparition d'une structuration spatiale et comportementales des sociétés de macaques à partir de relations de dominance fondées sur des variables individuelles.

4.3.2.1 Émergence de la Réciprocité

Une grande partie des phénomènes sociaux complexes qui peuvent apparaître dans les sociétés de primates ont longtemps été attribués à l'intelligence des individus. Les phénomènes de coalition et de réciprocité entrent dans cette catégorie : certains macaques n'hésitent pas à prendre part à des combats pour supporter l'un des deux camps. Ces comportements ont été supposés

Les éthologues ont supposé que l'individu faisait cela par intérêt car il attend une réciprocité de l'aide dans le futur (cf [Hem96]). Ces raisonnements supposent une reconnaissance des situations de conflit et nécessitent des capacités sociales développées. Les phénomènes de réciprocité observés dans les groupes de macaques ont ainsi pendant longtemps constitués une preuve de l'intelligence développée de ces animaux.

Hemelrijk a proposé un modèle de différenciation qui permet d'expliquer ces phénomènes à partir de règles extrêmement simples permettant à un individu d'intégrer son contexte social.

4.3.2.2 Modèle

Le modèle proposé par Hemelrijk est constitué
- d'agents représentant des macaques caractérisés par une valeur de dominance et par une position au sein de leur environnement
- d'un environnement dans lequel les agents peuvent se déplacer

Règles de déplacement Les agents évoluent selon des règles observées dans des groupes de macaques et fondées sur la notion de distance critique. Le comportement d'un agent est le suivant :

- Si un individu ne perçoit pas de congénères, il explore l'environnement pour en trouver.
- Si un individu aperçoit un (ou plusieurs) autre(s) individu(s) dans son rayon perceptif à une distance supérieure à une distance critique, il se rapproche d'eux.
- Enfin, si deux individus sont éloignés d'une distance inférieure à la distance critique, une interaction de dominance a lieu entre les deux agents.

Interaction de dominance Une interaction de dominance implique deux agents et a pour résultat la victoire de l'un d'entre eux. Chaque agent possède une valeur de dominance propre *dom*. Lorsque les deux agents sont proches l'un de l'autre, chaque agent observe la valeur de dominance de l'autre et une interaction a lieu entre ces deux agents.

La résolution du combat dépend des agents impliqués et le résultat est déterminé stochastiquement en fonction des valeurs de dominance relatives des agents. La probabilité que possède l'agent A de remporter le combat sur l'agent B est calculée par l'équation suivante :

$$P(victoire_A) = \frac{dom_A}{dom_A + dom_B}$$

Une fois l'interaction résolue, les valeurs de dominance des agents sont renforcées selon un 'winner and looser effect' : l'agent victorieux voit sa valeur de dominance augmenter et l'agent perdant voit sa valeur de dominance décroître. De plus, les mises à jour sont effectuées en fonction des probabilités de victoire : plus la probabilité de victoire de l'agent victorieux est faible, plus les modifications seront importantes. Le renforcement est effectué selon les formules suivantes ($victoireA$ désigne une valeur 0 ou 1 correspondant au booléen l'agent A est victorieux).

$$dom_A = dom_A + (victoire_A - \frac{dom_A}{dom_A + dom_B}) * \delta_{dom}$$

$$dom_B = dom_B - (victoire_A - \frac{dom_A}{dom_A + dom_B}) * \delta_{dom}$$

A l'issu d'un combat, l'agent victorieux parvient à repousser l'agent perdant qui doit répondre en fuyant dans la direction opposée.

4.3.2.3 Résultats

Les résultats obtenus à l'exécution de la simulation permettent d'expliquer un nombre important de phénomènes observés dans les groupes de macaques :

- A partir de variations effectuées sur les seuls coefficients de renforcement, il est possible d'observer un continuum de comportements collectifs entre ceux observés dans les groupes de macaques égalitaires et ceux observés dans des groupes de macaques despotiques
- La disposition spatiale des agents est couplée à leur rang de dominance conformément aux observations biologiques [Hem00]
- Des phénomènes de réciprocité parviennent à être reproduits par ces règles : les déplacements induisent une structuration spatiale des agents. Cette structuration spatiale constitue le support des combats et permet d'expliquer pourquoi ce sont les mêmes agents qui se supportent mutuellement lors des combats.

En conclusion, le couplage d'une hiérarchie de dominance avec d'autres comportements permet l'émergence de comportements collectifs complexes et permet d'observer une structuration du groupe.

4.3.2.4 Synthèse du modèle des relations de dominance

Les travaux d'Hemelrijk parviennent à reproduire à partir de relations de dominance une structuration spatiale du groupe d'agents. Cette structuration se fonde sur deux types d'interactions : les interactions indirectes et des interactions directes. Les interactions indirectes sont dues aux positions des agents qui influencent leurs comportements de déplacement. Les interactions directes ont lieu lors des relations de dominance : les agents échangent leur valeur de dominance et une décision collective locale est prise, consistant à déterminer quel agent sort victorieux à l'issue de l'interaction.

La structuration spatiale émergente à l'issue de la simulation distribue physiquement les individus en fonction de leur valeur de dominance individuelle non perceptible directement. Chaque agent au cours de ses déplacements et de ses rencontres a réagi en fonction de ses variables individuelles mais aussi des variables des autres agents impliqués dans les interactions. Les interactions de dominance ont permis d'échanger de l'information entre les agents au sujet d'une partie de leur comportement (caractérisé par leur relation de dominance). C'est justement cet échange d'information individuelle qui manquait aux réponses à seuil pour expliquer la structuration du groupe d'agents à partir de stimuli individuels.

Ainsi, comme les relations de dominance couplées à des déplacements induisent une structuration spatiale, nous avons cherché à étudier si de la même manière les relations de dominances couplées avec des renforcements de comportements permettent d'induire une structuration comportementale dans le système permettant d'expliquer la spécialisation observée dans des groupes de rats.

4.3.3 Fondement du modèle Hamelin

Les éthologues ont montré que l'expérience de la piscine conduit à l'apparition d'une spécialisation comportementale entre les individus mis en présence et que cette spécialisation implique une intégration de l'environnement social au niveau individuel. Afin de comprendre ce mécanisme et d'en avoir une première instanciation manipulable, nous avons choisi de construire un modèle permettant de reproduire les observations de ce système biologique.

Pour ce faire, nous nous sommes focalisés dans cette partie sur les modèles de spécialisation existants :

- le modèle de réponse à seuil fondé sur des interactions indirectes permet d'expliquer comment il est possible de produire une adaptation collective à partir de processus d'adaptation individuels mais nécessite un stimulus global perceptible par tous les agents.
- le modèle de relation de dominance permet une structuration spatiale du système à partir d'échanges locaux mais n'intègre pas de processus d'adaptation individuel à une tâche.

L'idée a été de coupler les deux modèles pour disposer :

- d'un mécanisme d'adaptation individuel permettant à un agent d'aller chercher sa nourriture lorsqu'il a faim : l'item comportemental de plongée va être modélisé par des réponses

à seuil adaptatives
- d'un mécanisme d'adaptation collective chargé du transfert de nourriture entre les agents et fondé sur les relations de dominance.

afin de voir s'il était possible de reproduire une structuration comportementale à partir de stimuli individuels.

Enfin, les comportements de combat et de plongée sont couplés par le transfert de croquettes qui a lieu entre les rats. Ainsi un rat qui gagne fréquemment ses combats n'aura pas à plonger tandis qu'un rat qui les perd de manière systématique sera contraint de plonger pour pouvoir satisfaire ses besoins (cf figure 4.6).

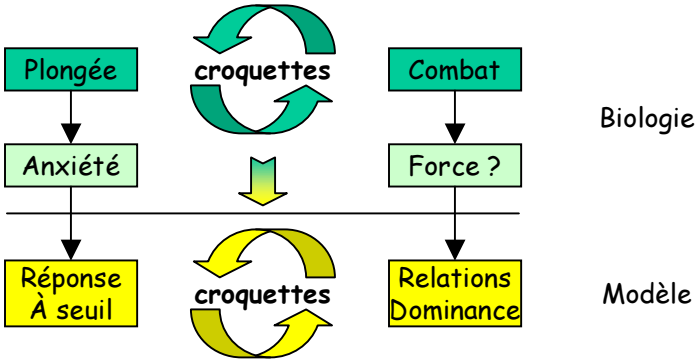


FIG. 4.6 – Principes du modèle

Ces considérations ont donné lieu au modèle Hamelin que nous allons décrire dans la partie suivante.

Ce modèle a été décrit par ailleurs dans plusieurs publications (cf [TBCD04], [TBCD02])

4.4 Modèle Hamelin

Le modèle Hamelin est basé sur un système multi-agents et a pour objectif de vérifier comment des mécanismes d'adaptation individuels peuvent permettre de produire une structure collective adaptative. En particulier, nous avons cherché à voir s'il était possible de produire cette structuration de la collectivité sans cognition sociale.

Le modèle Hamelin a été développé conformément au principe de parcimonie, proche du critère de Morgan, qui cherche à reproduire l'essence du phénomène à partir des hypothèses les plus simples possibles.

Dans cette partie nous nous contrerons sur les différents composants du système multi-agents Hamelin, à savoir, les agents, l'environnement et les interactions.

4.4.1 Agents

Selon le principe de parcimonie, les rats (bien que dotés de capacités cognitives complexes) sont modélisés par des agents réactifs. Ils sont régis par des règles de type stimulus-réponse et

prennent leurs décisions en fonction de leur perception immédiate et de leurs variables internes.

L'état interne d'un agent est caractérisé par 4 variables. Ces variables correspondent à des paramètres qui se sont révélés avoir de l'importance dans les expériences biologiques. Ces variables sont :

- la faim f qui caractérise le besoin de nourriture et constitue la motivation de l'agent.
- la quantité de nourriture *nour* possédée par l'agent. Les croquettes ont une taille initiale et sont consommées progressivement. La quantité de nourriture est implémentée comme la taille restante de la croquette que l'agent a en possession.
- la force *dom* d'un agent qui correspond à sa capacité à remporter un combat lorsqu'il est impliqué. Cette force est analogue au rang de dominance dans les travaux de [Hem99].
- l'anxiété θ de l'agent vis à vis de l'eau correspondant à l'appréhension qu'a l'agent de plonger dans l'eau.

Du fait du principe de parcimonie que nous avons suivi, nous n'avons pas introduit de couplage direct entre ces variables. Par exemple, la capacité d'un agent à plonger est supposée entièrement indépendante de son niveau de faim.

L'activité d'un agent est la combinaison de trois items comportementaux qui seront décrits ultérieurement :

- un item comportemental de plongée
- un item comportemental d'agression
- un item comportemental d'alimentation

Chaque agent est doté de perceptions partielles. A un instant donné, il sait s'il possède une croquette et peut détecter les autres croquettes possédées par les agents dans la cage. Par contre, toujours du fait de notre principe de parcimonie, les perceptions des agents sont très simples : un agent ne peut pas identifier les autres agents présents dans la cage ni reconnaître quel agent possède une croquette.

4.4.2 Environnement

L'environnement correspond au dispositif expérimental. Toujours selon le principe de parcimonie, l'environnement n'a pas de topologie. Il est caractérisé par la taille du couloir immergé et par la taille initiale des croquettes insérées dans le dispositif. Ces caractéristiques sont implémentées sous la forme de deux variables :

- τ le nombre de pas de temps mis pour consommer entièrement une croquette
- η l'énergie absorbée pendant un pas de temps consacré à consommer la croquette

Ainsi, l'énergie contenue dans une croquette entière est $\tau\eta$.

Il est à noter que l'unité de temps choisie est la durée d'un cycle ce qui correspond au temps nécessaire à un agent pour traverser le couloir immergé et revenir dans la cage. Ainsi, une augmentation de τ peut être interprétée de deux manières : soit la taille initiale des croquettes dans le dispositif augmente, soit la longueur du couloir immergé diminue.

4.4.3 Item comportementaux

Chacun des items comportementaux (plonger, attaquer et manger) est déclenché ou exécuté de manière stochastique. Les probabilités qui y sont associées sont calculées à partir de l'état

interne de l'agent.

Quand un item est déclenché, l'action est effectivement exécutée et un renforcement modifie l'état interne de l'agent afin de modifier son comportement en fonction de ses actions passées et de doter l'agent de capacités d'adaptation.

4.4.3.1 Plonger

Condition de déclenchement Cet item comportemental est considéré par un agent à chaque fois qu'il ne possède pas de croquette.

Modèle Cet item comportemental est régi selon une réponse à seuil adaptative basée sur la faim de l'agent et son anxiété vis à vis de l'eau.

Probabilité Plus la faim de l'agent est importante, plus sa probabilité de plonger le sera. Au contraire, plus son anxiété est faible, plus cette probabilité sera faible. Ces tendances sont résumées par l'équation suivante directement inspirée des réponses à seuil adaptatif (cf partie 4.3.1.2) :

$$P_{plonger} = \frac{f^2}{f^2 + \theta^2}$$

Conséquences Une fois que l'agent a décidé de plonger, l'action est automatiquement exécutée. L'agent plonge dans l'eau, parvient à prendre une croquette de la mangeoire et revient à la cage. Cette action est supposée instantanée mais ne peut être effectuée qu'une fois par cycle. L'agent possède alors une nouvelle croquette et la quantité de nourriture en sa possession est mise à jour :

$$nour \leftarrow \tau$$

Renforcement De plus, lorsque l'action est exécutée, l'anxiété du rat est réduite selon une formule proche des renforcements des réponses à seuil adaptatifs :

$$\theta \leftarrow \theta.\delta_\theta$$

$\delta_\theta \in [0, 1]$ est un paramètre global. Du fait de ce renforcement, l'agent apprend à réagir plus rapidement au même niveau de faim. Ce renforcement est responsable de l'adaptation individuelle des agents et de l'apparition d'un partage de tâche entre les agents.

Enfin, les agents ont tendance à oublier leurs expériences passées. Ainsi, leur anxiété croit graduellement au cours du temps

$$\theta \leftarrow \theta + (1000 - \theta).\delta_f$$

$\delta_f \in [0, 1]$ constitue aussi un paramètre global du système responsable de cet augmentation. La valeur 1000 désigne la plus haute anxiété possible et est un paramètre global donné a priori.

4.4.3.2 Attaquer

Condition de déclenchement L'action est considérée par un agent lorsqu'il n'a pas de croquette et qu'il perçoit la présence de croquettes dans la cage.

Modèle Les combats sont régis par des règles de dominance identiques à celles utilisées par Hemelrijk.

Lois de probabilité Lorsque les conditions sont réunies, l'item comportemental est systématiquement déclenché et la victime est choisie de manière aléatoire parmi les possesseurs de croquettes. L'issue de l'interaction est par contre déterminée stochastiquement en fonction des forces relatives des agents impliqués dans le combat. La probabilité pour l'agresseur (agent A) de voler la croquette de la victime (agent B) est calculée à partir des valeurs de dominance des agents :

$$P(victoire_A) = \frac{dom_A}{dom_A + dom_B}$$

Conséquences Si l'action est réussie, l'agresseur parvient à voler la croquette à la victime. Les quantités de nourriture possédées par les agents sont alors mises à jour :

$$nour_A \leftarrow nour_B, \text{ et } nour_B \leftarrow 0$$

Dans le cas contraire, l'action n'a pas de conséquence pour le système en dehors des renforcements qui vont s'exercer.

Renforcement Que l'action soit réussie ou non, la force de l'agent victorieux (A ou B) est renforcée tandis que la force de l'agent vaincu est réduite selon le principe du "winner and looser effect". Les modifications des forces sont calculées selon les formules de mises à jour des valeurs de dominance présentées dans [Hem00] (cf partie 4.3.2) :

$$dom_A = dom_A + (victoire_A - \frac{dom_A}{dom_A + dom_B}) * \delta_{dom}$$

$$dom_B = dom_B - (victoire_A - \frac{dom_A}{dom_A + dom_B}) * \delta_{dom}$$

δ_{dom} est aussi un paramètre global de la simulation défini a priori et $victoire_A$ correspond au booléen "l'agent A a gagné le combat".

4.4.3.3 Manger

Condition de déclenchement Cette action est systématiquement exécutée.

Conséquences Si l'agent possède une croquette, celle-ci diminue de taille et la faim de l'agent diminue en conséquence

$$f \leftarrow f - \eta, nour \leftarrow nour - 1$$

Sinon, la faim de l'agent augmente

$$f \leftarrow f + 1$$

4.4.4 Cycle d'exécution

Une simulation est constitué par l'exécution séquentielle de plusieurs cycles. Chaque cycle se décompose de la manière suivante :

1. Les items comportementaux de plongée sont testés et exécutés simultanément pour l'ensemble des agents
2. Les agents possédant une croquette et les agents n'en possédant pas sont séparés dans deux groupes. Chaque agent du groupe ne possédant pas de croquette attaque un agent du groupe possédant des croquettes. Les choix des agents entrant en conflit sont déterminés de manière aléatoire. A chaque fois qu'un combat est déclenché, il est résolu et les groupes des agents sont modifiés en conséquence. Une même croquette peut donc transiter entre plusieurs agents en un seul cycle.
3. Tous les agents déclenchent leur item comportemental d'alimentation.

4.5 Validation de l'adaptation

Des simulations ont été conduites afin de confronter les résultats obtenus par le modèle aux résultats observés dans les expériences biologiques (cf [TBCD04]).

Du fait de la complexité de la dynamique, discrète, stochastique et non-linéaire, nous avons opté pour une analyse empirique du système.

Dans cette partie, nous nous concentrerons principalement sur l'analyse du système en tant que reproduction d'un phénomène biologique. Nous effectuerons des simulations correspondant à certaines expériences biologiques qui ont pu être faites et vérifieront que les résultats sont qualitativement similaires.

4.5.1 Adaptation individuelle

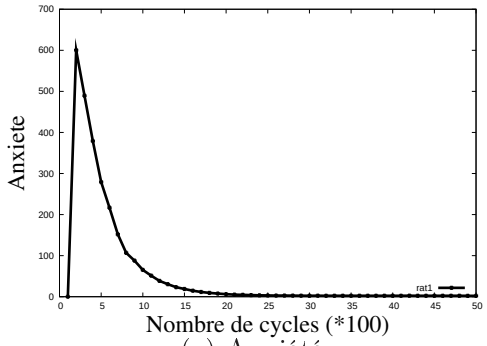
Quand un agent seul est confronté à une difficulté d'accès à la nourriture, il n'a pas la possibilité d'interagir avec d'autres agents. Son comportement est donc régi uniquement par un modèle de réponse à seuil.

La faim de l'agent augmente jusqu'à ce que l'agent déclenche son comportement de plongée. Son anxiété décroît et il peut alors consommer la croquette. Il est ainsi amené à réagir plus rapidement dans les situations futures. La figure 4.7 présente l'évolution de l'anxiété et de la faim de l'agent au cours de l'exécution de la simulation. Elle montre que cette anxiété chute très rapidement vers 0 et que la faim de l'agent est quasiment nulle. Il y a donc une adaptation individuelle de l'agent à la situation.

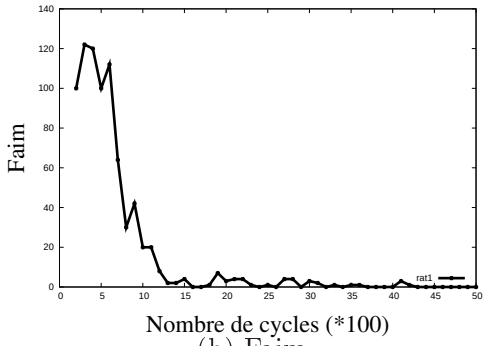
La partie suivante se concentre sur l'adaptation collective qui peut être observée dans le système lorsque plusieurs agents sont introduits simultanément dans l'environnement.

4.5.2 Adaptation collective et spécialisation

Nous avons conduit plusieurs exécutions du modèle Hamelin . Les paramètres initiaux des simulations qui ont été conduites sont présentés dans la tableau 4.1. Certains des paramètres sont globaux comme $\delta_\theta, \delta_{dom}, \delta_f$ et ont été déterminés empiriquement pour observer le phénomène de



(a) Anxiété



(b) Faim

FIG. 4.7 – Adaptation individuelle

nombre d'agents	δ_θ	δ_s	δ_f
6	0.95	0.15	0.00001
τ	η	$\theta_{initial}$	$s_{initial}$
10	2	600	100

TAB. 4.1 – Paramètres des simulations

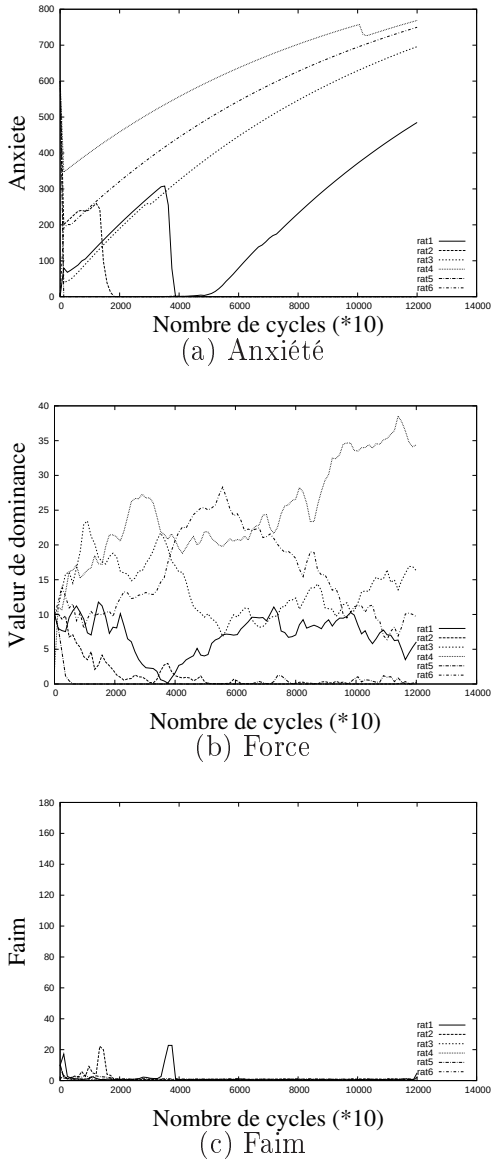
spécialisation parmi des agents initialement homogènes. Les autres paramètres (η, τ, $\theta_{initial}$ et $dom_{initial}$) représentent les conditions à partir desquelles les expériences sont conduites et seront amenés à être modifiés pour certaines expériences.

4.5.2.1 Observation de la spécialisation

A l'issue d'une expérience conduite avec les paramètres du tableau 4.1, le groupe d'agents peut être divisé en deux sous-groupes :

Le premier est constitué des agents avec une anxiété faible et une valeur de dominance faible. Ces agents ont tendance à plonger, à perdre les combats lors des tentatives de vols contre les agents appartenant au second sous-groupe. Ils correspondent aux rats transporteurs.

Le second sous-groupe est constitué d'agents avec une anxiété importante et une valeur de dominance élevée. Ces agents ne plongent jamais, et parviennent à satisfaire leurs besoins en volant les croquettes des agents du premier sous-groupe. Ils correspondent aux rats non-transporteurs.



(a) Anxiété

(b) Force

(c) Faim

FIG. 4.8 – Résultats obtenus par simulation pour une expérience

La figure 4.8 montre les résultats obtenus à partir d'une simulation. La courbe a) présente l'évolution de l'anxiété des agents au cours du temps. Lorsque la valeur est importante, cela signifie que l'agent à une anxiété importante et n'est pas incité à plonger. En outre, il est possible de déduire les comportements des agents à partir de cette courbe : si l'anxiété d'un agent croit au cours du temps, cela signifie que l'agent ne plonge pas. Au contraire, une chute de l'anxiété d'un agent apparaît quand cet agent émet l'item comportemental de plongée. La courbe b) montre les valeurs de dominance des agents et leurs évolutions. Initialement, les agents disposent de la même valeur de dominance, puis les interactions conduisent à une amplification des petites différences. La courbe c) présente enfin l'évolution de la faim des agents au cours du temps.

Sur les courbes de la figure 4.8, nous pouvons constater que quatre agents (numérotés 1, 3 , 4 et 6) ont une anxiété élevée (Fig 4.8(a)) et un rang élevé dans la structure hiérarchique (Fig 4.8(b)). Ces quatre agents appartiennent donc au groupe des agents non transporteurs. Deux agents ont une anxiété faible, ont tendance à plonger pour aller chercher de la nourriture et appartiennent au groupe des agents transporteurs. La faim de tous les agents est contrôlée et proche de 0 sauf pour certains événements apparaissant dans la simulation (les inversions de profil) que nous détaillerons par la suite.

Avec les valeurs initiales présentées dans le tableau 4.1, la taille des sous-groupes sur plusieurs expériences évolue peu. Sur 100 expériences conduites, deux états globaux du système ont pu être observés : un état global constitué de 3 agents transporteurs et 3 agents non transporteurs qui apparaît dans environ 60% des cas et un autre état global constitué de 4 agents non transporteurs et 2 agents transporteurs qui apparait dans environ 40% des cas. La stabilité des organisations montrent effectivement la présence d'une spécialisation entre individus. De plus, dans toutes les expériences, la faim des agents reste bornée et relativement basse.

Il est donc possible d'affirmer que le système parvient à reproduire une adaptation collective : certains agents parviennent à éviter d'affronter l'élément liquide, le besoin de toute la collectivité (correspondant à l'ensemble des besoins individuels) est satisfait et la viabilité du système est reproduite.

4.5.2.2 Stabilité

Les expériences ont pu montrer que les profils associés aux agents peuvent évoluer au cours du temps. Cependant, à chaque fois qu'un changement de profil a lieu, il est souvent compensé par une autre modification de profil. Cela assure une taille constante dans les sous-groupes. Pour les 100 expériences qui ont été conduites, un nombre moyen de 2.1 inversions de profils ont pu être observé au cours d'une expérience (écart type de 1.22). Ces inversions sont la conséquence de la fonction que l'on a choisi d'utiliser pour calculer la probabilité des résultats d'une agression entre agents.

En effet, cette fonction correspond au rapport des forces des agents impliqués et conduit à une organisation instable dans le système pour laquelle les positions des agents dans la hiérarchie de dominance évoluent au cours du temps.

Dès lors, un agent dont la position dans la hiérarchie a chuté n'arrive plus à accéder à la nourriture en effectuant des vols et doit maintenant plonger dans l'eau pour aller chercher de quoi s'alimenter. Simultanément, l'agent dont la position a augmenté peut maintenant accéder à des croquettes en les volant à ses congénères dotés d'une valeur de dominance plus faible. Ce changement de profil s'accompagne d'une augmentation temporaire des faims puisque l'agent dont la position a diminué doit se réadapter à sa nouvelle situation.

4.5.3 Re-différenciation

La spécialisation apparaît encore si l'on modifie les valeurs initiales des agents introduits dans le système, afin de simuler des agents ayant été préalablement différenciés.

Quand le système ne contient initialement que des agents transporteurs (faible anxiété initiale de 50 et faible force initiale de 1), une spécialisation peut encore être observée même si tous les agents parviennent initialement à accéder à la nourriture (cf figure 4.9). On observe une séparation identique en deux sous-groupes, chacun étant caractérisé par les mêmes profils. Les tailles de ces groupes sont en outre globalement les mêmes que celles observées dans les expériences précédentes. Trois états globaux ont été détectés : un est constitué de 4 agents non transporteurs et 2 agents transporteurs et apparaît dans 30% des cas, un second est constitué de 3 agents non transporteurs et 2 agents transporteurs et apparait dans 60% des cas, un dernier est constitué de 2 non transporteurs et 4 transporteurs et apparait dans 10% des situations. Le nombre d'inversions de profils augmente aussi drastiquement (nombre moyen par experience d'environ 4.8).

Ces expériences montrent que le phénomène d'adaptation collective conduit au même état global caractérisé par la présence de deux sous-groupes de taille similaire indépendamment du nombre d'agents introduits dans le système et met en évidence la robustesse du mécanisme d'adaptation.

4.5.4 Hamelin pour les éthologues

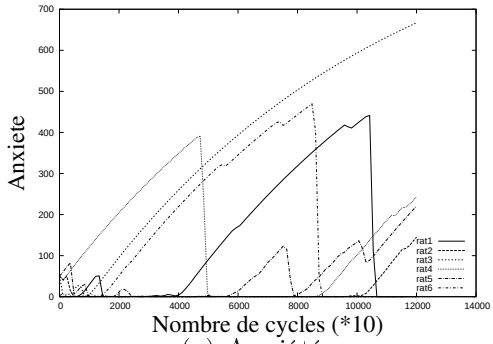
La simulation a permis de reproduire la spécialisation qui a été observée dans des expériences réelles. La spécialisation reproduite de manière artificielle est caractérisée par les mêmes profils, les mêmes proportions globales et la capacité des agents à se re-différencier. Enfin, comme les faims des agents restent bornées et proches de 0, le système est supposé parvenir à reproduire la viabilité du système biologique.

Hamelin a montré que la reconnaissance individuelle ou la cognition sociale n'est pas nécessaire (même si elle est intégrée en partie dans les valeurs de dominance) pour expliquer la spécialisation observée dans des groupes de rats confrontés à des contraintes environnementales. Hamelin a aussi montré que des règles de renforcement analogues à celles observées dans d'autres communautés animales peuvent expliquer le phénomène. De plus, Hamelin parvient à reproduire les deux types d'adaptation observées : l'adaptation individuelle quand un agent est introduit seul dans le dispositif et l'adaptation collective lorsque plusieurs agents sont mis en présence. Il permet de faire le lien entre ces deux adaptations.

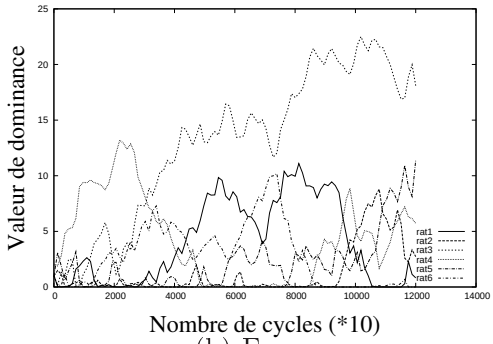
Hamelin constitue donc un premier pas vers la compréhension des mécanismes biologiques responsables de l'apparition d'une différenciation sociale dans une situation de "diving for food". Le lecteur intéressé pourra consulter l'article [CTB+05] qui fournit plus de renseignements à ce sujet.

4.6 Autres propriétés d'Hamelin

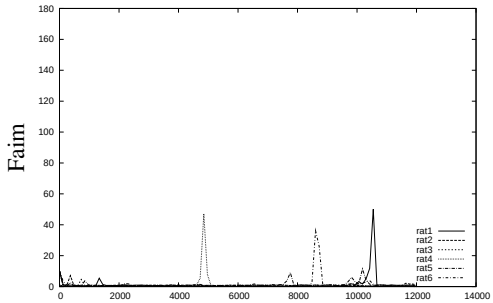
Le système Hamelin peut aussi être vu comme un système multi-agents qui génère une structure globale (la spécialisation) à partir de règles individuelles locales : les items comportementaux des agents. Afin d'avoir une meilleure évaluation des capacités d'adaptation de ce modèle, et pour avoir une meilleure compréhension des mécanismes mis en œuvre, d'autres expériences qui n'ont pas été entreprises avec de vrais rats ont été conduites sur le modèle.



(a) Anxiété



(b) Force



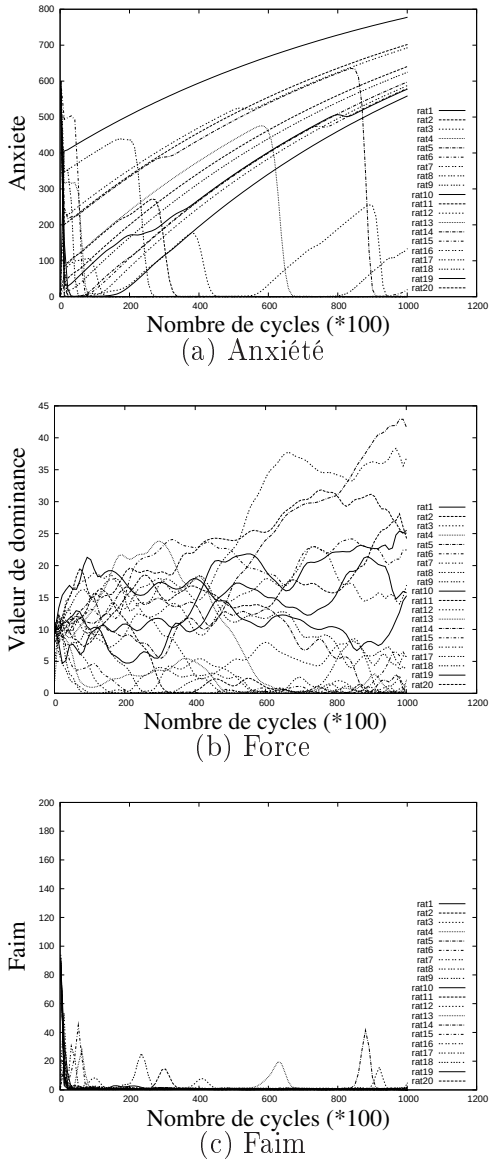
(c) Faim

FIG. 4.9 – Résultats obtenus par re-différenciation

4.6.1 Adaptation au nombre d'agents

Nous nous sommes concentrés sur l'influence du nombre d'agents dans la différenciation. Nous avons ainsi entrepris des expériences avec 20 agents (cf Figure 4.10). Les agents n'ont toujours pas de cognition sociale et ne peuvent pas reconnaître les autres agents du système. Néanmoins, la différenciation peut à nouveau être observée et la somme des faims des agents du système reste très basse sauf lors des inversions de profils qui sont désormais beaucoup plus fréquentes qu'auparavant.

L'état global dépend beaucoup des expériences qui ont été conduites, mais le nombre d'agents transporteurs reste majoritairement compris entre 12 et 14. Cela montre que le système parvient à s'adapter à l'environnement social des agents sans avoir besoin d'une représentation globale des agents et de l'environnement.



(a) Anxiété

(b) Force

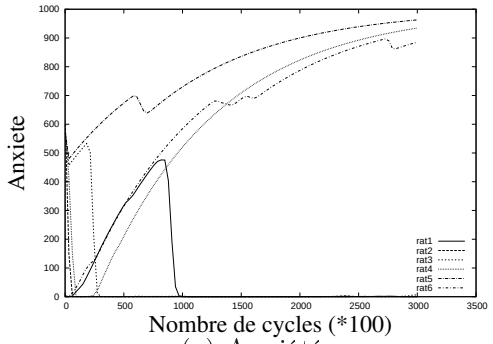
(c) Faim

FIG. 4.10 – Adaptation au nombre d'agents (passage à 20 agents)

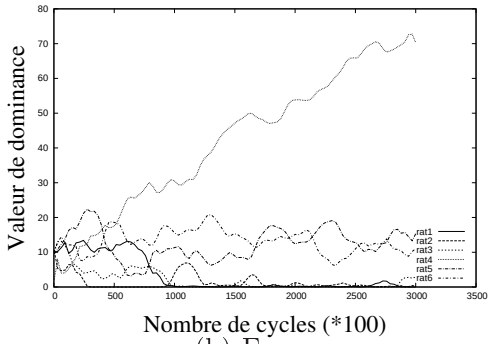
De plus, l'ajout d'agents à l'exécution du système ne remet pas en cause la spécialisation et le système parvient à se ré-adapter à la nouvelle situation.

4.6.2 Adaptation aux conditions extérieures

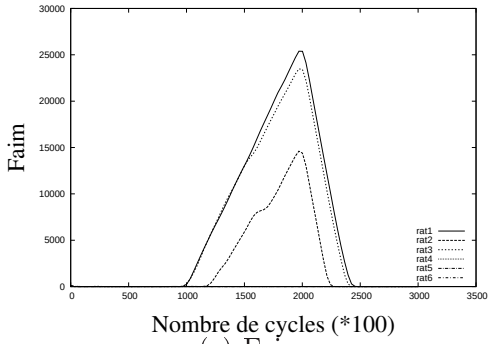
Nous nous sommes aussi intéressés à la manière dont le système réagit lorsque les caractéristiques de l'environnement évoluent durant l'exécution du système. Des simulations ont été conduites à partir des valeurs de paramètres du tableau 4.1. Les premiers 100000 pas de temps, ces paramètres restent inchangés. Après cette date, la valeur de τ est modifiée de 10 à 2. Cette nouvelle valeur correspond à une contrainte forte du système. Quand un agent atteint la mangeoire et prend une croquette, celle ci est désormais petite et contient beaucoup moins d'énergie. En conséquence, pour satisfaire les besoins de la collectivité, il est nécessaire que plus d'agents



(a) Anxiété



(b) Force



(c) Faim

FIG. 4.11 – Adaptation aux conditions extérieures au cours de l'exécution (au pas de temps 100000, τ est fixé à 2, et au pas de temps 200000, il est à nouveau fixé à 10)

adoptent un profil de transporteur.

Dans les expériences que nous avons conduites, nous avons pu observer une re-différenciation lorsque la valeur de τ est modifiée (cf fig 4.11). Dans cette expérience, un agent, le premier rat, qui était non transporteur devient transporteur. [5] Son anxiété chute, signifiant qu'il plonge dans l'eau et renforce ce comportement. De plus, ce changement de profil n'est pas compensé par l'apparition d'un agent non transporteur. L'état global du système a donc changé en réponse

[5]Pour plus de lisibité les courbes ont été lissées. Ceci explique le décalage qu'il est possible d'observer dans le changement de profil de l'agent correspondant au rat 1 : alors que l'agent change de profil au pas de temps 100000, cet événement apparaît légèrement plus tôt sur les courbes

aux variations des contraintes environnementales.

Cependant, cette re-différenciation n'est pas suffisante pour permettre à la collectivité de disposer d'assez de nourriture : la faim des agents augmente linéairement. Ceci provient de l'augmentation des contraintes d'accès à la nourriture dans la nouvelle situation.

Après le pas de temps 200000, la variable τ est affectée à son ancienne valeur de 10, les agents parviennent à nouveau à satisfaire leurs besoins et leurs faims décroît vers 0.

Le système parvient donc à se réadapter à des conditions environnementales changeantes dans une certaine mesure. Si les contraintes deviennent par contre trop importantes, la dynamique du système ne parvient pas à trouver une réponse adaptée.

4.6.3 Adaptation de l'organisation à la tâche

Dans certains cas, la faim des agents croît linéairement et remet en cause la viabilité de la collectivité. Ceci est dû à l'absence de transmission d'information au sein du système. Dans ces situations, les agents transporteurs parviennent à aller chercher assez de nourriture pour réduire la faim des agents non transporteurs mais ne disposent pas d'assez de temps pour pouvoir se nourrir. A cause des perceptions partielles des agents, ce genre de situation ne peut pas être détecté par les agents non-transporteurs qui ne sont donc pas incités à modifier leur comportement.

Comme nous nous intéressons désormais au système en lui-même hors considérations biologiques, il est possible d'envisager des modifications dans le processus d'organisation pour obtenir des capacités d'adaptation plus importantes. Nous nous sommes intéressés à lier les résultats des combats à la faim des agents qui y participent. Ceci constitue un moyen permettant de transmettre l'information manquante entre les agents au cours des interactions.

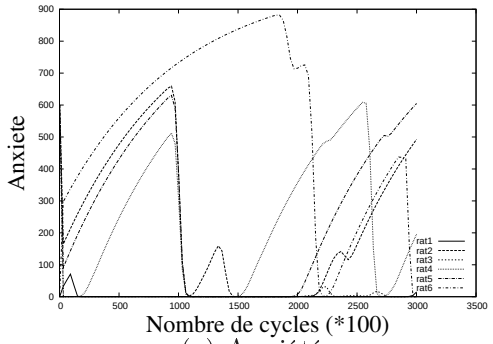
Ainsi, nous avons modifié la probabilité de gagner un combat en fonction de la faim de la victime potentielle. Si la faim de l'agent agressé est supérieure à un seuil donné a priori, la tentative de vol se solde toujours par un échec. Cependant, comme les renforcements sont toujours appliqué, ces situations conduisent à des inversions importantes de valeur de dominance permettant dorénavant à l'agent d'accéder à de la nourriture en volant les autres agents.

Quand cette modification des interactions est implémentée dans le système (avec une valeur de seuil de 100), le problème de l'augmentation linéaire de la faim disparaît et le système se réadapte lui-même de manière plus efficace.

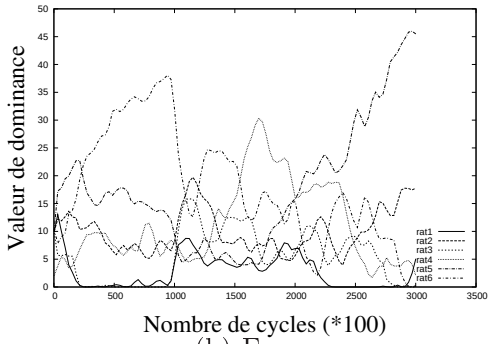
La figure 4.12 présente les résultats obtenus pour le même scénario que la figure 4.11. Désormais, le système parvient à se réadapter au pas de temps 100000. A cet instant, les contraintes environnementales deviennent très importantes et tous les rats excepte un deviennent des rats transporteurs. Le nombre de rats transporteurs est alors suffisant pour satisfaire tous les besoins du groupe. Quand, après 200000 pas de temps, la taille des croquettes devient plus importante, le système se réadapte à un état proche de l'état global précédent.

4.6.4 Bilan des propriétés d'Hamelin

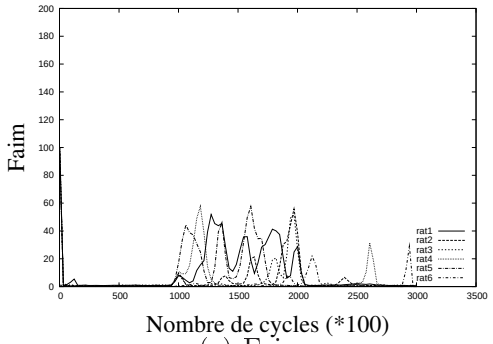
La simulation Hamelin peut être considérée comme un système complexe inspiré par des observations biologiques. Ce système parvient à organiser les comportements des agents à l'exécution selon les interactions qui se produisent dans le système. Cette organisation est la conséquence des observations partielles des agents et des règles de renforcement individuelles. Elles permettent au système de se réadapter à des conditions diverses sans avoir besoin d'une représentation explicite



(a) Anxiété



(b) Force



(c) Faim

FIG. 4.12 – Spécialisation observée après l'ajout d'un retour dans les interactions de combat

de ces changements.

Le système présente donc un certain nombre de propriétés adaptatives : il parvient à se réadapter à des conditions extérieures, au nombre d'agents du système et à différentes conditions initiales comme cela a pu être mis en évidence dans les expériences concernant la re-différenciation.

4.7 Discussion

Pour le moment, nous avons présenté le modèle Hamelin et les résultats qu'il est possible d'obtenir avec ce modèle. Notre objectif est de mettre en évidence des mécanismes utiles à la résolution décentralisée de problèmes collectifs.

Nous allons analyser ici plus précisément le modèle Hamelin pour essayer de mettre en évidence les mécanismes d'organisation sous-jacents afin de les abstraire et de pouvoir les proposer dans le cadre formel DEC-POMDP dans le chapitre suivant.

4.7.1 Interprétation de Hamelin

Le modèle Hamelin est caractérisé localement par un besoin à satisfaire (la faim de chaque rat) et par une difficulté d'accès à la nourriture nécessaire à la satisfaction de ce besoin. Chaque agent a ses propres besoins qu'il est le seul à percevoir et peut accéder à la ressource de deux manières différentes : en volant une croquette aux autre agents ou en allant chercher une croquette de la mangeoire.

Initialement, les individus sont indifférenciés, ils n'ont pas de moyen privilégié d'accès à la ressource. La dynamique collective du système basée sur des règles d'adaptation purement locales conduit à une organisation entre les agents pour laquelle des moyens privilégiés d'accéder à la nourriture émergent : certains agents préfèrent accéder directement à la nourriture, d'autres agents préfèrent l'obtenir de manière indirecte en interagissant directement avec les autres agents.

Le processus de différenciation est la conséquence d'un couplage de deux modèles de renforcements :

- le premier modèle inspiré des réponses à seuil adaptatif contrôle l'accès direct à la ressource.
- le second inspiré par les relations de dominance contrôle l'accès indirect par des interactions représentées de manière distribuées au sein des agents par la notion de valeur de dominance.

Le couplage de ces deux mécanismes est responsable de la dynamique du système et de l'attribution de profils au sein des agents.

Les relations de dominance permettent la distribution des croquettes entre les agents. Elles constituent un moyen permettant de transférer les croquettes des agents les plus faibles vers les agents les plus forts. Comme la croquette constitue la ressource permettant aux agents de satisfaire leurs besoins, le transfert de croquette peut être aussi compris comme un transfert de besoin au sein de la collectivité allant des agents les plus forts vers les agents les plus faibles.

Les réponses à seuil permettent aux agents les plus faibles qui concentrent les besoins de la collectivité d'adapter leur comportement pour répondre à leur faim. Ce mécanisme d'adaptation individuel permet ainsi à l'agent de répondre aux besoins de la collectivité qu'il intègre dans sa faim.

Hamelin peut alors être compris comme un couplage entre un processus organisationnel capable de répartir les stimuli internes (faims des agents) au sein de la collectivité et un processus d'adaptation local qui permet la satisfaction des besoins individuels. Le transfert d'informations locales et la possibilité de résoudre localement une partie du problème par l'agent le plus expérimenté (celui qui plonge le plus rapidement) aboutit à l'adaptation de l'ensemble des agents donc du système sans avoir recours à une représentation explicite des besoins globaux.

4.7.2 Positionnement du modèle Hamelin

On peut noter que le modèle Hamelin est proche du modèle satisfaction-altruisme de [Sim01]. Les travaux de [Sim01] traitent d'agents devant explorer un environnement. La méthode proposée pour atteindre des comportements coopératifs est de coupler un modèle de sélection d'action

à un modèle d'interaction entre les agents. Le modèle d'interaction est basé sur des signaux de satisfaction et sur la pression sociale exercée par la communauté sur les agents : si la satisfaction d'un agent est faible, il peut forcer les autres agents à libérer le passage.

Les interactions présentées dans le modèle satisfaction-altruisme peuvent être comparées avec les relations de dominance du modèle Hamelin et le modèle de sélection d'action avec le modèle de réponse à seuil. Le modèle satisfaction altruisme et le modèle Hamelin sont fondés sur le même principe abstrait consistant à coupler un modèle d'interaction permettant de transmettre des informations locales et un modèle de sélection d'actions permettant de résoudre localement la tâche.

Cette comparaison nous laisse penser que ce principe général peut être réutilisé dans de nombreuses autres applications pour atteindre des comportements coopératifs entre les agents en se limitant à considérer des situations locales.

Le chapitre suivant va se charger d'extraire ces mécanismes que nous réutiliserons par la suite dans le formalisme que nous proposons

4.7.3 Abstraction des mécanismes à la base de Hamelin

Hamelin est un système qui est fondamentalement décentralisé :
- chaque agent dispose d'un état interne
- les décisions prises par un agent sont purement individuelles
- les agents ont une perception partielle de leur environnement
- ils peuvent émettre des actions et des interactions directes

La résolution des combats permet néanmoins d'effectuer des prises de décisions plus collectives :
- le déclenchement d'un combat est décidé de manière individuelle par un agent. Ici le combat est déclenché systématiquement en fonction de ses perceptions [6], mais d'autres approches sont facilement envisageables.
- l'interaction est résolue entre un nombre réduit d'agents
- le choix de ce déclenchement s'accompagne du choix du protagoniste avec lequel un agent va combattre (ici déterminé de manière aléatoire)
- l'issue du combat est déterminée à partir des forces relatives des agents. A partir d'un nombre réduit de variables distribuées au sein de la collectivité (une valeur de dominance par agent), les issues d'un combat peuvent être calculées. Cette résolution peut alors être faite de manière semi-locale : les agents s'échangent leur valeur de force, un des deux effectue la décision stochastique et le choix est entériné et modifie l'état du système
- il est possible de lier la force de chacun des agents avec des données du problème, comme nous l'avons montré. Dans ce cas, l'interaction a pour objectif de prendre une décision collective en tenant compte des états internes de chacun des agents. Le choix de résolution d'une interaction dépend alors des perceptions qu'a l'autre agent de son état auxquelles l'agent n'a pas accès mais qui modifient sa force (comme cela peut être la cas pour la faim), des données comportementales de l'autre agent (comme sa capacité à accéder à de la nourriture par vol, on peut éventuellement lier la force à sa capacité physique et au coût individuel à aller chercher de la nourriture), de l'historique des actions de l'autre agent

[6] Ces perceptions correspondent aux croquettes dans la cage.

Par l'échange de valeur de dominance, l'interaction permet donc de transmettre de manière implicite de l'information concernant les perceptions des agents, les capacités comportementales des agents ou l'historique des actions d'un agent. En cela, l'interaction directe nous semble un moyen de prendre des décisions collectives, de manière distribuée à partir de communications locales pour la résolution d'une tâche.

4.8 Bilan du Chapitre

Dans le chapitre précédent, nous avons présenté les modèles markoviens et avons mis en évidence la nécessité de formaliser la notion d'interaction. Nous avons décidé de nous concentrer sur l'interaction directe et nous sommes demandés comment instancier ce concept dans un système pour en tirer parti. La démarche que nous avons suivie a consisté à étudier avec des éthologues des systèmes naturels dans lesquels il est possible d'observer un comportement collectif issus des interactions ayant lieu entre les individus et différent de la simple superposition de comportements individuels.

Dans ce chapitre, nous avons présenté un phénomène collectif de spécialisation observé dans des groupes de rats devant faire face à des contraintes environnementales. Les expériences biologiques qui ont été menées suggèrent qu'un processus de régulation robuste était en oeuvre et que les individus adoptaient un profil en fonction de leur environnement social. Nous nous sommes demandés quels mécanismes pouvaient expliquer l'apparition de cette spécialisation et plus particulièrement s'il était possible de reproduire le phénomène sans introduire de représentation complexe des autres.

Pour cela, nous avons développé un modèle original : le modèle Hamelin. Ce modèle constitue un processus auto-organisé basé sur des interactions locales entre agents et des processus d'adaptation individuels. Les expériences montrent qu'Hamelin parvient à reproduire qualitativement le phénomène de différenciation et l'adaptation collective du groupe caractérisés par la survie de l'ensemble des individus tout en limitant le nombre d'agents plongeurs. L'exécution du modèle conduit à une spécialisation fondée sur des profils analogues à ceux observés dans les expériences biologiques. Le modèle Hamelin parvient à reproduire les phénomènes de re-différenciation qui ont pu être observés dans la nature. Enfin, ce modèle est basé sur des agents réactifs sans représentation explicite des autres et prouve donc que des capacités cognitives complexes ne sont pas nécessaires pour expliquer la spécialisation apparaissant dans cette situation. Hamelin fournit donc de nouvelles pistes pour permettre à un agent d'intégrer son environnement social sans qu'il n'ait besoin de représentation complexe des autres agents du système.

De plus, la simulation peut être analysée indépendamment de considérations biologiques et s'avère avoir des propriétés adaptatives plus importantes : le système collectif parvient à se réadapter aux perturbations des conditions extérieures et du nombre d'agents.

Le coeur du modèle a été isolé : il s'agit d'un couplage entre un processus d'organisation qui se charge de répartir les besoins au sein de la collectivité et d'un processus d'adaptation individuel permettant de résoudre la tâche. Ces deux processus utilisent des règles de renforcement locales et nous semblent adaptés pour envisager de construire des processus d'apprentissage collectifs par renforcement. Cependant pour le moment, le modèle Hamelin est purement descriptif et ne

constitue pas un processus de résolution.

Le noyau du modèle Hamelin a néanmoins permis de répondre aux questions que nous nous étions posées initialement en conclusion du chapitre 3 :

• *Comment l'interaction directe peut elle être mise en oeuvre dans un système multi-agents autonome pour lequel toute initiative est d'origine individuelle ?* Dans Hamelin, l'utilisation d'une interaction directe est décomposée en deux phases. Dans la première phase, un agent prend la décision d'interagir avec un autre agent. Dans la seconde phase, un échange des valeurs de dominance permet de prendre une décision collective. Ainsi, même si l'interaction consiste à prendre une décision à deux, l'interaction est d'initiative individuelle et respecte les principes de l'autonomie telle que nous l'avons définie.

• *Comment intégrer une composante sociale et prendre en compte le comportement d'autres agents à partir d'agents réactifs disposant de capacités cognitives réduites et sans représentation complexe ?* Le modèle Hamelin a montré qu'il n'est pas nécessaire de disposer d'un modèle complexe de l'autre agent pour prendre des décisions utiles à la collectivité. Dans le cas d'Hamelin, ces décisions sont prises au moment de la résolution des agressions. Nous avons montré qu'en modifiant la manière dont sont résolues les agressions (cf 4.6.3), il est possible de transmettre implicitement de l'information entre les agents et d'adapter les comportements collectifs à la tâche à résoudre. Le coeur d'Hamelin réside dans le fait que les résultats des interactions qu'un agent souhaite entreprendre sont conditionnés par l'état et le comportement de l'autre agent avec lequel il interagit.

• *Comment adapter localement les comportements pour permettre de construire une solution collective plus performante que la simple juxtaposition de comportements individuels ?* Le modèle Hamelin montre que des renforcements locaux et des confrontations de variables individuelles (valeurs de dominance) sont suffisants pour tirer parti de mécanismes d'adaptation individuelle et pour construire une réponse collective caractérisée par la présence d'agents non-plongeurs. En outre, d'autres expériences ont montré qu'il est possible d'adapter la résolution des interactions pour obtenir un système capable de s'adapter à des perturbations importantes.

La question à laquelle nous allons répondre dans le chapitre suivant sera de savoir comment réutiliser ces mécanismes dans un cadre générique et les adapter à une tâche définie a priori. Dans un premier temps (cf chapitre 5), notre objectif va être de proposer un cadre formel dans lequel l'interaction sera définie comme un moyen mis en oeuvre dans le système et manipulable par les agents. Dans un second temps (cf chapitre 6), nous nous intéresserons à des solutions algorithmiques permettant d'utiliser ces interactions à bon escient.

Le système Hamelin constituera notre guide pour proposer une alternative au DEC-POMDP afin de construire des processus permettant d'organiser les agents et de tirer parti de capacités d'adaptation à une tâche collective perçue localement par chaque agent.

Chapitre 5

Formalisme Interac-DEC-POMDP

Avant d'entamer cette partie, nous proposons d'effectuer un bilan succinct des éléments que nous avons présentés dans les parties précédentes afin de re-expliciter notre démarche.

En introduction, nous avons explicité notre objectif consistant à construire de manière entièrement distribuée des systèmes multi-agents réactifs pour résoudre automatiquement des problèmes posés à la collectivité et avons choisi de nous concentrer sur des approches formelles qui permettent d'automatiser une partie du processus de construction.

Dans le chapitre 2, nous avons mis l'accent sur ce qui constitue un Système Multi-Agents Réactif en terme de concept. Nous avons présenté le concept d'agent isolé et nous sommes intéressés aux conséquences de la présence de plusieurs agents autonomes dans un système. Cela nous a amené à repréciser la notion de rationalité et d'autonomie dans un cadre multi-agents et à présenter les différentes facettes du concept d'interaction au coeur des SMAs. Nous en sommes arrivés à la nécessité pour les agents de disposer de compétences sociales pour intégrer la présence des autres agents dans leurs prises de décision.

Dans le chapitre 3, nous nous sommes concentrés sur un cadre formel permettant d'instancier les concepts présentés dans la partie précédente. Nous avons été amenés à considérer les processus de décision markoviens décentralisés qui permettent de formaliser le problème de construction de comportements d'agents réactifs autonomes en interaction. Nous avons analysé la manière dont le concept d'interaction entre agents a été instancié et mis en évidence la présence des seules interactions indirectes. Nous avons enfin décrit les techniques de résolution proposées dans la littérature et insisté sur le fait que ces différentes techniques présentent toujours un aspect centralisé à des niveaux divers (observabilité, communication, etc...) pour pouvoir considérer les interactions entre agents définies à un niveau global dans la matrice de transition. Comme ces approches ne répondent pas totalement à nos attentes, nous allons chercher à proposer un nouveau cadre formel permettant d'intégrer les interactions directes pour structurer les relations entre les agents et permettre à ceux-ci d'intégrer la présence d'autres agents dans le système.

Dans le chapitre 4, afin de nous guider pour la production d'un cadre formel, nous nous sommes inspirés de systèmes naturels pour lesquels des individus parviennent à s'organiser à partir d'interactions locales pour résoudre un problème perçu localement par les individus. Nous avons développé un système, appelé système Hamelin inspiré de la biologie. Ce système présente des capacités d'adaptation collective issues de processus d'adaptation individuels. Des expé-

riences effectuées sur ce modèle ont permis de mettre en évidence l'utilité de l'interaction directe comme confrontation de variables internes pour adapter le système à partir de règles d'adaptation individuelles. Ce modèle constitue une instanciation particulière de l'utilisation du concept d'interaction directe et présente les propriétés attendues pour nos systèmes : des capacités d'adaptation collectives construites à partir de processus d'adaptation individuels. Afin de construire des comportements collectifs à partir de règles d'adaptation individuelles, nous proposons de réintroduire cette instanciation du concept d'interaction directe dans la cadre des formalismes de Markov en tant qu'élément de premier ordre du formalisme. L'objectif à long terme de ce nouveau cadre formel est de proposer des processus génériques et décentralisés de construction de comportements collectifs réutilisables pour des problèmes variés.

L'objectif de ce chapitre est de faire la synthèse de ces éléments afin de proposer un nouveau cadre formel appelé Interac-DEC-POMDP.

Dans ce chapitre, nous allons donc décrire le nouveau cadre formel original des Interac-DEC-POMDP qui a pour objectif
- de formaliser la notion d'interaction directe et de la représenter explicitement
- de décrire les systèmes l'utilisant conjointement à des actions

Le chapitre 6 se chargera plus spécifiquement
- de proposer une classe de problème dans le cadre Interac-DEC-POMDP pour mettre en évidence l'utilité de la représentation explicite de la notion d'interaction
- de proposer des algorithmes permettant de construire de manière entièrement décentralisée des comportements collectifs dans un système complexe partiellement observé.

5.1 Présentation de l'Interac-DEC-POMDP

5.1.1 Objectif du formalisme Interac-DEC-POMDP

Le problème initial que nous nous sommes fixés consistait à concevoir des systèmes multi-agents distribués et a été énoncé par Ferber comme consistant à répondre aux questions suivantes :
- Quelle est l'architecture de l'agent sachant que le comportement de l'agent dépend de cette architecture ?
- Quelles sont les formes d'interaction permettant à plusieurs agents de maximiser leur satisfaction ?
- Comment faire évoluer le comportement des agents de manière à ce qu'ils puissent tirer parti des expériences passées ?
- Comment réaliser et implémenter de tels systèmes ?

Nous avons séparé ce problème en deux sous-problèmes :
- le premier est le sous-problème consistant à décrire dans un formalisme les architectures des agents, et les formes d'interaction possibles
- le second est le sous-problème de construction des comportements consistant à construire et à faire évoluer les comportements des agents afin de répondre au mieux au problème posé.

Nous nous sommes focalisés sur une question bien particulière, à savoir est-il possible de construire des comportements collectifs à partir de processus de construction de comportements individuels et ce de manière entièrement décentralisée ?

Le cadre formel a pour objectif de représenter des problèmes posés dans les systèmes multi-agents en permettant l'utilisation des deux types d'interaction mis en évidence dans la partie 2 : l'interaction directe et l'interaction indirecte. Il constitue une réponse au premier sous-problème et va conditionner la réponse que l'on fournira au deuxième sous-problème : à savoir le problème de construction des comportements des agents.

Comme le formalisme DEC-POMDP présenté dans le chapitre 3 constitue déjà un cadre pour représenter des problèmes de prise de décision distribuée, le modèle que nous proposerons sera basé sur ce formalisme. Néanmoins, les processus de décision markovien décentralisés ne permettent aux agents que d'effectuer des interactions indirectes et ces interactions ne sont pas représentées localement.

L'interaction que nous allons réintroduire dans ce modèle sera une interaction inspirée de celle qui a pu être construite dans le modèle Hamelin décrit dans le chapitre 4 : il s'agit d'une décision collective prise par concertation ayant pour objectif de modifier l'état global du système localement aux agents impliqués dans l'interaction.

Pour cela, nous allons chercher à répondre aux question suivantes dans ce chapitre :
- Comment représenter l'architecture externe des agents, à savoir les senseurs, et les effecteurs ?
- Comment formaliser la nouvelle possibilité donnée aux agents d'interagir directement entre eux ?
- Comment représenter un problème posé à la collectivité ?
- Comment décrire les comportements des agents et l'utilisation des possibilités qui leurs sont offertes ?
- Comment exécuter le système à partir de tous ces éléments ?

Le formalisme original Interac-DEC-POMDP que l'on propose dans ce chapitre constituera notre réponse à ces questions.

5.1.2 Systèmes à représenter

Avant de décrire le formalisme, nous allons présenter les caractéristiques des systèmes que l'on souhaite représenter dans ce formalisme. Nous souhaitons que nos systèmes présentent de fortes contraintes de localité qui leur permettront d'être utilisées dans des applications réelles.

Les **perceptions** des agents sont limitées dans l'espace et les actions émises par un agents n'ont que des conséquences locales (au sens d'une topologie dépendante du problème). Les perceptions limitées impliquent en outre qu'un agent ne dispose que d'une évaluation locale de la tâche globale en cours et qu'il n'a pas la possibilité de construire une représentation globale du système.

Nous souhaitons de la même manière que les **communications** entre agents soient limitées en terme de nombre d'agents et au sens d'une topologie définie dans le cadre du problème. Ces considérations ont pour conséquence que les agents ne peuvent pas constamment échanger des informations entre eux et doivent donc dans certaines circonstances pouvoir décider d'une action à partir d'informations limitées.

Nous souhaitons néanmoins que les agents puissent échanger localement des informations au cours des interactions directes comme cela peut être le cas dans le système Hamelin. Ces in-

teractions seront des **interactions locales** : pour une configuration donnée du système, seules certaines interactions seront possibles du fait des distances entre les agents.

Enfin, nous souhaitons qu'il soit possible de **restructurer les relations** entre agents découlant des interactions directes. Contrairement à des systèmes fondés sur un réseau donné a priori définissant les relations entre agents (comme cela est par exemple le cas dans les approches proposées par Guestrin [Gue03] ou les fonctions de valeur distribuées [SWMR99]), nous souhaitons que le réseau de relations se forme à l'exécution sur la base des interactions possibles. Les agents avec lesquels un agent interagit doivent pouvoir changer au cours de l'exécution du système en fonction des circonstances, de l'évolution de la tâche et des motivations des agents du système. De plus, ces relations entre agents sont à l'initiative des agents eux-mêmes puisque nous souhaitons des systèmes autonomes au sens défini dans la partie 2, c'est a dire pour lesquels toute initiative est individuelle.

Maintenant que nous avons énoncé les systèmes sur lesquels nous nous concentrons, nous allons analyser en quoi Hamelin est un système vérifiant ces propositions et justifier pourquoi il peut constituer une source d'inspiration.

5.1.3 Inspiration du modèle Hamelin

Le système Hamelin est très proche du cadre formel que l'on souhaite construire.

Chaque agent est un agent réactif et ne dispose que d'une perception partielle de son environnement : il ne sait pas reconnaître les autres agents du système, et ne perçoit que ceux disposant d'une croquette. De plus, chaque agent ne connaît que son besoin et ne construit pas de représentation des besoins de la collectivité.

Les communications entre les agents sont limitées : les seules communications possibles sont les échanges de valeur de dominance qui ont lieu au cours des interactions directes entre agents lors d'une tentative de vol. Ces valeurs de dominance synthétisent ainsi au sein de l'agent son expérience des relations avec les autres agents du système et permet de construire une réponse collective à partir de mécanismes d'adaptation individuels.

Le système Hamelin parvient à organiser une société par l'intermédiaire de combats et d'échanges. Ces combats sont fondés sur la notion de valeur de dominance renforcée par les actions des agents au cours du temps et éventuellement dépendante de la tâche (cf force liée à la faim). L'interaction permet d'agencer les comportements des agents pour produire des structures collectives utiles.

Hamelin présente donc les caractéristiques des systèmes que l'on souhaite construire (excepté les contraintes topologiques concernant les possibilités d'utiliser certaines interactions). Dans les parties suivantes nous allons nous intéresser à la structure du système Hamelin qui nous servira de guide pour la construction du formalisme Interac-DEC-POMDP.

5.1.3.1 Deux modules

Tout d'abord, le système Hamelin est constitué de deux modules : un module d'action et un module d'interaction

Module d'action Le module d'action est exécuté lors de la première partie du cycle d'exécution du modèle Hamelin (cf partie 4.4.4) : il correspond aux actions des agents ne nécessitant pas d'échange d'informations. Il s'agit des items comportementaux de plongée et d'alimentation.

Au cours de l'exécution de ce module, les agents peuvent exercer des interactions indirectes : le fait qu'un agent-rat aille chercher une croquette, modifie les perceptions des agents dans la cage qui sont amenés à émettre d'autres actions par la suite.

Ce module d'action est en outre le lieu d'adaptation individuelle. Les agents renforcent leurs items comportementaux de plongée afin de pouvoir répondre plus rapidement à leur besoin.

Module d'interaction C'est au cours de l'exécution du module d'interaction qu'ont lieu les échanges fondés sur des interactions directes entre agents. C'est dans cette partie du système que les agents sans croquette sont sollicités pour tenter de voler des croquettes aux autres agents.

Le résultat d'une telle tentative dépend des autres agents. Ce module permet la structuration des échanges dans le système et permet de tirer parti de mécanismes d'adaptation individuels pour produire une adaptation collective du système.

Maintenant qu'on a décrit la structure globale du système Hamelin, nous allons nous intéresser à la manière dont l'interaction est représentée et executée.

5.1.3.2 Interaction directe dans Hamelin

Les agents du système Hamelin utilisent des interactions directes pour effectuer des échanges de croquettes.

L'utilisation d'une interaction se fait en deux phases : une phase de déclenchement et une phase de résolution de l'interaction.

Déclenchement Au cours de la phase de déclenchement, un agent décide individuellement en fonction de ses perceptions d'exécuter une tentative de vol vers un autre agent. Une interaction est déclenchée quand l'agent perçoit des croquettes dans la cage et l'agression est dirigée vers un possesseur de croquettes.

Résolution Dans une seconde phase nommée phase de résolution, les agents résolvent l'interaction pour décider de son résultat. Dans le modèle Hamelin, le résultat peut être un vol effectif de croquette ou une tentative échouée. La résolution est stochastique et dépend des valeurs de dominance relatives des agents impliqués. Au cours d'une interaction, les agents échangent leurs valeurs de dominance et une décision est prise par rapport à celles-ci sans qu'à aucun moment, il ne soit nécessaire aux agents d'avoir une vue globale du système.

Structure de l'interaction directe dans Hamelin Le principe d'une interaction directe dans Hamelin consiste donc à

- effectuer un choix collectif

- suite à un échange de signaux très simples
- entre un nombre réduit d'agents

Ce choix collectif consiste à décider à plusieurs d'une action jointe ponctuelle. Cette action est ponctuelle car elle est exécutée instantanément après la prise de décision et a pour objectif de modifier l'état du système.

Les interactions inspiré du modèle Hamelin que nous considérerons ont plusieurs composantes :

- un agent émetteur : l'agent à l'origine de l'interaction
- un agent receveur : l'agent vers lequel l'interaction est dirigée (potentiellement plusieurs)
- un moyen : des échanges de signaux numériques entre agents
- des possibilités : un ensemble d'actions jointes comme conséquences possibles de l'interaction
- une influence réciproque correspondant à une prise de décision collective
- un effet : une évolution de l'état du système dûe à une action jointe coordonnée choisie par les agents impliqués

Maintenant que nous avons précisé ce que l'on entendait par la notion d'interaction, nous allons nous concentrer sur le formalisme Interac-DEC-POMDP afin d'intégrer actions et interactions directes dans un cadre homogène et de représenter des catégories d'interactions fondées sur le même principe qu'Hamelin.

5.2 Description du formalisme Interac-DEC-POMDP

5.2.1 Agencement

Au formalisme DEC-POMDP, on souhaite rajouter la notion d'interaction. Ceci nécessite d'agencer les actions et les interactions à l'exécution du système.

Plusieurs agencements sont envisageables

- Il est possible d'avoir des actions et interactions concurrentes, un agent pouvant émettre soit une action soit une interaction
- Il est possible d'avoir des actions et des interactions sérialisées : les agents émettent leurs actions, émettent ensuite leurs interactions, et le processus se répète

Nous nous limiterons au cas le plus simple où actions et interactions sont sérialisées. Ce cas permet déjà de poser des problèmes intéressants comme nous le verrons par la suite et peut donc constituer une première approche pour atteindre à long terme des systèmes constitués par des interactions choisies en concurrence aux actions.

Ainsi, comme Hamelin, un Interac-DEC-POMDP est constitué de deux modules : un module d'action et un module d'interaction qui seront exécutés en série(cf fig 5.1).

Dans les parties suivantes nous décrirons ces modules et les autres composants d'un Interac-DEC-POMDP. Différentes parties de cette description ont déjà été présentées dans [TBC04a], [TBC04b] et [TCC05].

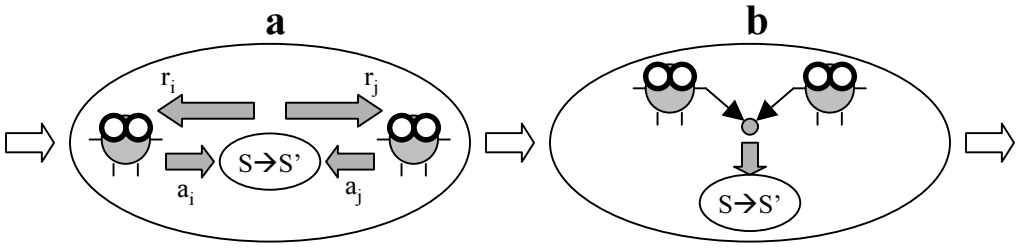


FIG. 5.1 – Présentation générale de l'Interac-DEC-POMDP : a) un module d'action et b) un module d'interaction executés en série

5.2.2 Interac-DEC-POMDP - Environnement et agents

On supposera par la suite un système constitué de n agents. Le symbole $agent_i$ désigne l'agent numéro i.

L'environnement du système est supposé décrit par un ensemble fini d'états S.

Enfin, la tâche est décrite par l'intermédiaire d'une fonction de récompense globale perçue localement par les agents (cf la description du module d'action). Les interactions directes ne génèrent pas de récompenses mais constituent un moyen permettant aux agents de modifier l'état global du système afin de générer des récompenses ultérieures.

5.2.3 Interac-DEC-POMDP - Module d'action

5.2.3.1 Objectifs

L'objectif du module d'action est de pouvoir représenter :
- les actions individuelles possibles des agents du système
- l'utilisation des actions par les agents via leur comportement
- l'évolution du système par rapport aux actions émises
- la résolution des conflits entre actions
- l'avancement de la tâche perçue localement par les agents

5.2.3.2 Formalisme

Dans le formalisme DEC-POMDP, un certain nombre d'éléments existent et nous semblent adaptés aux systèmes multi-agents :
- la notion d'action est clairement définie et la simultanéité des actions permet de représenter des systèmes d'agents en interaction comme le modèle influence-réaction [FM96].
- les modèles DEC-POMDP permettent aux agents d'exercer des influences indirectes entre eux par l'intermédiaire de l'environnement.
- Les modèles DEC-POMDP constituent un formalisme dans lequel il est possible d'exprimer des agents dotés d'observabilité partielle.
- la notion de problème multi-agents est définie dans ce cadre par l'intermédiaire d'une fonction de récompense globale.
- La notion de comportement d'un agent est clairement définie dans ce cadre et constitue un concept calculable.

Le module d'action est donc défini par un DEC-POMDP $< \alpha, S, A_i, T, \Gamma_i, O, R >$. Les comportements des agents sont caractérisés par leurs politiques individuelles $\pi_i : \Gamma_i \times A_i \rightarrow [0, 1]$

Enfin, nous supposerons que la fonction de récompense se décompose sous la forme de fonction de récompenses individuelles $R = \sum_i r_i$ et que chaque agent i peut observer $r_i : \Gamma_i \times A_i \times \Gamma_i \rightarrow \mathbb{R}$.

5.2.3.3 Algorithme d'exécution

L'exécution du module d'action est décrite par l'algorithme 5 et correspond à l'exécution d'un DEC-PODMP.

Algorithme 5 Executer le module d'action

Observation : $O = (o_0, ...o_n) \leftarrow O(s)$
pour tout agents $i \in [0..n]$ **faire**
  choix de l'action i : $a_i \leftarrow \pi_i(o_i)$
fin pour
Action jointe : $a \leftarrow (a_0, ..., a_n)$
Exécution de l'action jointe : $s' \leftarrow T(s, a)$
Récompense globale reçue par le système : $R \leftarrow R(s, a)$
Modification de l'état du système : $s \leftarrow s'$

5.2.4 Interac-DEC-POMDP - Formalisation de l'interaction

5.2.4.1 Description générale

Une interaction directe est définie comme une action mutuelle réciproque entre plusieurs agents. Elle aboutit à l'émission d'une action jointe décidée collectivement par ces agents. Cette action jointe modifie l'état global du système dans le voisinage des agents impliqués dans l'interaction.

Afin de respecter les principes de localité que nous avons énoncé auparavant, nous souhaitons
- que cette interaction soit déclenchée par un agent pour disposer d'un système centré agent, pour respecter les localités des prises de décision et pour limiter les communications dans le systèmes aux communications utiles (celles nécessaires pour la résolution de l'interaction).
- que cette interaction soit résolue à la suite de communications réduites entre agents proches
- que ces interactions n'impliquent que des communications ponctuelles et locales

Enfin, comme nous souhaitons pouvoir représenter des interactions qui peuvent restructurer les relations entre agents à l'exécution, une interaction n'implique pas toujours les mêmes agents, et un agent doit pouvoir décider avec qui il souhaite interagir.

De la même manière que dans le modèle Hamelin, nous distinguerons par la suite deux phases dans une interaction :
- la phase de **déclenchement** consistant pour un agent à décider quelle interaction directe déclencher et avec quel(s) agent(s) interagir. Cette phase est la conséquence d'une décision purement individuelle. Elle permet de restructurer les interactions puisque les agents impliqués dans l'interaction ainsi que leur rôle font partie de cette décision de déclenchement.

– la phase de **résolution** de l'interaction consistant à choisir collectivement une action jointe parmi celles proposées par l'interaction afin de faire évoluer le système. Cette action jointe sera appelé 'résultat de l'interaction'.

Exemples Par exemple, dans le cadre de sport collectif, une interaction possible serait pour un agent de demander à un autre agent de lui faire une passe. Cette interaction implique deux agents : l'agent demandeur et l'agent qui a la balle. La phase de résolution de l'interaction consiste pour les deux agents à décider ensemble si la passe est effective ou non. En fonction du résultat choisi, cette décision modifiera localement les états des agents impliqués par le transfert de la balle d'un agent à l'autre.

De la même manière, la distribution de ressources (décrite dans l'article [MHK+98]) peut être comprise comme une interaction consistant à répartir les ressources entre l'ensemble des agents :
– le déclenchement de l'interaction est systématique
– la résolution est une décision prise par l'ensemble des agents (consistant à répartir les ressources pour maximiser la récompense globale à partir des estimations des agents)
– à partir de communications locales (qui sont les échanges de valeurs)

La question à laquelle nous allons chercher à répondre désormais consiste à déterminer comment représenter de manière homogène action et interaction dans un Interac-DEC-POMDP.

5.2.4.2 Représentation d'une interaction

Principe Dans un MDP, l'action est définie comme un symbole et l'évolution du système est décrite par une matrice de transition consistant en une fonction de l'état courant vers un autre état (éventuellement stochastique).

Pour une interaction nous souhaitons faire de même. Le choix effectué lors d'une interaction fournit un résultat qui doit faire évoluer le système. Le système évoluera de manière similaire par une matrice de transition définie pour chaque résultat qui associe à un état du monde donné, l'état du monde obtenu après exécution du résultat d'interaction décidé collectivement. Afin de rendre compte de la localité des interactions et des agents impliqués, cette matrice de transition dépendra aussi des agents impliqués dans l'interaction.

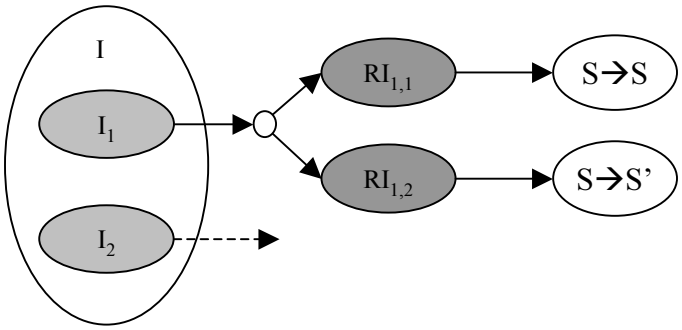


FIG. 5.2 – Structure des représentations des interactions

Structure des interactions Le module d'interaction est défini par des interactions structurées de la manière suivante (cf figure 5.2) :

- Les interactions possibles sont représentées par un ensemble de symboles d'interactions $I = \{I_k\}$. Une interaction I_k va être définie par les actions jointes que l'interaction propose et le nombre d'agents impliqués dans les interactions de ce type. Dans le cadre d'un jeu de sport collectif, l'interaction "demander une passe" sera représentée par un symbole $I_j \in I$. Les agents impliqués dans une interaction directe ont des rôles par rapport à cette interaction. Le rôle d'un agent est déterminé par sa position dans la liste triée des agents impliqués.
- Un ensemble de résultats possibles $RI_k = \{RI_{k,l}\}$ est associé à chaque type d'interaction I_k. Ces résultats correspondent à des étiquettes désignant des actions jointes. Les actions jointes sont définies de manière implicite dans le résultat et donc supposées connues. Par exemple, à l'interaction "demander une passe" (I_j) sont associés deux résultats possibles : le résultat $I_{j,0}$ "effectuer la passe" et le résultat $I_{j,1}$ "la passe est refusée". Le résultat "effectuer la passe" peut être compris comme le déclenchement d'une action jointe entre les deux agents impliqués : l'agent receveur enverra la balle et l'agent émetteur déclenchera une action pour la réceptionner.
- Pour chaque résultat $RI_{k,l}$, un ensemble de matrice de transition $TRI_{k,l} : agent^* \times S \rightarrow S$ [7] permet de déterminer l'état d'arrivée du système après application de l'interaction. L'état d'arrivée dépend de l'état de départ et des agents impliqués dans l'interaction. Ainsi au résultat "effectuer la passe" est associée la matrice de transition correspond aux transformations d'états consistant à faire passer le ballon du receveur à l'émetteur. Bien entendu, ces matrices modifient différemment l'état du système en fonction des agents émetteur et receveur.

Lorsqu'un agent déclenche une interaction, en raison de ses perceptions partielles, il ne peut pas savoir si l'agent destinataire est en mesure de répondre et si l'interaction peut être exécutée. La fonction *possible* est une fonction correspondant à la réponse du système à un déclenchement d'interaction. Cette fonction a pour objectif de rendre compte de la localité des communications et des échanges d'information. Il s'agit d'une fonction $possible : S \times I \times agents^* \rightarrow \{0, 1\}$ qui associe à tout état global, toute interaction et tout tuple d'agents destinataires un booléen. Ce booléen détermine si l'interaction directe avec les autres agents est envisageable dans la situation globale donnée. Si le booléen est faux, les agents ne peuvent pas entrer en contact, ils ne peuvent pas échanger d'information et l'interaction déclenchée échoue. Si le booléen retourné est vrai, les agents peuvent interagir, échanger de l'information et décider collectivement d'un résultat.

L'agent n'a pas accès directement à la fonction *possible* mais le système fournit une réponse à l'agent après que celui-ci ait déclenché l'interaction. Cette réponse peut ainsi être interprétée comme la réception par l'agent d'un signal de communication venant d'un autre agent (dans le cas d'une réponse positive) initiant la résolution de l'interaction ou comme le dépassement d'un délais d'attente de réception (dans le cas d'une réponse négative) conduisant l'agent à annuler l'interaction. L'objectif de la fonction *possible* est de s'affranchir de protocole de communication qui ne constituent pas le cœur de notre proposition.

Il reste maintenant à représenter la manière dont les agents utilisent ces interactions, les prises de décision qui sont impliquées et comment une interaction s'exécute dans le système.

[7] Le symbole $agent^*$ désigne l'ensemble des listes triées d'agents.

5.2.4.3 Représentation des politiques

Avant de pouvoir décrire l'exécution d'une interaction, il est nécessaire de représenter les fonctions de prise de décision des agents vis à vis des interactions. On distingue les politiques individuelles de déclenchement, des politiques collectives de résolution.

Politiques de déclenchement Chaque agent dispose d'une politique de déclenchement $\pi_{decl,i}$. Il s'agit d'une politique individuelle qui permet à un agent de choisir de manière autonome le type d'interaction qu'il souhaite entreprendre et les agents avec lesquels il souhaite interagir.

En toute généralité, cette fonction est dépendante de l'historique des perceptions de l'agent H, mais nous nous limiterons à des agents sans mémoire pour lesquels les décisions sont prises à partir de leurs perceptions instantanées Γ. Dans le cas d'une interaction à plusieurs participants, les places des agents dans le liste triée d'agents désignent les rôles des agents conformément à la définition de l'interaction. Cet ordre reste inchangé par la suite de l'interaction afin de conserver les affectations des rôles au sein de l'interaction : $\pi_{decl,i} : \Gamma_i \rightarrow I \times agent^*$.

L'avantage d'inclure les agents dans le résultat d'une décision de déclenchement d'interaction est de pouvoir permettre à l'agent déclencheur de choisir avec qui il souhaite interagir et de restructurer les interactions au fur et à mesure de l'exécution du système.

Politiques de résolution Les interactions sont résolues grâce à une politique de résolution Π_{agent^*,I_k} définie pour chaque liste d'agents et chaque type d'interaction I_k.

Une politique de résolution d'interaction associe à tout ensemble d'historiques des agents impliqués un résultat parmi les résultats possibles de l'interaction qui sera exécuté. Nous nous limiterons à des politiques sans mémoire. Pour une politique de résolution, un résultat est associé au tuple des perceptions instantanées des agents impliqués $\Pi_{agent^*,I_k} : \Gamma^*_{agent} \rightarrow RI_k$.

L'intérêt de l'interaction directe que nous envisageons repose sur ces politiques collectives de résolution. Ce sont elles qui font que le résultat d'une interaction déclenchée par un agent dépend aussi des autres agents impliqués, de leurs connaissances et de leurs motivations vis à vis de la tâche à résoudre. Ces considérations seront développées dans une section suivante (cf partie 5.3.2)

Un problème qui restera à résoudre par la suite et que nous aborderons dans le chapitre suivant 6 sera la manière de représenter toutes ces politiques de manière distribuée dans le système à partir de communications locales autorisées au cours de l'interaction. Nous verrons comment il peut être possible de construire ces politiques d'interactions en fonction des comportements des agents et des mécanismes d'adaptation individuels.

5.2.4.4 Schéma d'exécution d'une interaction

L'exécution d'une interaction se fait en plusieurs étapes (cf figure 5.3) :

une étape de déclenchement Dans un premier temps, un agent décide de déclencher une interaction vers un ensemble d'agents particuliers en fonction de sa politique individuelle de déclenchement $\pi_{decl,i} : \Gamma_i \rightarrow I \times agent^*$ (cf fig 5.3 a). A l'issue de l'étape de déclenchement,

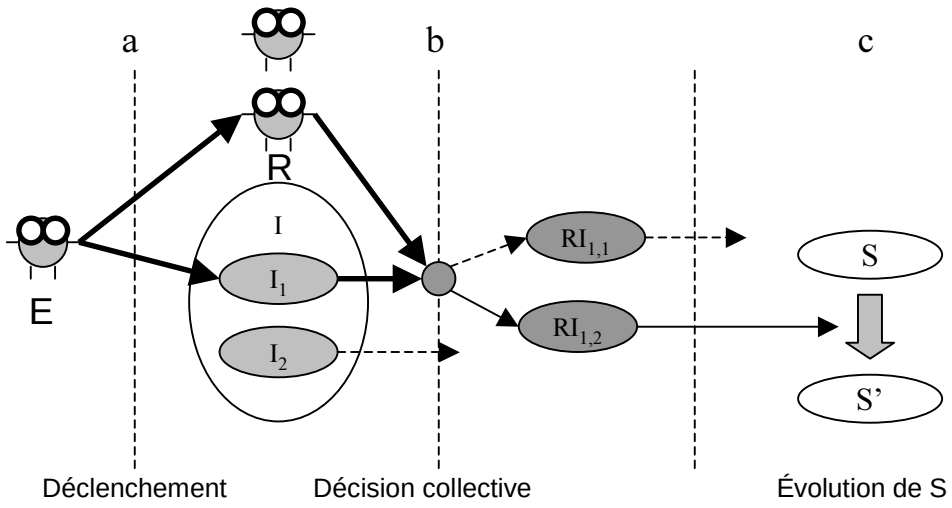


FIG. 5.3 – Exécution d'une interaction en trois phases : a) déclenchement, b) résolution, c) exécution du résultat

la fonction *possible* est sollicitée pour savoir si les échanges et la résolution de l'interaction peuvent avoir lieu. Si ce n'est pas le cas, l'exécution de l'interaction s'arrête.

Une étape de résolution des interactions Si l'interaction est possible, un résultat est décidé collectivement à partir de la politique jointe définie sur l'ensemble des agents impliqués : $\Pi_{agent^*} : \Gamma_{agent^*} \rightarrow RI_k$. Cette politique fournit en fonction des observations des agents impliqués le résultat décidé.

Une transition Enfin, le résultat d'interaction choisi correspond à l'exécution d'une action jointe coordonnée. Cette dernière conduit à une évolution de l'état courant du système en fonction de la matrice de transition associée au résultat d'interaction choisi (cf Figure 5.3.c) $s \leftarrow TRI_{RI_{i,l}}(S, agent^*)$

Algorithme d'exécution L'algorithme 6 décrit l'exécution d'une interaction à l'initiative de l'agent i et n'impliquant qu'un seul autre agent.

Algorithme 6 Executer une interaction

Observation de l'agent i : $o_i \leftarrow O(s)_i$
Choix de l'interaction : $(I_k, agent_j) \leftarrow \pi_{decl,i}(o_i)$
si $possible(I_k, agent_j)$ **alors**
  Observation de l'agent j : $o_j \leftarrow O(s)_j$
  Échanges locaux et choix du résultat de l'interaction : $(RI_{k,l} \leftarrow \Pi_{i,j}(o_i, o_j))$
  Exécution du résultat : $s \leftarrow TRI_{k,l}(S, i, j)$
  Fin de l'interaction
fin si

5.2.5 Interac-DEC-POMDP - Module d'interaction

5.2.5.1 Objectif

Maintenant que l'interaction directe a été formalisée, l'objectif du module d'interaction est d'agencer l'exécution des interactions dans le système.

Ce module se charge de solliciter séquentiellement les agents comme agent émetteur d'interaction. L'agent choisit une interaction directe parmi celles qui lui sont offertes. Une fois l'interaction décidée, elle est exécutée et c'est à l'agent suivant d'exécuter une autre interaction. Ainsi, au début de chaque interaction, chaque agent à la possibilité d'effectuer une observation de son environnement pour pouvoir déclencher l'interaction en conformité avec l'état du système et pour que le choix du résultat d'interaction soit lui aussi conforme à la politique de résolution.

L'ordre de sollicitation des agents est déterminé aléatoirement pour éviter d'introduire des biais au sein du système. Ces biais peuvent se produire lorsque deux interactions exécutées successivement impliquent des agents identiques. Par exemple, si A veut échanger un objet avec B et que C souhaite échanger un objet avec B, la manière dont va s'organiser les interactions va fortement influencer l'état global final. Si A interagit avec B avant C, B va recevoir l'objet initialement possédé par C. Si C interagit avec B avant A, B recevra l'objet initialement possédé par A.

5.2.5.2 Algorithme

L'algorithme 7 présente l'algorithme d'exécution du module d'interaction en faisant référence à l'algorithme 6 d'éxecution d'une interaction.

Algorithme 7 Executer le module d'interaction

Liste aléatoire des agents : $liste \leftarrow ordre_{aleatoire}$
pour tout agent $\in liste$ **faire**
  Execute interaction (agent)
fin pour

5.2.6 Interac-DEC-POMDP - vue générale

Jusqu'à présent, nous avons présenté les différentes facettes d'un Interac-DEC-POMDP. Cette partie constitue une synthèse de ces éléments et présente de manière globale le formalisme Interac-DEC-POMDP.

5.2.6.1 Formalisation

Un Interac-DEC-POMDP est donc défini par un tuple $< S, A_i, T, R, \Gamma_i, O, I, RI, TRI >$
- S désigne l'état global du système.
- A_i désigne les actions possibles pour l'agent i, A désigne l'ensemble des actions jointes.
- T désigne la matrice de transition du système : $T : S \times A \times S \rightarrow [0, 1]$
- R désigne la fonction de récompense globale $R : S \times A \rightarrow \mathbb{R}$
- Γ_i désigne les observations possibles pour l'agent i, Γ désigne l'ensemble des tuples d'observation jointe des agents.
- O désigne la fonction d'observation $O : S \times A \times \Gamma \rightarrow [0, 1]$

- I désigne l'ensemble des interactions possibles dans le système
- RI_k désigne les résultats possibles de l'interaction k
- $TRI_{k,j}$ désigne une matrice de transition modélisant l'évolution du système suite à l'exécution du résultat $RI_{k,j}$ en fonction de l'état global du système et des agents impliqués.

5.2.6.2 Exécution

L'exécution globale du modèle se fait selon des cycles en deux phases. La première phase concerne le module d'action, la seconde le module d'interaction.

Module d'action Chaque agent i choisit individuellement son action a_i en fonction de sa politique π_i. Les actions sont émises simultanément et le système évolue conformément à la matrice de transition et à l'action jointe comme pour un DEC-POMDP .

Module d'interaction Séquentiellement et dans un ordre aléatoire, chaque agent déclenche une interaction (potentiellement aucune) en fonction de sa politique individuelle $\pi_{decl,i}$. En déclenchant une interaction, il désigne le type d'interaction ainsi que les agents avec lesquels il souhaite interagir. L'interaction est résolue grâce aux politiques jointes de résolution d'interaction Π, le résultat choisi est exécuté et un autre agent est sollicité pour le déclenchement d'une interaction.

5.2.6.3 Algorithme d'exécution

Avec les éléments que nous avons proposés, l'algorithme 8 regroupe les algorithmes précédents. Il présente l'algorithme d'exécution d'un Interac-DEC-POMDP en se limitant à des interactions impliquant deux agents [8].

5.3 Discussion sur le formalisme

5.3.1 Caractéristiques du formalisme pour les systèmes multi-agents

5.3.1.1 Éléments du formalisme

Le cadre formel Interac-DEC-POMDP présente un ensemble d'éléments pour décrire un problème multi-agents.

Tout d'abord, le formalisme permet ainsi de décrire les architectures externes des agents. Un agent est doté de perceptions partielles et les possibilités offertes aux agents sont constituées d'actions simples, et d'interactions qui peuvent avoir plusieurs résultats possibles.

Ensuite, ce formalisme décrit les réactions du système à l'utilisation de ces possibilités. Ces réactions sont de deux ordres :

- les réactions du système aux actions des agents représentées par une matrice de transition sur les actions jointes
- les réactions du systèmes par rapport aux interactions représentées par des matrices de transitions dépendantes des agents impliqués dans l'interaction.

[8] lorsque plus d'agents sont impliqués, il est nécessaire de remplacer $agent_j$ par une liste triée d'agents et de modifier les équations qui en découlent.

Algorithme 8 Execution d'un Interac-DEC-POMDP pendant t_{fin} pas de temps

$t \leftarrow 0$
répéter
  Module d'action
  Observation : $O = (o_0, ...o_n) \leftarrow O(s)$
  pour tout agents $i \in [0..n]$ **faire**
    choix de l'action i : $a_i \leftarrow \pi_i(o_i)$
  fin pour
  action jointe : $a \leftarrow (a_0...a_n)$
  exécution de l'action jointe : $s' \leftarrow T(s, a)$
  récompense : $R \leftarrow R(s, a, s')$
  modification de l'état du système : $s \leftarrow s'$
  Module d'interaction
  Liste aléatoire des agents : $liste \leftarrow ordre_{aleatoire}$
  pour tout agent $\in liste$ **faire**
    Observation de l'agent i : $o_i \leftarrow O(s)_i$
    Choix de l'interaction : $(I_k, agent_j) \leftarrow \pi_{decl,i}(o_i))$
    si $possible(I_k, agent_j)$ **alors**
      Début de l'exécution de l'interaction
      Observation de l'agent j : $o_j \leftarrow O(s)_j$
      Échanges locaux et choix du résultat de l'interaction : $(RI_{k,l} \leftarrow \Pi_{i,j}(o_i, o_j)$
      Exécution du résultat : $s \leftarrow TRI_{k,l}(S, i, j)$
      Fin de l'exécution de l'interaction
    fin si
  fin pour
  $t \leftarrow t + 1$
jusqu'à $t > t_{fin}$

Il permet de représenter les comportements des agents concernant l'utilisation de ces possibilités par des fonctions politiques. Le formalisme ne stipule rien quant aux architectures internes possibles des agents (même si nous nous sommes focalisés sur des politiques d'agents sans mémoire à court terme).

Enfin, le schéma d'exécution du modèle formalise l'agencement des comportements et des transitions effectuées dans le système.

Le formalisme proposé a été conçu pour représenter des systèmes multi-agents et est bien adapté car :

- Il est fondé sur des initiatives individuelles que ce soit au niveau des actions ou au niveau des interactions puisque celles ci sont déclenchées à l'initiative d'un individu. Cette individualité permet à chaque agent de disposer d'une autonomie (au sens de Jennings).
- Il se fonde sur des interactions locales en ce sens qu'elles impliquent un nombre réduit d'agents
- Il se fonde sur des communications limitées

5.3.1.2 Contraintes de localité

Les principes de localités souhaités peuvent être représentés dans ce formalisme :
- En toute généralité, l'observabilité des agents est partielle
- Les déclenchements d'interactions sont locaux.
- Le modèle d'interaction permet de représenter tous les types d'interactions, mais le principe que nous suivrons par la suite sera d'utiliser des interactions et des communications locales. c'est à dire que lorsque les agents sont trop éloignés au sens d'une topologie sur l'espace d'état à définir, les interactions déclenchées n'auront aucun résultat sur le système. La fonction *possible* permet de rendre compte des localités des interactions.
- les agents perçoivent localement l'avancement de la tâche : celle ci est définie par une récompense globale décomposée en récompenses locales additives. Chaque agent ne perçoit qu'une récompense locale liée à ses perceptions. Ainsi, il est effectivement possible de guider les agents tout en restant dans un cadre réaliste pour lequel aucune représentation globale n'est possible.

5.3.2 Intérêts de l'interaction directe

5.3.2.1 Résolution collective

Tout d'abord, l'interaction directe permet de représenter explicitement la présence d'autres agents dans le système et permet aux agents de raisonner sur la multiplicité des prises de décision. Dans ces conditions, les agents peuvent intégrer dans leur décision les caractéristiques comportementales des autres agents puisque ceux-ci participent aussi aux décisions collectives.

Par exemple, l'interaction demander une passe à un agent A est différente de l'action consistant à prendre le ballon à l'agent A. Dans le cas d'une action, l'agent à l'origine de l'action parvient à prendre la ballon en fonction de la matrice de transition T représentant les lois de l'environnement. Dans le cas d'une interaction, la politique de résolution d'interaction permet à l'agent A d'avoir une résistance vis-a-vis de l'interaction en refusant éventuellement de faire la passe. Cette résistance peut venir des :
- des perceptions de l'agent A puisque la politique de résolution dépend des perceptions de tous les agents impliqués dans l'interaction. Par exemple, si l'agent A perçoit qu'il n'a aucun obstacle, il peut choisir de conserver le ballon
- des comportements de l'agent A puisque résoudre une interaction avec i est différent de résoudre la même interaction avec l'agent j. Par exemple, l'agent A peut savoir qu'il a énormément intérêt à conserver le ballon au vu de sa politique future et de ses espérances de gain.

Ainsi, pour des politiques d'interactions données, les interactions consistent à effectuer des modifications de l'état global du système fondées non pas uniquement sur les lois d'évolution du système mais aussi sur les caractéristiques comportementales des agents avec lesquels on interagit.

5.3.2.2 Actions jointes déterminées a priori

La notion d'interaction est une des composantes centrales de l'élaboration des systèmes entre agents que l'on retrouve fréquemment :
- au niveau des méthodologies de conception
- au niveau des descriptions de systèmes multi-agents

– parmi les entités que le concepteur souhaite implanter
mais elle n'apparaît pas explicitement dans les DEC-POMDPs.

En considérant des actions jointes, l'interaction directe permet de résoudre un certain nombre de problèmes de coordination auxquels le concepteur a déjà pensé en définissant les interactions qu'il souhaite mettre en œuvre.

5.3.2.3 Nouvelles entités de calcul

Elle définit de nouvelles entités constituées par les agents interagissant. Il est alors possible d'effectuer des calculs sur cette entité comme des transferts de récompense entre agents pour répondre à la tragédie des communs et construire des systèmes à partir de récompenses individuelles locales. Ceci permet de définir de nouveaux moyens de construire des systèmes coopératifs.

5.3.2.4 Structuration implicite du système

Enfin, elle définit une structure au problème à partir d'un concept facilement interprétable fondamental dans les systèmes multi-agents. En explicitant la notion d'interaction directe, on fait émerger une structure au problème et à la dynamique du système. Une certain nombre d'approches montrent qu'il est alors possible de tirer parti de cette structure pour proposer des solutions approchées comme celle de Guestrin [GKP01].

5.3.3 Problème associé à un Interac-DEC-POMDP

Les Interac-DEC-POMDP permettent ainsi de représenter des actions individuelles et des interactions directes dans un même formalisme. Ils permettent de définir une nouvelle classe de problèmes dans laquelle les agents peuvent interagir directement.

Résoudre une instanciation, consiste à trouver pour un tuple $< S, A_i, T, R, \Gamma_i, O, I, RI, TRI >$ l'ensemble des politiques individuelles π_i pour tout i, l'ensemble des politiques individuelles de déclenchement $\pi_{trig,i}$ pour tout i, l'ensemble des politiques de résolution d'interaction Π_{i,j,I_k} pour tout i, j, k permettant de maximiser la moyenne des sommes des récompenses globales reçue par le système à l'exécution à partir d'un état s_0.

En d'autres termes, résoudre un Interac-DEC-POMDP consiste à déterminer :

Quoi faire par les politiques d'actions individuelles π_i

Quand, comment et avec qui interagir par les politiques de déclenchement individuelles $\pi_{decl,i}$

Comment résoudre l'interaction par les politiques jointes de résolution Π pour toutes les interactions et tous les ensembles d'agents

Ce problème possède un nombre fini de solutions puisque les politiques sont dénombrables. Il existe donc bien une solution dans l'espace des politiques individuelles, politiques de déclenchement et les politiques jointes de résolution.

5.3.4 Positionnement du formalisme Interac-DEC-POMDP

L'objectif de cette partie est de positionner le formalisme Interac-DEC-POMDP par rapport au formalisme DEC-POMDP sur lequel il est construit.

5.3.4.1 Inclusion des DEC-POMDPs

Tout DEC-POMDP peut être représenté par un Interac-DEC-POMDP sans interaction. Les DEC-POMDPs sont donc inclus dans les Interac-DEC-POMDP.

Ainsi la complexité de résolution d'un Interac-DEC-POMDP de manière centralisée est en toute généralité plus complexe que celle d'un DEC-POMDP.

Comme résoudre un DEC-POMDP est NEXP complet et est irréalisable pour le moment, nous nous limiterons à la recherche de solutions approchées. Comme nous le montrons par la suite, l'interaction peut permettre de construire des solutions approchées à moindre coût.

5.3.4.2 Différence fondamentale

La différence fondamentale entre un DEC-POMDP et un Interac-DEC-POMDP réside dans la présence d'interactions et les politiques qui y sont liées qui utilisent l'ensemble des perceptions locales des agents impliqués et pour lesquelles les politiques de résolution dépendent des agents impliqués : par exemple Π_{A_i,A_j} est différente de Π_{A_i,A_k}. Ceci permet d'intégrer de manière explicite l'autre (et donc son comportement dans la décision).

5.3.4.3 Résolution centralisée

Il est possible d'envisager de résoudre un Interac-DEC-POMDP de manière entièrement centralisée. Ce problème est en général très complexe mais est accessible dans certaines conditions : lorsque les agents perçoivent l'état global de l'environnement, il est possible de se ramener à un MDP. Il s'agit d'un cas similaire au MMDP.

Dés lors, le MDP équivalent est constitué de cycles permettant de reconstituer l'exécution d'un pas de temps d'un Interac-DEC-POMDP :

- le premier correspond à l'exécution du modèle d'action et consiste à raisonner sur l'action jointe émise par les agents.
- les autres cycles sont plus complexes et consiste à raisonner sur l'émission des interactions :
  - raisonner d'abord sur l'étape de déclenchement
  - raisonner ensuite sur l'étape de résolution qui dépend

Cette résolution se heurte directement aux principes de localité mis en avant mais permet d'obtenir des politiques optimales pour pouvoir évaluer notre approche. Cette approche de résolution doit en outre se limiter à des agents en faible nombre pour éviter l'explosion combinatoire du nombre d'état et du nombre d'actions jointes à considérer.

5.4 Bilan du chapitre

En conclusion, nous avons présenté un formalisme original intitulé Interac-DEC-POMDP. Ce formalisme permet de représenter des systèmes multi-agents permettant d'avoir des agents s'influençant mutuellement

- par des interactions directes
- et par des interactions indirectes

En introduisant des interactions directes représentées au niveau individuel, nous introduisons en outre des couplages explicites entre les agents pour

- structurer le système
- permettre des prises de décisions collectives en raisonnant sur la multiplicité des acteurs dans le système (par les fonction Π)
- tout en respectant les contraintes de localités des SMAs.

Une des caractéristiques de ce formalisme réside dans la capacité des agents à restructurer les interactions et les relations qu'ils peuvent entretenir avec les autres agents au cours de l'exécution du système.

Ce formalisme peut être compris de deux manières :

- Il peut être compris comme un modèle permettant d'exécuter les comportements d'agents capables d'agir et d'interagir sans rien spécifier concernant leurs architectures internes. Ainsi, par exemple, le modèle Hamelin peut être représenté et exécuté dans ce formalisme.
- Il peut être compris comme la formalisation d'un nouveau problème d'optimisation consistant à construire les politiques d'agents capables d'agir et d'interagir.

Dans la suite, nous allons chercher à résoudre le problème d'optimisation correspondant au problème de construction des comportements des agents. Afin de voir dans quelle mesure il était possible de tirer parti de ces propriétés, nous nous sommes concentrés sur une sous-classe des Interac-DEC-POMDP qui permet d'isoler la problématique de conception des interactions.

Chapitre 6

Mise en œuvre

Dans le chapitre précédent, nous avons présenté le formalisme interac-DEC-POMDP. Ce formalisme permet de représenter des systèmes constitués de plusieurs agents en interaction capables d'émettre des actions et d'utiliser des interactions directes.

Nous n'avons cependant pas encore répondu à la question que nous nous étions posés au début du manuscrit : à savoir s'il est possible de construire des comportements collectifs à partir de processus de construction de comportements individuels. Maintenant que nous avons formalisé la notion d'interaction directe qui nous semble l'élément clef permettant d'appréhender le passage de mécanismes de construction et d'adaptation de comportements individuels à des mécanismes de construction de comportements collectifs, nous allons chercher à construire les comportements des agents.

Nous aborderons le problème de construction des comportements à l'aide d'un exemple correspondant à une sous-classe du formalisme Interac-DEC-POMDP pour laquelle l'interaction directe entre agents est centrale.

Nous présenterons tout d'abord une description de cette sous-classe et nous décrirons le problème particulier sur lequel nous nous sommes focalisés. Nous présenterons ensuite les principes de l'algorithme puis l'algorithme lui-même. Enfin, nous présenterons et discuterons les résultats que nous avons pu obtenir par cette approche.

6.1 Résolution d'un Interac-DEC-POMDP

6.1.1 Une sous-classe de problèmes

Cette partie a pour objectif de présenter la sous-classe d'Interac-DEC-POMDP sur laquelle nous allons nous concentrer. Cette sous-classe est caractérisée par le fait que les interactions directes constituent les seules influences possibles entre agents. Elle permettra donc de mettre en évidence la pertinence de l'introduction de ce concept dans les cadres formels markoviens.

6.1.1.1 Description générale

Dans cette sous-classe, les actions entre agents sont indépendantes et le module d'action peut se décomposer en des modules d'action individuels :

- L'espace d'état peut se partitionner en espaces d'états individuels S_i. Chaque agent i est caractérisé par $s_i \in S_i$ et l'état global du système est défini par $S = \times S_i$.
- La matrice de transition T peut se décomposer en matrices de transition individuelles T_i définies sur les espaces d'état individuels $T_i : S_i \times A_i \times S_i \rightarrow [0, 1]$
- Chaque agent perçoit intégralement son état individuel s_i mais ne perçoit pas l'état global du système. Ainsi un agent ne connaît pas l'état et la position des autres agents ni la configuration globale du système.
- Chaque agent peut évaluer localement l'avancement de la tâche globale. La fonction de récompense globale R est supposée pouvoir se décomposer en fonctions locales additives r_i, telles que $R = \sum r_i$. Chaque r_i est perçue localement par l'agent i et s'exprime dans son espace d'état individuel : $r_i : S_i \times A_i \rightarrow \mathbb{R}$

Comme on souhaite des interactions directes aux conséquences locales, le module d'interaction revêt lui aussi une forme particulière dans ce cadre. Les matrices de transition associées aux résultats d'une interaction directe peuvent s'exprimer en fonction des espaces individuels des agents impliqués. Par exemple, la signature d'une matrice de transition associée au résultat d'interaction $R_{k,l}$ impliquant deux agents i et j peut s'écrire sous la forme suivante : $TRI_{k,l,i,j} : S_i \times S_j \times S_i \times S_j \rightarrow [0, 1]$.

Les interactions directes constituent le seul moyen qu'a un agent i de modifier l'espace individuel de l'agent j. De plus, du fait des contraintes de localité que nous souhaitons, nous supposerons que les 'états d'interface' à partir desquels un agent peut tenter d'exercer une interaction sur un autre agent sont en faible nombre. Ces dernières contraintes s'expriment par la fonction $possible : S \times I \times agent^* \rightarrow \{0, 1\}$ qui évalue, à partir de l'état global du système et d'une liste d'agents, la possibilité d'exécuter l'interaction I_k lorsque celle-ci est déclenchée.

Nous nous sommes intéressés à cette sous-classe de problèmes pour plusieurs raisons :
- Elle est proche de problèmes concrets comme le partage de tâches sur un réseau, pour lequel chaque machine traite des données et peut effectuer des transferts de données entre elles ou comme les problèmes de chaîne de production.
- Cette sous-classe correspond à des problèmes collectifs : même si les seuls couplages possibles entre agents résident dans les interactions directes, l'ensemble des comportements des agents sont couplés (cela sera décrit plus précisément ultérieurement) et un agent doit pouvoir intégrer la présence d'autres agents pour pouvoir choisir l'action optimale.
- Cette sous-classe correspond à des problèmes pour lesquels l'interaction directe est centrale. Elle pose le problème de la construction de ces interactions. Résoudre ce type de problème constitue une première étape pour intégrer l'interaction dans des situations plus complexes.
- Cette sous-classe permettra de mettre en avant la valeur ajoutée de l'interaction.

6.1.1.2 Problème des pompiers

Agents et espace d'état Le problème particulier sur lequel nous nous focalisons consiste à éteindre des feux répartis dans un environnement. Chaque agent-pompier i évolue dans une pièce caractérisée par des états en nombre fini. Un problème est constitué par un assemblage de ces pièces dont certaines peuvent contenir de l'eau ou du feu. Ainsi, seuls quelques agents peuvent atteindre les feux à éteindre et seuls quelques agents peuvent accéder à des puits[9]. Les agents

[9] permettant de remplir un seau.

ont en outre des seaux à leur disposition afin de transporter l'eau d'un endroit à un autre.

Les agents pompiers sont répartis en plusieurs catégories :
- les **agents pompiers ravitailleurs** qui ont accès à de l'eau et ont la possibilité de remplir des seaux.
- les **agents pompiers extincteurs** qui ont accès au feu et peuvent l'éteindre à condition qu'ils disposent d'un seau rempli
- les **agents pompiers couloirs** qui sont situés dans des pièces sans feu ni eau. Chacune de ces pièces dispose de quatre portes menant à des pièces connexes.

Nous utiliserons ces termes par la suite pour désigner les différents types d'agents.

L'état individuel d'un agent correspond au couple constitué par sa position dans la pièce et la variable booléenne 'état de son seau' (rempli ou vide). La position d'un agent est très importante car c'est elle qui qui va conditionner la possibilité d'exécuter certaines interactions.

Actions Au cours de l'exécution du module d'action, un agent pompier peut émettre au choix l'une des actions suivantes :
- Il peut se déplacer au sein de la pièce dans laquelle il se trouve.
- S'il est situé sur une case connexe à une case d'eau, il peut remplir son seau[10].
- S'il est situé sur une case connexe à une case de feu et que son seau est rempli, il peut verser le contenu du seau sur le feu[11]. Lorsque cette dernière action a lieu, l'agent qui en est à l'origine reçoit une récompense individuelle positive (+100). Cette récompense n'est perçue que par cet agent qui est alors le seul capable d'évaluer l'avancement de la tâche.

Dans tous les autres cas, aucune récompense n'est distribuée aux agents.

Interactions directes Enfin, les agents peuvent interagir entre eux. Les seules interactions possibles entre agents sont des interactions directes qui correspondent à des échanges de seaux entre deux agents.

Pour qu'une interaction directe puisse avoir lieu, les deux agents doivent se trouver sur des cases connexes dans leurs pièces respectives. Une interaction directe peut avoir deux résultats :
- soit l'échange est effectif, les contenus des seaux sont échangés et les états individuels des agents sont modifiés en conséquence (modifications des booléens 'état du seau' correspondants)
- soit l'échange est refusé et les états individuels des agents restent inchangés.

Si les agents ne se trouvent pas sur des cases connexes, aucun échange et aucune communication ne sont possibles. La fonction *possible* décrite dans le chapitre précédent rend compte de cette contrainte du système.

Ainsi, cette interaction ne peut avoir lieu que localement et respecte bien nos contraintes de localité tant au niveau de ses conditions d'utilisation que de ses conséquences. L'interaction directe n'est liée à aucune récompense. Son utilisation ne permet pas de résoudre directement

[10] Seuls les agents pompiers ravitailleurs ont cette possibilité.
[11] Seuls les agents pompiers extincteurs ont cette possibilité.

la tâche mais peut néanmoins conduire d'autres agents à le faire (en mettant un seau à leur disposition par exemple).

Enfin, afin de limiter le nombre d'interactions au cours de l'exécution et pour valider notre apprentissage des politiques de déclenchement, le déclenchement de l'interaction directe d'échange sera mis en concurrence avec le déclenchement d'aucune interaction.

Problème posé Le problème posé consiste pour une configuration d'agents pompiers donnée, à construire de manière décentralisée les politiques d'action, d'interaction et de déclenchement des agents pour que la somme des récompenses globales R reçues par le système à l'exécution soit maximale. Il s'agit donc effectivement d'un problème coopératif défini au niveau global même si les agents sont guidés par des récompenses perçues localement.

De manière plus générale, nous cherchons à construire de manière entièrement décentralisée le comportement collectif du système. Ce comportement pourra s'exprimer par l'apparition d'organisations correspondant à des chaînes constituées d'agents transportant des seaux pour éteindre les feux.

6.1.1.3 Difficultés associées au problème

Même si les seules influences entre les agents sont locales et représentées explicitement dans le système, la résolution reste non triviale.

Récompenses individuelles Tout d'abord, les agents sont guidés par des récompenses individuelles. Ces récompenses individuelles correspondent à l'avancement de la tâche en cours mais sont perçues localement par les agents. Ainsi, dans certaines circonstances, un agent peut participer à l'avancement de la tâche sans recevoir directement une récompense immédiate. Il ne peut alors pas évaluer correctement son comportement. Cela est le cas, par exemple, lorsqu'un agent couloir fournit un seau à un agent extincteur.

Interdépendance des politiques De plus, le système est caractérisé par une interdépendance entre les politiques d'actions et d'interactions des agents.

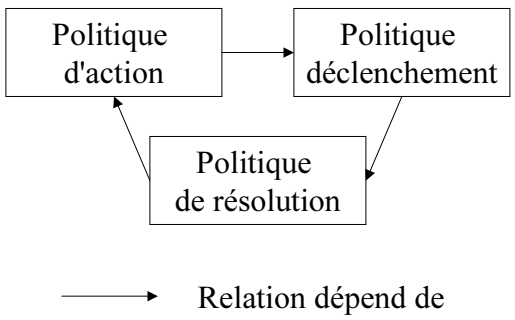


FIG. 6.1 – Inter-dépendance des politiques

Cette inter-dépendance entre les politiques des agents s'exprime selon le schéma 6.1 :
- Les politiques d'action des agents font évoluer leurs positions et rendent possible l'exécution de certaines interactions.
- Les politiques de déclenchement décident des interactions à exécuter.
- Les politiques de résolution d'interaction modifient les états individuels des agents en interaction. Ces nouveaux états individuels amèneront les agents à émettre d'autres actions par la suite.

Ainsi, pour construire les politiques de résolution d'interaction directe, il faut tenir compte des politiques d'action qui elles même doivent tenir compte des politiques de déclenchement et des politiques de résolution d'interaction. Les politiques de résolution d'interaction impliquent en outre plusieurs agents et introduisent des couplages au sein de la communauté.

Dans le cas du problème des pompiers, ce couplage entre les différentes politiques des différents agents s'exprime par exemple de la manière suivante : lorsqu'un agent dispose d'un seau, pour savoir quelle direction il doit emprunter, il est nécessaire de savoir quel agent est intéressé par l'obtention d'un seau plein (ce qui dépend de ses actions futures) afin de pouvoir le lui transmettre. De plus, il faut en outre être sûr que l'agent intéressé sache que l'agent considéré ait l'intention de lui amener un seau pour qu'il puisse être en situation de déclenchement d'interaction.

Localité des perceptions A ces problèmes s'ajoute le fait que les agents n'ont qu'une perception partielle de leur environnement.

Considérons un agent couloir. Cet agent est situé dans une pièce constituée de plusieurs cases dont certaines permettent d'interagir avec les agents situés dans d'autres pièces connexes. L'agent n'a pas d'information sur les pièces voisines. Il ne sait pas, par exemple, si une pièce connexe contient des feux ou des puits, ni même si elle est occupée par un agent. Il n'a pas non plus accès à la position des autres agents, à leurs états, à leurs intentions ou leurs comportements. Enfin, il ne connaît pas la structure globale du système et n'a pas d'information concernant la localisation des feux et des puits.

La figure 6.2 présente la situation telle qu'elle peut être perçue par un agent pompier et différentes configurations globales pouvant correspondre à cette perception.

Chaque pièce de la figure 6.2 contient un agent (même s'il n'est pas représenté). Les pièces ne contenant pas d'agents ne sont pas représentées mais les agents couloirs ne savent pas de quelles pièces il peut s'agir. Le type de l'agent dépend du symbole représentant la pièce (cf légende de la figure 6.2).

La figure 6.2 présente ainsi quelques topologies de systèmes possibles à partir des perceptions identiques d'un agent couloir. Pour ces différentes topologies indiscernables par les perceptions partielles de l'agent couloir qui nous concerne, les politiques optimales sont différentes :
- Dans la première situation, l'agent n'a aucun moyen d'accéder à un seau plein puisqu'il n'existe aucun agent ravitailleur dans le système. Quelles que soient les politiques des agents, il est impossible d'éteindre le feu et de générer des récompenses positives.

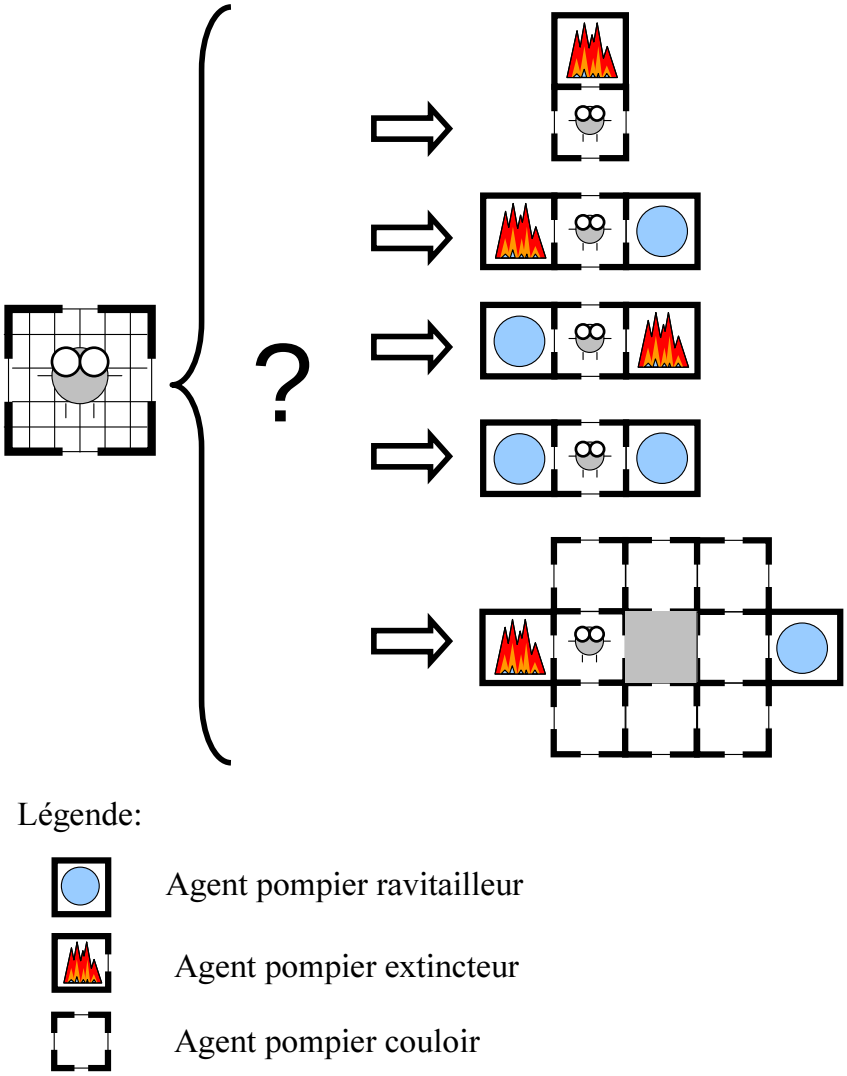


FIG. 6.2 – Quelques topologies possibles à partir des perceptions d'un agent couloir

- Dans la seconde situation, l'agent doit demander un seau plein à l'agent situé à sa droite et donner le seau à l'agent situé à sa gauche.
- Dans la troisième situation, l'agent doit demander un seau plein seau à l'agent situé à sa gauche et donner le seau à l'agent situé à sa droite.
- Dans la quatrième situation, puisqu'il n'y a aucun feu à éteindre, l'agent n'a aucun moyen de générer des récompenses dans le système.
- Dans la dernière situation, l'agent ne peut savoir a priori dans quelle direction se procurer un seau plein puisque cela dépend des comportements des autres agents couloir du système.

Comme, à partir de ses perceptions locales, l'agent n'a pas moyen de distinguer les différentes configurations globales possibles du système et n'a aucun moyen d'accéder directement à une récompense, il est nécessaire d'introduire de nouveaux mécanismes pour guider l'agent et lui permettre de tirer parti de mécanismes d'adaptation individuels. L'approche que nous proposons

consiste à utiliser les communications locales entre agents lors d'une interaction directe. Au cours de ces communications, les agents vont pouvoir échanger de l'information et adapter leurs comportements en fonction des agents avec lesquels ils interagissent.

6.1.2 Principe de l'approche de résolution

6.1.2.1 Objectif

Nous allons dans cette partie construire à partir de processus entièrement décentralisés et de récompenses individuelles un comportement collectif capable de générer la somme de récompense globale la plus importante possible. Il est à noter que du fait de l'architecture interne (pas de mémoire à court terme) et de l'architecture externe des agents (perception partielle de l'état global, des autres agents et des récompenses) le comportement optimal est en toute généralité inatteignable.

La question de la construction des comportements des agents s'instanciera de la manière suivante :
- comment construire les politiques d'interactions à partir des politiques individuelles
- comment construire les politiques d'actions en tenant compte des politiques d'interactions

6.1.2.2 Principe général

L'approche que nous proposons repose sur plusieurs propositions jointes (cf [TBC06]) :
- la première s'inspire des travaux de Parr et stipule qu'il est possible de déterminer séparément les comportements individuels de leur agencement collectif
- la seconde consiste à synchroniser les comportements des agents à partir de leur mémoire à long terme et de techniques d'apprentissage par renforcement [WD92]
- la dernière réside dans des heuristiques qui résolvent les interactions en fonction des Q-valeurs individuelles et qui permettent de répartir les récompenses dans le système au cours des interactions.

Réduction des comportements individuels Le problème des pompiers est proche des MDPs faiblement couplés pour lesquels un problème mono-agent peut se décomposer en sous-problèmes reliés par un faible nombre d'états (cf [Par98] présenté dans la partie 3.2.6). La principale différence réside dans le fait que dans les MDPs faiblement couplés les sous-problèmes sont organisés de manière séquentielle, alors que dans le problème des pompiers, ils sont organisés en parallèle, chaque agent évoluant dans son sous-espace **simultanément** aux autres agents.

Néanmoins, la première partie des travaux de [Par98] peut être utilisée. Elle consiste à extraire pour chaque sous-problème un cache de politiques constitué par l'ensemble des politiques potentiellement optimales reliant les états d'interface aux autres sous-problèmes. Pour le problème des pompiers, cela consiste pour chaque agent à construire les politiques potentiellement optimales pouvant mener aux états d'interface permettant d'interagir avec les autres agents.

Il est alors alors possible de simplifier le problème en ne considérant que ces politiques et les états d'interface au niveau de chaque agent. Par exemple, pour un agent couloir, il est inutile de considérer l'ensemble des états et des déplacements possibles au sein du couloir. En particulier, beaucoup de comportements consistent à osciller entre plusieurs cases d'une pièce et ne sont

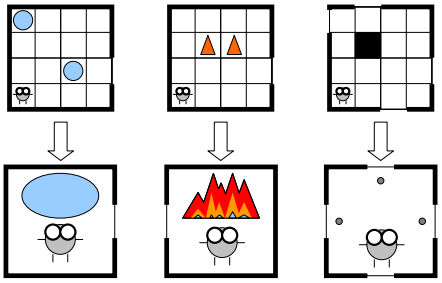


FIG. 6.3 – Réduction des différents sous-problèmes

d'aucune utilité pour la résolution. Les seules politiques utiles pour le système correspondent aux comportements conduisant à un état permettant d'interagir avec un autre agent, et ce indépendamment de la taille du couloir. Ainsi, bien que nous nous intéressons par la suite uniquement à quelques états et quelques actions par agent, il faut garder à l'esprit que la portée est plus générale.

Nous supposerons ainsi qu'il est possible d'extraire pour chaque pompier un ensemble de politiques potentiellement optimales (cf figure 6.3) et de se limiter aux états d'interface avec les autres agents. Nous nous limiterons ainsi par la suite à un problème pour lequel :

- les états d'interface possibles pour un pompier ravitailleur sont au nombre de deux : 'seau plein' ou 'seau vide' et les actions correspondant aux politiques potentiellement optimales sont 'aller chercher de l'eau' ou 'ne rien faire'.
- les états d'interface possibles pour un pompier ravitailleur sont au nombre de deux : 'seau plein' ou 'seau vide' et les actions correspondant aux politiques potentiellement optimales sont 'aller éteindre le feu' ou 'ne rien faire'.
- les états d'interface possibles pour un pompier couloir sont constitué d'une position et du contenu de son seau. Seules quatre positions sont intéressantes : celles consistant à se trouver devant une porte dans une des quatre directions cardinales. Un agent couloir est donc caractérisé par 8 états. Les actions sont au nombre de 4 et correspondent aux politiques potentiellement optimales consistant à se déplacer dans la pièce pour atteindre un état d'interface.

Techniques d'apprentissage Puisque les agents n'ont pas accès aux comportements des autres agents, les techniques de planification sont difficilement envisageables. Nous avons donc opté pour des techniques d'apprentissage par renforcement (cf partie 3.2.4) pour synchroniser l'exécution des politiques.

A l'exécution du système, les agent effectuent simultanément des apprentissages décentralisés. Au cours de leurs apprentissages, les agents construisent leur comportement mais évaluent en parallèle les gains qu'ils espèrent recevoir en effectuant une action a dans un état s par l'intermédiaire des fonction de Q-valeur. Ces évaluations seront utilisées au cours des interactions pour construire les politiques de résolution d'interaction et les politiques de déclenchement à moindre coût.

Dans la partie 6.1.2.3, nous décrirons plus précisément comment cela peut être fait.

Résolution des interactions et répartition des récompenses Dans un Interac-DEC-POMDP, la tâche à accomplir est caractérisée par le signal de récompense réparti entre les agents. Afin de guider les interactions par la tâche, ce sont les signaux de récompenses qui vont guider la résolution des interactions.

Les hypothèses sur lesquelles se fonde cette algorithmique réside dans le fait que les Q-valeurs d'un agent qui intègrent les récompenses qu'il a reçues constituent l'information utile permettant
- de caractériser son comportement vis à vis de la tâche à résoudre : plus les Q-valeurs d'un agent pour un état sont importantes, plus cet agent a la possibilité de générer des récompenses dans le futur en partant de cet état.
- de caractériser ses observations et ses connaissances par rapport à la tâche : comme la fonction de Q-valeur d'un agent dépend de ses perceptions, les Q-valeurs d'un agent à un état donné fournissent des informations quant à l'objectif qu'il pense atteindre à partir de la configuration qu'il a pu observer. Transmettre une Q-valeur permet alors de transmettre de manière implicite les informations liées à la tâche dont dispose l'agent.

La résolution des interactions se fera alors en comparant les intérêts de chaque agent pour un certain résultat d'interaction. Ces intérêts, fonction des espérances de gain des agents, seront évalués par les fonctions de Q-valeurs.

En outre, afin de répartir la résolution de la tâche entre les agents, des signaux de récompenses seront échangés au cours des interactions. Il s'agit de récompenses sociales données par un agent à un autre agent impliqué. Ces récompenses ont pour objectif de motiver un autre agent à réitérer ses actions et à atteindre un état propice à l'exécution d'une interaction directe utile pour le système.

6.1.2.3 Représentation des politiques

Q-learning (Rappel) Dans l'algorithme du Q-learning appliqué dans un système constitué d'un seul agent (cf [WD92] et la partie 3.2.4), l'agent dispose de mémoire à long terme pour pouvoir synthétiser l'ensemble de son expérience et mettre à jour ses comportements. Cette mémoire est constituée de Q-valeurs $Q : S \times A \rightarrow \mathbb{R}$. Au cours de l'apprentissage, cette fonction de Q-valeurs converge vers Q^* qui associe à tout état s et toute action a la somme des récompenses pondérées reçues par l'agent lorsqu'il effectue l'action a à partir de l'état s et suit ensuite la politique optimale.

La politique π est construite par rapport à ces Q-valeurs selon la formule : $\pi(s) = argmax_a\{Q(s,a)\}$ et π peut ainsi être comprise comme une politique paramétrée par les variables que sont les Q-valeurs.

Vue générale Dans un Interac-DEC-POMDP, chaque agent est caractérisé par deux politiques : sa politique d'action et sa politique de déclenchement et chaque couple d'agents par une politique de résolution d'interaction. Pour respecter nos contraintes de localité, nous souhaitons en outre que les politiques de résolution d'interaction soient représentées et mises à jour de manière décentralisée.

Pour cela, on supposera que chaque agent dispose de variables paramétrant ses politiques. Par analogie avec l'algorithme du Q-learning, nous nommerons ces variables Q-valeurs. Chaque agent dispose ainsi de trois types de Q-valeurs : ses Q-valeurs d'action, ses Q-valeurs d'interaction et ses Q-valeurs de déclenchement.

Le sens de ces Q-valeurs est difficile à préciser puisqu'elles ne respectent pas les équations de Bellman (comme cela peut être déjà le cas pour les fonctions de valeur distribuées (*'distributed value functions'*) présentées dans la partie 3.4.3.8). Néanmoins, afin de donner un sens aux équations futures, nous présenterons dans les paragraphes suivants la manière dont il est possible de les interpréter et ce à quoi elles sont censées correspondre.

Q-valeurs d'action Les Q-valeurs d'action sont associées à la politique d'action. Pour un agent, elles associent à chaque état individuel s et chaque action a une valeur numérique $Q_{action}(s, a)$. Ces Q-valeurs synthétisent les récompenses individuelles et les récompenses sociales reçues par l'agent et sont censées représenter l'intérêt qu'a un agent à exécuter une action dans un état donné.

La politique d'action consiste pour un agent à choisir l'action qui lui semble la plus profitable, selon la formule :

$$\pi_i(s_i) = argmax_a\{Q_{action,i}(s_i, a)\}$$

Q-valeurs de déclenchement Les Q-valeurs de déclenchement d'un agent sont associées à sa politique de déclenchement. Elles associent à tout état individuel et tout couple $(interaction, agent^*)$ un indice correspondant à l'intérêt pour le système [12] à déclencher l'interaction considérée. Cet intérêt sera évalué comme la somme des gains de l'ensemble des agents impliqués dans l'interaction (ces gains seront détaillés ultérieurement dans le manuscrit).

La politique de déclenchement d'un agent peut être comprise comme une politique paramétrée par ses Q-valeurs de déclenchement. Elle consiste à choisir l'interaction la plus utile pour le système :

$$\pi_{decl,i}(s_i) = argmax_{I,agent^*}\{Q_{decl,i}(s_i, I, agent^*)\}$$

Q-valeurs d'interaction Les Q-valeurs d'interactions sont elles aussi individuelles, elles sont associées à la politique d'interaction qui est reconstruite au cours des communications locales qui ont lieu à chaque interaction directe entre agents.

Dans Hamelin (cf partie 4), nous avons mis en évidence qu'une confrontation de valeurs de dominance individuelles peut conduire à l'exécution d'interactions utiles pour le système. La valeur de dominance d'un agent peut être comprise comme sa capacité à imposer son point de vue lors de la résolution d'une interaction directe. Pour faire de la résolution d'interaction un processus utile à la tâche, nous avons en outre montré dans le modèle Hamelin que l'introduction de conditions supplémentaires liant la force d'un agent à son état d'insatisfaction [13] permet de restructurer le système quand le besoin s'en fait sentir.

[12] Cet intérêt est une estimation faite par l'agent déclencheur.
[13] comme cela a pu être fait dans le modèle satisfaction altruisme [Sim01]

Nous souhaitons nous inspirer de ce principe pour construire les politiques de résolution d'interactions de manière décentralisée. Nous faisons l'hypothèse par la suite qu'un échange de Q-valeurs locales entre agents permet de prendre une bonne décision collective. La capacité dont dispose un agent pour influencer le résultat d'une interaction sera fonction de ses espérances : si un agent a beaucoup d'intérêt à ce qu'un résultat particulier soit décidé par rapport à un autre, sa capacité à imposer ce résultat sera plus importante. Afin de prendre en compte les points de vue des différents agents, c'est cependant la confrontation des espérances des agents impliqués qui décidera entièrement du résultat.

Chaque agent dispose de Q-valeurs d'interaction individuelles :

$$QInterac : I \times agent^* \times S_{agents} \times Resultat \rightarrow \mathbb{R}$$

Nous supposerons que la Q-valeur d'interaction de l'agent i pour une interaction I_k donnée et un résultat $R_{k,l}$ donné fournit l'intérêt qu'a l'agent i à imposer le résultat $R_{k,l}$ connaissant les les états des agents impliqués dans l'interaction. Au cours de l'interaction, les agents échangent leurs perceptions ainsi que ces Q-valeurs d'interaction pour reconstruire la politique de résolution d'interaction fondée sur ces Q-valeurs d'interaction.

L'objectif étant de résoudre collectivement le problème, la politique d'interaction peut être construite à partir de la maximisation des intêrets des agents impliqués. Ainsi, le résultat choisi par la politique de résolution sera celui maximisant la somme des Q-valeurs d'interaction des agents impliqués [14] :

$$\Pi_{agent^*,I_k}(S_{agents}) = argmax_{R_{k,l}}\{\sum_{agent^*}(QInterac_{agent}(I_k, agent^*, S_{agents}, R_{k,l})\}$$

Ainsi, plus un agent a intérêt à ce qu'un résultat particulier soit exécuté, plus ce résultat aura tendance à être choisi. Les Q-valeurs d'interaction peuvent donc bien être comprises comme la capacité dont dispose un agent pour imposer sa volonté aux autres. La manière de résoudre les interactions permet l'apparition de comportements altruistes : comme la maximisation s'effectue sur la somme des Q-valeurs d'interaction, dans certaines situations, le résultat d'interaction décidé peut conduire à une diminution de l'espérance de gain d'un agent compensée par l'augmentation de l'espérance de gain de l'autre agent impliqué.

Cette genre d'approche est déjà présent dans les travaux de [MHK+98] (cf partie 3.4.3.9) qui cherchent à répartir des ressources entre les agents à partir d'une heuristique identique. Dans [MHK+98], cette heuristique s'exprime sous la forme de gain marginal du à l'attribution d'une ressource à un agent mais correspond exactement à la même opération.

6.1.2.4 Apprentissage des politiques

Nous avons exprimé les différentes politiques des agents sous la forme de fonctions paramétrées par des variables que nous avons nommées Q-valeurs. Le calcul des politiques des agents se

[14] S_{agents} désigne le tuple des états individuels des agents impliqués et $agent^*$ les listes triées d'agents.

résume alors à la construction de ces fonctions de Q-valeurs.

Au cours de l'exécution, trois apprentissages couplés ont lieu dans le système : l'apprentissage des Q-valeurs d'action, l'apprentissage des Q-valeurs de déclenchement et l'apprentissage des Q-valeurs d'interaction.

Afin de mettre en évidence la manière dont ces apprentissages fonctionnent et sont couplés les uns avec les autres, nous présentons leurs principes de manière séquentielle. Il faut cependant garder à l'esprit que tous ces apprentissages se font de manière progressive et de manière cyclique en raison de l'interdependance des politiques (cf partie 6.1.1.3.0).

L'algorithme d'apprentissage sera décrit dans la partie 6.1.3 et formalisera les principes présentés dans cette partie.

Apprentissage des Q-valeurs d'interaction L'apprentissage des Q-valeurs d'interaction consiste pour chaque agent à évaluer l'intérêt qu'il a pour un résultat d'interaction donné. Cet intérêt est évalué comme la Q-valeur d'action maximale associée à l'état de l'agent après exécution du résultat d'interaction. Ainsi, un résultat d'interaction conduisant un agent vers un état pour lequel l'agent dispose de Q-valeurs d'action importantes sera censé être intéressant pour l'agent considéré. Après exécution d'un résultat d'interaction, chaque agent met à jour ses Q-valeurs d'interaction en fonction des Q-valeurs d'action de l'état d'arrivée.

Enfin, afin d'inciter les agents à reproduire une situation d'interaction utile pour le système, une récompense est transmise au cours de l'interaction. Cette récompense est une récompense sociale donnée par l'agent ayant eu le plus d'intérêt à choisir le résultat d'interaction vers les autres agents. Cet **échange de récompense** a pour objectif de répartir la tâche à résoudre au sein de la communauté d'agents. Il permet de répartir la tâche entre les différents individus même si certains n'accèdent pas directement à des récompenses et constitue une réponse au problème du 'credit assignment' que nous avons mis en évidence dans la partie 3.4.3.

La récompense sociale est évaluée en fonction du gain global de l'interaction pour le système. Ce gain correspond à la différence des espérances de gain du système global avant et après l'interaction. Comme les états des agents impliqués dans l'interaction sont les seuls états individuels modifiés, ce gain peut être calculé localement par les agents impliqués comme la différence de la sommes de leurs Q-valeurs d'actions entre avant et après l'exécution de l'interaction directe.

Apprentissage des Q-valeurs de déclenchement Les Q-valeurs de déclenchement sont mises à jour à partir de ce même gain global associé à l'interaction directe. Plus ce gain collectif est important, plus la Q-valeur de déclenchement sera importante et plus l'agent aura tendance à déclencher cette interaction.

Considérer le gain collectif pour le déclenchement des interactions permet à un agent de déclencher des interactions altruistes à l'issue desquelles ses espérances de gain vont diminuer mais qui peuvent permettre d'améliorer les performances globales du système.

Apprentissage des Q-valeurs d'action Enfin, il reste à apprendre les Q-valeurs d'action. Une action peut être renforcée de deux manières dans le système.

Elle peut être renforcée parce que l'agent en l'exécutant reçoit une récompense locale associée à l'avancement local de la tâche à résoudre. En cela il s'agit d'une récompense obtenue de la même manière que celle qu'il est possible d'obtenir dans un MDP.

Une action peut aussi être renforcée parce que la position après action de l'agent permet aux agents d'effectuer une interaction utile pour le système. Cette considération est implémentée par la notion de récompense sociale transmise entre les agents. Les récompenses sociales permettent ainsi de renforcer certaines actions parce qu'elles permettent à d'autres agents d'interagir et de générer une récompense globale non perceptible directement par l'agent.

L'apprentissage des Q-valeurs d'action se fait à l'aide d'un Q-learning classique mais intégrant les récompenses individuelles distribuées dans le système et les récompenses sociales transmises entre les agents.

Les interactions et leur exécution constituent au cours de l'apprentissage des Q-valeurs d'action des transitions spontanées dans le système qui ne sont pas considérées directement par les agents mais se répercutent sur leur politique par l'intermédiaire des récompenses sociales.

6.1.3 Processus de construction des comportements

Maintenant que nous avons décrit les principes de cet apprentissage, cette partie présente l'algorithme d'apprentissage de manière détaillée et la manière dont il s'insère dans l'algorithme d'exécution de l'Interac-DEC-POMDP.

6.1.3.1 Cycle d'apprentissage

Cette partie présente de manière séquentielle les différentes étapes d'un cycle d'apprentissage.

Exécution des actions Un agent décide de son action en fonction de sa politique d'action déterminée par ses Q-valeurs d'action. Afin de permettre l'exploration de nouvelles pistes, cette politique est stochastique et a tendance à privilégier l'action la plus prometteuse. Pour ce faire, nous avons utilisé des politiques ϵ-greedy caractérisées pour un ϵ donné par la formule suivante :

$$a_i = \pi_i(s_i) = \begin{cases} \text{aléatoire} & \text{probabilité} \quad (\epsilon) \\ argmax_a(Q_{action}(s_i, a)) & \text{probabilité} \quad (1-\epsilon) \end{cases}$$

Tous les agents choisissent et émettent simultanément leur action et le système évolue en fonction de l'action jointe. Chaque agent reçoit une récompense individuelle r_i et se trouve désormais dans l'état $s'_i = T_i(s_i, a_i)$.

Déclenchement d'interactions Les interactions sont déclenchées et résolues séquentiellement. Chaque agent est sollicité pour déclencher une interaction et exécuter un résultat. Sans perte de généralité, nous supposerons que l'agent 1 déclenche une interaction avec l'agent 2.

De la même manière que les politiques d'action, les politiques de déclenchement sont des politiques ϵ-greedy (avec le même ϵ). Les deux interactions possibles étant un échange ou une

interaction vide, il peut arriver qu'aucune interaction ne soit déclenchée.

On considérera en outre par la suite que lorsqu'un agent a autant intérêt à déclencher une interaction directe correspondant à un échange qu'aucune interaction, il préférera choisir de ne pas déclencher d'interaction (hors phase d'exploration). Cet ajout permet de limiter à l'exécution les interactions entre agents[15].

Choix du résultat de l'interaction La résolution d'une interaction consiste à choisir un résultat d'interaction parmi les résultats possibles (échange effectif ou non dans notre cas). L'heuristique que nous avons proposée consiste à maximiser (avec un facteur d'exploration $0.2 * \epsilon$ permettant de rendre l'exécution d'interaction plus stable) la somme des Q-valeurs individuelles d'interaction à partir de communication locales selon la formule :

$$R_{k,l} \rightarrow argmax_{R_{k,l}}\{(QInterac_1(I_k, agent_2, S_{agents}, R_{k,l}) + (QInterac_2(I_k, agent_1, S_{agents}, R_{k,l})\}$$

Exécution du résultat L'interaction est exécutée et les états des agents concernés sont modifiés :

$$(s''_1, s''_2) \leftarrow TRI_{R_k}(s'_1, s'_2)$$

Transfert de récompense sociale Afin de prendre en compte l'interaction dans le long terme et d'inciter les agents à reproduire l'interaction si elle est bénéfique, le gain collectif obtenu par exécution de l'interaction est réparti entre les agents sous forme d'un transfert de récompense sociale. Ce gain collectif peut s'exprimer par la différence entre la somme des fonctions de valeurs d'action des agents impliqués avant et après interaction. La seconde heuristique que nous proposons consiste à répartir ce gain de manière équitable pour inciter au mieux les agents à reproduire leurs actions.

De manière analogue aux fonctions de valeurs classiques, on notera

$$v(s) = max_a Q(s, a)$$

$$\begin{aligned} gain_1 &= v_{action,1}(s''_1) - v_{action,1}(s'_1) \\ gain_2 &= v_{action,2}(s''_2) - v_{action,2}(s'_2) \\ r_{s,1} &= -gain_1 + \lambda(gain_1 + gain_2) \\ r_{s,2} &= -gain_2 + \lambda(gain_1 + gain_2) = -r_{s,1} \end{aligned}$$

avec $\lambda = 0.5$

Apprentissage des interactions *Résolution*

Chaque agent peut apprendre après exécution les conséquences de l'interaction. Pour cela, chaque agent met à jour sa Q-valeur d'interaction en fonction de la valeur d'action de l'état après

[15] Nous supposerons de manière implicite que les communications ont un coût de manière implicite et qu'il vaut donc mieux limiter leurs utilisations.

interaction (comme nous supposerons que les transitions d'interaction sont déterministes, nous assimilons la Q-valeur d'interaction à la Q-valeur d'action aprés interaction).

$$QInterac_1(Ik, agent_2, s'_1, s'_2, R_{k,l}) = v_{action,1}(s''_1)$$
$$QInterac_2(Ik, agent_1, s'_1, s'_2, R_{k,l}) = v_{action,2}(s''_2)$$

Déclenchement

De la même manière, l'agent 1 déclencheur de l'interaction met à jour ses Q-valeurs de déclenchement à partir du gain global du système.

$$Q_{decl,1}(s'_1, Ik, agent_2) = (1-\alpha)Q_{decl,1}(s'_1, Ik, agent_2) + (\alpha)(gain_1 + gain_2)$$

Comme pour le Q-learning, α désigne un coefficient d'apprentissage permettant de conserver une mémoire du passé.

Apprentissage des actions A l'issue de l'ensemble des interactions, chaque agent i est supposé se trouver dans un état s'''_i (et non pas s''_i puisque d'autres interactions ont pu avoir été exécutées après celle menant dans l'état s''_i).

Chaque agent met à jour sa Q-valeur d'action de départ avant action $Q_{action,i}(s_i, a_i)$ en fonction
- des récompenses reçues au cours de la phase d'action r_i
- des récompenses sociales reçues au cours des interactions $r_{s,i}$
- de la Q-valeur de l'état d'arrivée s'''_i aprés l'exécution de l'ensemble des interactions

La formule de mise à jour des Q-valeurs est analogue à celle du Q-learning [WD92] ($\gamma \in [0,1]$ est le facteur de décompte (*discount factor*))

$$Q_{action,i}(s_i, a_i) = (1-\alpha)Q_{action,i}(s_i, a_i) + \alpha(r_i + r_{s,i} + \gamma v(s'''_i))$$

6.1.3.2 Stratégie d'apprentissage

L'algorithme d'apprentissage est caractérisée par deux paramètres globaux :
- le paramètre ϵ qui traite du dilemme exploration-exploitation.
- le paramètre α qui correspond à un coefficient d'apprentissage.

Les paramètres ϵ et α dont fortement couplés dans le cas d'un apprentissage multi-agents :
- Le coefficient ϵ rend les politiques d'actions, de déclenchement et d'interaction stochastiques : plus ϵ est important, moins les politiques sont déterministes et plus il est difficile pour un agent de prévoir le comportement des autres agents.
- Le coefficient α correspond au coefficient d'apprentissage des politiques d'action et de déclenchement. Une valeur de α non nulle permet de conserver une mémoire des actions et des déclenchements passés. Plus la valeur d'*alpha* est importante plus l'agent considéré accorder de crédit à sa dernière expérience par rapport aux expériences passées.

Afin d'éviter que les agents ne modifient trop vite leurs politiques et que celles ci n'oscillent, nous avons choisi de prendre un coefficient α faible ce qui conduit les agents à évaluer leurs Q-valeurs à partir d'un grand nombre d'expériences et de conserver une mémoire importante

du passé. Nous avons de plus choisi un coefficient ϵ convergeant vers 0 au fur et à mesure de l'apprentissage conduisant les agents d'un comportement stochastique vers un comportement déterministe.

Le comportement global du système consiste ainsi
- D'abord en une exploration du système permettant aux agents d'apprendre les conséquences de leurs actions en terme de récompense individuelle (valeur initiale de ϵ importante). Comme les comportements des agents sont fortement stochastiques, les agents ne peuvent par contre pas retirer énormément d'information quant aux comportement des autres agents.
- Ensuite, en une exploration de plus en plus réduite (caractérisée par la diminution du coefficient ϵ) : au cours de cette phase, les agents stabilisent progressivement leurs politiques tout en explorant de nouvelles pistes. Cette phase a pour objectif de synchroniser les comportements des agents en tirant parti des tendances apprises dans la phase précédente
- Enfin une exploitation totale (ϵ égal ou proche de 0) permettant aux agents d'évaluer plus précisément leurs fonction de Q-valeurs et de remettre en cause leur comportement.

6.1.3.3 Algorithme

L'algorithme 9 présente le fonctionnement général de l'Interac-DEC-POMDP au cours d'apprentissage (les ajouts par rapport à l'algorithme d'exécution et concernant l'apprentissage sont mis en évidence par •)

6.1.3.4 Un exemple d'interaction

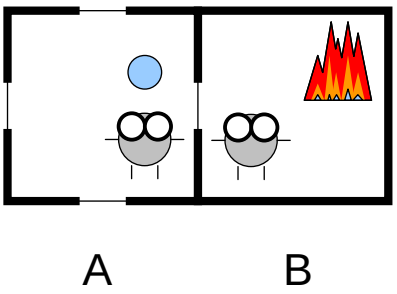


FIG. 6.4 – Exemple de résolution d'interaction

Considérons un agent pompier-couloir A avec un seau plein et un agent pompier-extincteur B avec un seau vide (cf fig 6.4) qui viennent d'états $avant_A$ et $avant_B$. Lorsqu'une interaction est déclenchée entre l'agent A et l'agent B, deux résultats sont possibles :
- le seau rempli est effectivement échangé et est transmis de l'agent A à l'agent B
- le seau rempli n'est pas échangé et reste en la possession de l'agent A

Rappel : on notera $v_{action}(s) = max_a Q_{action}(s, a)$

Supposons en outre que, pour simplifier, le système s'arrête au prochain pas de temps et que les Q-valeurs des agents sont les suivantes[16] :

[16] Les Q-valeurs apprises ne reflètent donc pas des exécutions à long terme ce qui explique que leur valeur

Algorithme 9 Apprentissage pendant t_fin pas de temps

$t \leftarrow 0$
répéter
Module d'action
Observation : $O = (o_0, ...o_n) \leftarrow O(s)$
pour tout agent $i \in [0..n]$ **faire**
choix de l'action i : $a_i \leftarrow \pi_i(o_i)$
fin pour
action jointe : $a \leftarrow (a_0...a_n)$
exécution de l'action jointe : $s' \leftarrow T(s, a)$
récompense pour chaque agent : $r_i \leftarrow [R(s, a, s')]_i$
modification de l'état du système : $s \leftarrow s'$
Module d'interaction
Liste aléatoire des agents : $liste \leftarrow ordre_{aleatoire}$
pour tout agent $\in liste$ **faire**
Observation de l'agent i : $o_i \leftarrow O(s)_i$
Choix de l'interaction : $(I_k, agent_j) \leftarrow \pi_{decl,i}(o_i)$
si $possible(I_k, agent_j)$ **alors**
Observation de l'agent j : $o_j \leftarrow O(s)_j$
Échanges locaux et choix du résultat de l'interaction : $RI_{k,l} \leftarrow \Pi_{i,j}(o_i, o_j)$
• Calcul des récompenses sociales : $r_{soc,i,j}$
Exécution du résultat : $s \leftarrow TRI_{k,l}(S, i, j)$
• Mise à jour des Q-valeurs d'interaction
• Mise à jour des Q-valeurs de déclenchement
Fin de l'interaction
fin si
fin pour
• Mise à jour des Qvaleurs d'action
$t \leftarrow t + 1$
jusqu'à $t > t_{fin}$

– Lorsque l'agent A possède un seau vide, il a une fonction de valeur d'action à 0 puisqu'il n'a pas moyen d'accéder à une récompense individuelle. $v_A(vide) = 0$. Nous supposerons pour l'exemple, que si l'agent A conserve le seau, il reçoit une récompense positive de 50 qu'il a déjà intégré dans sa fonction de valeur d'action : $v_A(plein) = 50$
– Lorsque l'agent B possède un seau rempli, il peut éteindre le feu et recevoir une récompense de $+100$. On supposera donc que sa fonction de valeur d'action vérifie $v_B(plein) = 100$. Si l'agent B ne possède pas de seau plein, il ne peut pas éteindre le feu et accéder à une récompense. La valeur associée à l'état 'vide' est donc nulle : $v_B(vide) = 0$

Les résultats possibles d'interaction ont pour valeurs respectives :
– $v(echange) = v_A(vide) + v_B(plein) = 100$
– $v(rien) = v_A(plein) + v_B(vide) = 50$

Le résultat choisi au cours de l'interaction sera donc un échange effectif car il est plus intéressant pour le couple d'agents que ce soit l'agent B qui possède le seau. Ainsi, alors que l'action égoïste pour l'agent A serait de conserver son seau en espérant recevoir une récompense de $+50$

correspond aux récompenses instantanées.

dans le futur, l'échange a néanmoins lieu.

Dès lors, les gains dûs à l'exécution de l'interaction seront :
- pour l'agent A : $gain_A = v_A(vide) - v_A(plein) = -50$, une perte de 50 conséquence de la perte du seau plein
- pour l'agent B : $gain_B = v_B(plein) - v_B(vide) = 100$, un gain de 100 dû à l'apport d'un seau plein permettant à l'agent d'éteindre le feu
- pour le système : $gain_S = gain_A + gain_B = 50$

Les récompenses sociales seront calculées de telle manière que le gain collectif est réparti équitablement entre les agents :
- pour l'agent A : $r_{s,A} = -gain_A + 0.5(gain_A + gain_B) = 50 + 25 = 75$
- pour l'agent B : $r_{s,B} = -75$

Les récompenses sociales permettent alors de modifier les Q-valeurs des agents correspondant à l'état *avant* en tenant compte des interactions :
- la valeur de l'agent A sera de $v_A(avant_A) = v_A(vide) + r_{s,A} = 75$ à posséder un seau rempli puisqu'une fois qu'il en possède un il sait que l'agent B lui donnera une récompense sociale de 75 dans le futur.
- la Valeur de l'agent B sera de $v_B(avant_B) = v_B(plein) + r_{s,B} = 25$ 25 à aller chercher un seau. Cette valeur est à mettre en relation avec la valeur 0 consistant à ne pas aller chercher de seau. L'agent est donc incité à aller chercher un seau mais du fait qu'il doit transmettre une partie de sa récompense à l'agent A, il ne peut esperer que 25 ce qui suffit à l'inciter

Cet exemple reste néanmoins simple puisqu'il faut prendre en compte les influences réciproques des actions et des interactions à long terme pour le calcul des Q-valeurs des agents.

6.1.4 Synthèse de l'approche de résolution

6.1.4.1 Problème posé

Dans cette partie, nous nous sommes concentrés sur une classe de sous-problèmes du formalisme Interac-DEC-POMDP. Cette classe est caractérisée par :
- le fait que les actions d'un agent n'ont pas d'influence sur les conséquences des autres actions envisagées par les agents
- le fait que les seules influences possibles entre agents résident dans les interactions
- le fait que la tâche à résoudre est évaluée localement par les agents.

De plus, le problème des pompiers vérifie un ensemble plus important de contraintes. Les interactions ponctuelles considérées sont limitées à deux agents, ne peuvent avoir lieu que dans certaines situations globales et ne peuvent être que d'un seul type (échange de seaux). Enfin, à un pas de temps donné, un agent ne peut être en interaction qu'avec un seul autre agent.

Résoudre ce type de problème reste néanmoins non trivial et de nombreux problèmes équivalents sont posés dans plusieurs domaines (partage de charge dans un réseau, organisation de chaînes de production). Ces types de problèmes induisent des problèmes de coordination importants entre les agents :

Tout d'abord, les prises de décisions individuelles et collectives sont couplées puisque c'est la dynamique globale du système qui fournit une réponse au problème.

Ensuite, les comportements des agents sont couplés. Les agents ne disposent que de perceptions partielles de l'état global du système et ne connaissent pas les actions émises par les autres agents ni leur comportement.

Enfin, le problème traité est un problème purement collectif : chaque agent doit intégrer à un instant ou un autre les autres agents du système pour effectuer une action pertinente. La présence d'un autre agent même situé loin peut modifier totalement le comportement à émettre (par exemple la présence d'un agent extincteur au bout d'une suite d'agents couloir).

6.1.4.2 Algorithmique

Nous avons proposé un algorithme pour tirer parti de la notion d'interaction. Afin de respecter les contraintes de localité présentées dans la partie 2, cet algorithme est basé sur sur des apprentissages distribués permettant à chaque agent de mettre à jour localement ses politiques et de s'affranchir en partie d'un modèle explicite de son environnement et des autres agents.

Pour se faire, il se fonde sur plusieurs principes :
- le premier principe stipule que les échanges de valeurs entre agents au cours de l'interaction suffisent pour prendre des décisions collectives intéressantes pour le système.
- le second consiste à reconstituer les politiques jointes à l'exécution du système grâce à une heuristique et à des échanges de valeurs numériques entre agents
- le troisième consiste à effectuer des échanges de récompenses sociales au cours des interactions directes afin d'inciter les agents qui bénéficient le moins de l'interaction à reproduire les situations d'interaction

6.2 Expérimentations et validation de notre approche

Afin d'évaluer notre approche, plusieurs expériences ont été faites. Une expérience consiste à définir une instance de problème constituée de plusieurs briques d'agents (couloir, feu, eau) en interaction et à effectuer un apprentissage dans ce système.

Une partie de ces expériences ont été décrites par ailleurs dans [TBC06].

6.2.1 Critères d'analyse et plans expérimentaux associés

Propriétés attendues Nous allons détailler les résultats qu'il est possible d'obtenir par notre apprentissage en mettant l'accent sur un certain nombre de points.

Dans un système markovien, la tâche est définie par rapport à une fonction de récompense. Comme nous nous intéressons à un système coopératif, cette récompense est globale et a été définie comme la somme des récompenses reçues individuellement par les agents. Le système sera donc évalué dans un premier temps par rapport à la quantité globale de récompenses qu'il reçoit lorsque les politiques des agents sont exécutées de manière décentralisée.

Dans un second temps, nous nous concentrerons plus sur les aspects qualitatifs des politiques construites. Nous nous focaliserons sur l'apparition spontanée d'organisations. Une organisation sera définie comme un ensemble d'agents liés de proche en proche par l'utilisation effective d'interactions les impliquant deux par deux. Dans le problème des pompiers, cette organisation correspondra à des chaînes d'agents permettant le transport des seaux des agents-ravitailleurs

aux agents-extincteur. L'apparition d'organisation utile à la tâche est la conséquence de plusieurs phénomènes imbriqués :

- le déclenchement individuel et l'utilisation adéquate des interactions dans le système consistant à échanger des seaux dans la bonne direction : des agents les plus loin du feu vers les agents les plus proches.
- les transferts de récompenses entre agents permettant de guider les agents n'ayant pas accès directement aux récompenses à reproduire certaines actions même s'ils n'en reçoivent pas un bénéfice direct
- la synchronisation des comportements consistant aux agents à reproduire des situations globales permettant des interactions utiles pour le système.

Nous évaluerons ces organisations en terme :

- **stabilité** : comme les Q-valeurs ne respectent plus les équations de Bellman, elles ne possèdent plus les propriétés de convergence de Q-valeurs classiques. Nous nous focaliserons sur la convergence et la stabilité de ces Q-valeurs au cours de l'exploitation des politiques.
- **robustesse** : nous étudierons la réaction des organisations générées et des comportements individuels face à des perturbations afin d'évaluer la robustesse et la capacité du système à se re-configurer à l'exécution. En modifiant les récompenses liées aux actions des agents, les interactions doivent apparaître et disparaître en fonction d'un critère global non perceptible par les agents. Ainsi, par exemple, si la sommes des coûts (récompenses négatives réparties au sein des agents) nécessaires à amener le seau est supérieure à la récompense reçue lorsque l'on éteint un feu, les organisations ne sont pas utiles au système et il préférable de ne rien faire.
- **passage à l'échelle** : nous nous intéresserons à la manière dont le système réagit à l'ajout et au retrait d'agent à l'exécution et à des systèmes constitués par un nombre important d'agents.
- **pertinence de l'organisation générée** : nous nous intéresserons aux organisations qui peuvent apparaître lorsque plusieurs organisations sont possibles et intéressantes pour le système.

Plan d'expériences La démarche sous-tendant les expériences décrites dans cette partie s'organise de la manière suivante :

- Dans un premier temps nous analyserons les résultats obtenus sur un exemple simple et nous nous intéresserons à la convergence et la stabilité du système
- Dans un second temps, nous comparerons ces résultats à un problème analogue modélisé dans un DEC-POMDP pour mettre en évidence l'utilité de la notion d'interaction
- Nous nous focaliserons sur des systèmes plus complexes avec un nombre d'agents importants pour plusieurs configurations données.
- Nous nous intéresserons à l'ajout et au retrait d'agents au cours de l'exécution du système pour tester sa capacité à se réorganiser et à intégrer des modifications des structures relationnelles entre agents
- Nous nous intéresserons à la résistance du système à l'ajout de récompenses individuelles.

6.2.2 Résultats bruts

6.2.2.1 Convergence

Expérience :

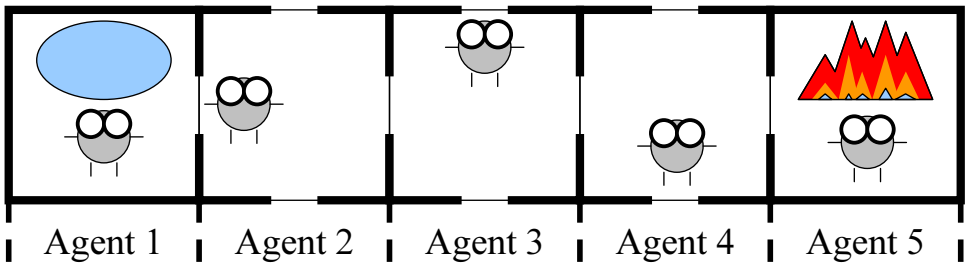


FIG. 6.5 – Exemple constitué par 5 agents

Nous avons tout d'abord conduit un apprentissage de 20000 pas de temps avec un facteur $\alpha = 0.02$ et ϵ décroissant linéairement de 1 à 0 sur un exemple constitué de 5 agents dans un couloir (cf fig 6.5).

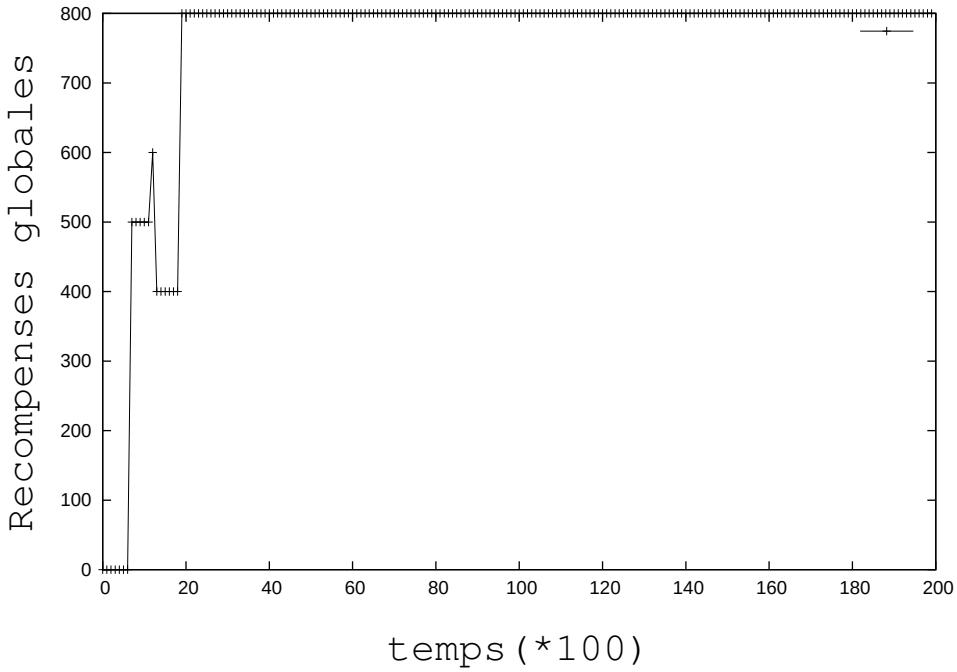


FIG. 6.6 – Récompenses reçues par exploitation des politiques pendant 20 pas de temps ($\epsilon = 0$) au cours de l'apprentissage en fonction de t

Tous les 100 pas de temps, le système est réinitialisé, les seaux disparaissent et les politiques construites sont exécutées sans exploration (à savoir avec un facteur ϵ nul) pendant 20 pas de temps et l'ensemble des récompenses reçues par le système durant cette période est mesurée. Ces données sont représentées sur la courbe de la figure 6.6.

Observations :

Au départ, seul l'agent extincteur accède directement à des récompenses en éteignant le feu. Au cours des interactions qui ont lieu, on observe des transferts de récompenses qui permettent

de distribuer la tâche au sein des agents. Chaque agent apprend

- à effectuer des échanges de seaux remplis uniquement de la gauche vers la droite du fait de la maximisation de la somme des Q-valeurs
- à amener le seau rempli à l'agent qui se trouve à sa droite du fait des transferts de récompenses sociales
- à aller chercher le seau vers l'agent qui se trouve sur sa gauche qui est incité à en amener un du fait des transferts de récompenses sociales.

Ces apprentissages ne doivent pas être considérés séparément les uns des autres mais sont la conséquence des apprentissages couplés des actions et des interactions.

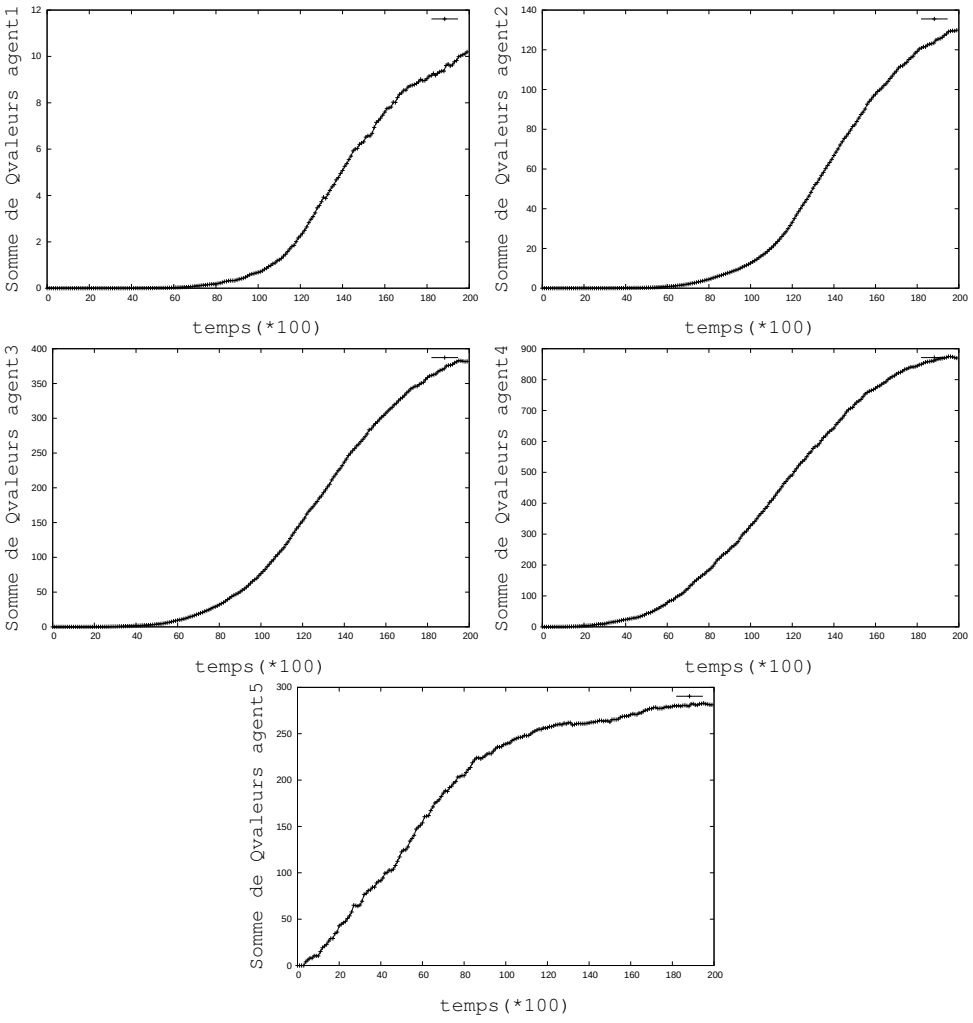


FIG. 6.7 – Évolution de la somme des Q-valeurs d'action pour chaque agent durant l'apprentissage avec $\alpha = 0.02$

Cette organisation permet d'éteindre le feu comme l'illustre les récompenses reçues par le système au cours de son apprentissage (cf figure 6.6). Elle parvient à apporter des seaux plein à un taux constant une fois le régime transitoire dépassé. Ce régime transitoire consiste à effectuer un premier transport de seau de l'agent extincteur à l'agent ravitailleur puisqu'à la réinitialisation de l'état global aucun seau n'est disponible. Le comportement collectif s'avère dans ce cas

simple être optimal (un exemple d'exécution de système est présenté en annexe 2).

De plus, bien que les agents ne perçoivent pas la présence des autres agents au moment du déclenchement, nous avons constaté que seules les interactions potentiellement utiles sont déclenchées :

- Pour un agent couloir, cela consiste à déclencher une interaction lorsqu'il se trouve à gauche avec un seau vide ou à droite avec un seau plein.
- Pour un agent extincteur, cela consiste à déclencher une interaction quand son seau est vide.
- Pour un agent ravitailleur, cela consiste à déclencher une interaction après avoir rempli son seau.

Bien entendu, au cours de l'exploration, il peut arriver qu'un déclenchement d'interaction soit inutile puisque l'agent déclencheur ne sait pas a priori si un autre agent est prêt à répondre à cette interaction. Mais, du fait d'un effet de moyenne, les agents parviennent à apprendre les déclenchements potentiellement utiles.

Enfin, la figure 6.7 présente l'évolution de la somme des Q-valeurs d'action des agents au fur et à mesure de l'apprentissage. Comme les valeurs présentées correspondent à une somme sur l'ensemble des états et des actions des Q-valeurs d'action, elles sont difficilement interprétables par contre la forme des courbes fournissent une indication concernant la convergence de l'algorithme. Elles mettent ainsi en évidence la présence de transfert de récompense entre les agents (les Q-valeurs des agents situés loin du feu augmentent alors qu'ils n'ont pas accès directement à des récompenses) et le système converge (ce qui est caractérisé par l'absence d'oscillations qui pourraient apparaître du fait de la présence d'apprentissages simultanés).

L'exécution de ces politiques parvient à reproduire le comportement global optimal du système. Un exemple d'exécution est présenté en annexe2 et présente comment s'agencent les déplacements des agents et les interactions.

Conclusion :

L'approche que nous avons proposée permet de générer automatiquement une organisation (chaînage de déplacements et de transferts de seaux entre les agents) utile à la tâche de manière entièrement décentralisée et sans que les agents n'aient de vue globale du système. Les états locaux des agents jouent le rôle de mémoire pour le système. Ainsi les comportements des agents couloirs obtenus consistent à

- se déplacer sur la droite si les agents possèdent un seau
- se déplacer sur la gauche si les agents n'en possèdent pas

En conséquence, les trois apprentissages que nous avons décrits dans la partie précédente permettent :

- d'apprendre les politiques de résolution d'interaction pour effectuer les échanges de seaux dans la bonnes direction.
- d'apprendre les politiques de déclenchement consistant à déclencher les interaction potentiellement utiles (les Q-valeurs de déclenchement des autres interactions restent nulles).
- d'apprendre les politiques d'action consistant à se déplacer dans la pièce pour reproduire des situations d'interaction potentiellement utiles.

6.2.2.2 Stabilité

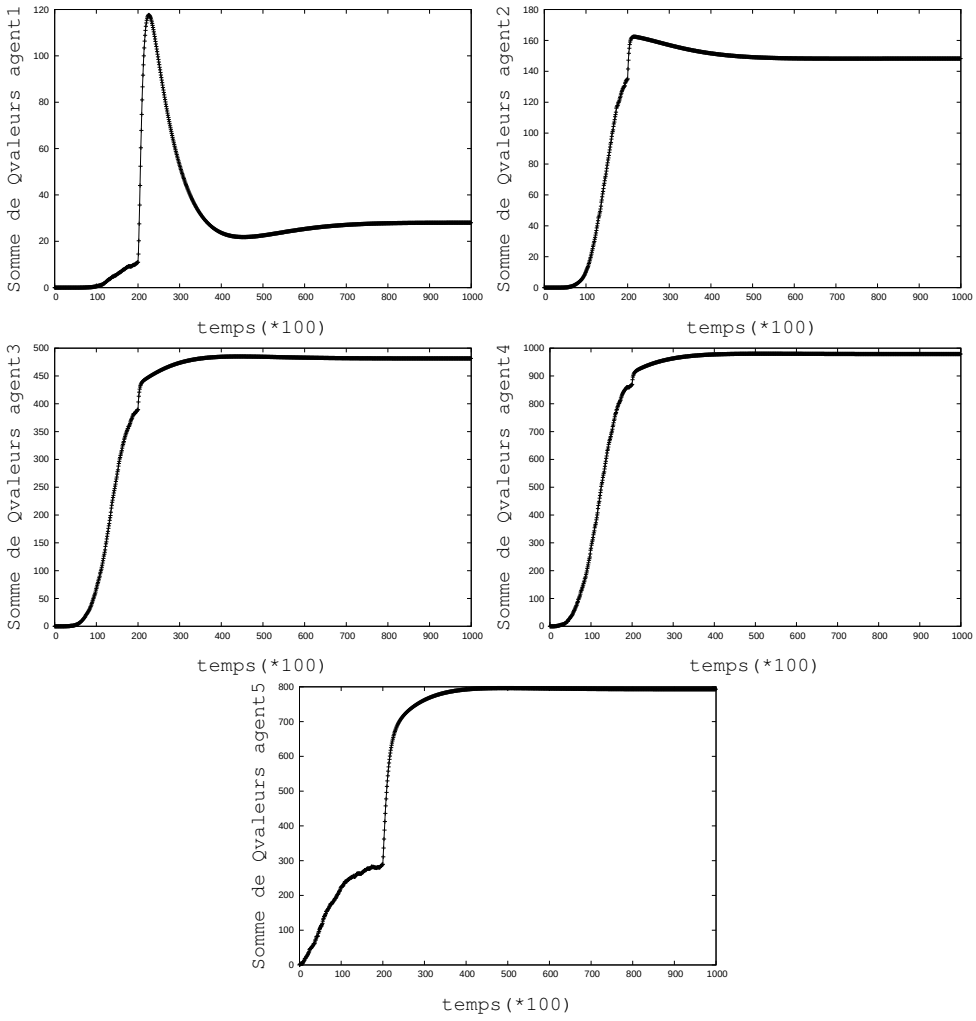


FIG. 6.8 – Évolution de la somme des Q-valeurs d'action pour chaque agent durant l'apprentissage prolongé

Expérience :

Afin d'évaluer la stabilité de l'organisation générée, nous avons conduit d'autres expériences identiques à la précédente, si ce n'est que le système continue à exploiter la politique trouvée pendant 80000 pas de temps supplémentaires (c'est-à-dire quatre fois plus longtemps).

Observations :

Au cours de cette exploitation, les fonctions de Q-valeurs continuent à évoluer du fait des récompenses données par l'environnement et des récompenses sociales. Nous avons pu constater néanmoins qu'elles se stabilisent au cours du temps (cf figure 6.8). De plus, même si les Q-valeurs évoluent encore à cause d'une re-répartition des récompenses dans le système, la politique globale reste inchangée.

Il est possible de distinguer deux phases dans l'évolution des Q-valeurs (cf fig 6.8).
- La première phase se déroule les 200000 premiers pas de temps. Durant cette phase les agents explorent leur environnement et le coefficient ϵ décroît linéairement de 1 vers 0
- La seconde phase a lieu ensuite. Durant cette phase, les agents exploitent leurs politiques tout en continuant à apprendre. Comme les politiques des agents ne sont plus stochastiques, il n'y a plus d'aléas dans le système et les agents sont plus aptes à extraire de l'information pertinente de leurs expériences, ce qui explique le changement des vitesses d'évolution des Q-valeurs (cf fig 6.8).

Interprétations :

La forme particulière des courbes des sommes de Q-valeurs dans la deuxième phase peut s'expliquer par le fait que les récompenses sociales sont évaluées à partir des valeurs d'actions des agents. On peut aussi diviser cette période d'exploitation en deux autres périodes
- Dans la première, les agents apprennent leurs Q-valeurs d'action à partir des récompenses données par l'environnement et des récompenses sociales. Les récompenses données par l'environnement ne sont pas encore intégrées dans le système et sont réparties entre les agents ce qui explique que la somme des Q-valeurs des agents augmente (y compris pour l'agent 1).
- Dans la seconde phase, les récompenses données par l'environnement sont intégrées dans les Q-valeurs d'action des agents et ne les font plus évoluer mais un autre phénomène prend le dessus. Les agents en distribuant et en recevant des récompenses sociales modifient leurs Q-valeurs d'actions. Ainsi, à un instant t, un agent extincteur donne une certaine récompense à un agent couloir qui lui amène un seau. Cette récompense est évaluée en fonction de ces Q-valeurs d'action à l'instant t, mais le fait de distribuer une récompense sociale réduit ces Q-valeurs et les récompenses sociales futures en conséquence. Ainsi, à l'interaction suivante, l'agent couloir recevra une récompense plus faible que celle qui avait été utilisée pour mettre à jour ses Q-valeurs. Ces phénomènes expliquent les changements d'allure des courbes d'évolution des Q-valeurs d'action et le fait que certaines d'entre elles re-diminuent par la suite.

Conclusion :

Bien que les Q-valeurs des agents continuent à évoluer au cours de l'exploitation des politiques du fait des transferts de récompenses sociales, ces Q-valeurs finissent par se stabiliser. De plus, cette évolution guidée par l'exécution des mêmes interactions ne remet par contre pas en cause les comportements globaux du système. Les organisations que nous parvenons à construire sont donc bien stables dans le temps.

6.2.2.3 Influence du coefficient d'apprentissage

Expérience :

Enfin, nous nous sommes intéressés à l'influence du coefficient d'apprentissage α sur le comportement collectif construit. Pour cela, nous avons effectué les mêmes expériences que précédemment en utilisant un coefficient α égal à 0.2 (au lieu de 0.02).

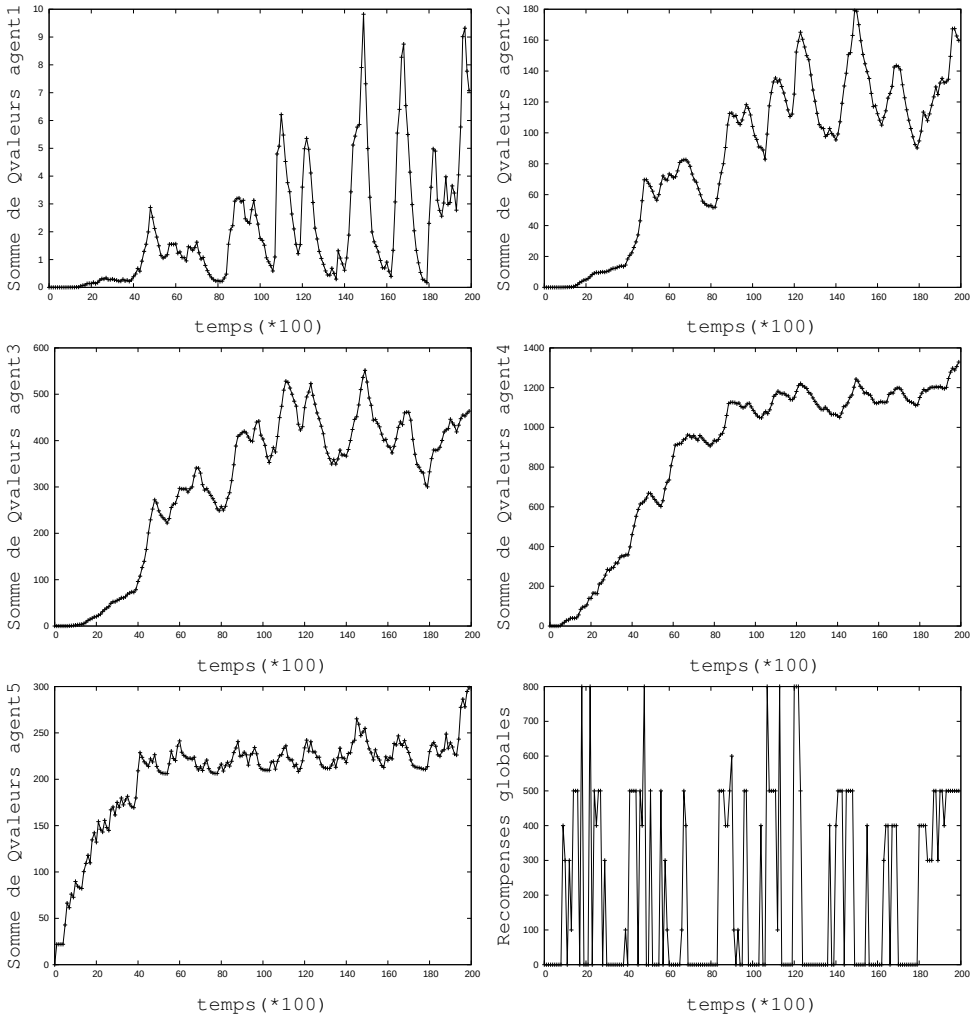


FIG. 6.9 – Influence de α sur l'apprentissage : Résultats obtenus par exploitation suite à un apprentissage effectué avec alpha=0.2 sur le problème de la figure 6.5

Observations :

Dans ce cas, des phénomènes oscillatoires apparaissent et sont d'autant plus marqués que les agents se situent loin des récompenses générées (en terme de nombre d'interactions) (cf figure 6.9).

Interprétations :

Ces phénomènes proviennent du fait que les agents apprennent trop rapidement et ne considèrent pas assez la présence d'aléas au cours de la phase d'exploration. En effet, l'intérêt de disposer d'un coefficient α constant réside dans le fait que les expériences passées ont tendance à être oubliées. L'influence d'une expérience passée sur sur les Q-valeurs actuelles est de l'ordre d'une fonction exponentielle décroissante en $(1-\alpha).t$ [17].

[17] puisque qu'à chaque modification des Q-valeur l'ancienne Q-valeur qui synthétise l'ensemble des expériences

Lorsque α est faible, cela permet d'oublier les apprentissages des phases les plus exploratoires au fur et à mesure que ϵ (et donc l'aspect stochastique des politiques) décroît. Lorsque α est trop important, un agent a tendance à résumer son expérience du système uniquement à ses dernières expériences.

Ainsi, dans ce dernier cas, au cours de la phase exploratoire, lorsqu'un agent A reçoit une récompense sociale, il va chercher à reproduire la situation permettant l'interaction pour pouvoir recevoir la récompense sociale par la suite. Or l'autre agent étant en train d'explorer son environnement, il y a peu de chance qu'il soit apte à être dans état permettant de répondre à l'interaction. L'agent A va alors avoir tendance à oublier trop rapidement qu'il a la possibilité de recevoir des récompenses sociales.

Conclusion :

Ces expériences mettent ainsi en évidence le couplage très important qui peut exister entre les paramètres d'apprentissages et le dilemme exploration exploitation :

- lorsque l'exploration est importante, comme les lois du monde vues par agent dépendent des politiques des autres agents, l'agent va voir le monde comme très chaotique et doit donc ne pas apprendre trop vite pour éviter d'avoir un comportement oscillant. Il doit cependant parvenir à extraire des tendances lui permettant d'agir au mieux par la suite.
- lorsque l'exploration devient moins importante, il doit avoir exploré une partie de son environnement et en avoir retenu un comportement intéressant pour pouvoir en tirer profit.

6.2.2.4 Résultats Comparés

Expérience :

Nous avons souhaité comparer les résultats obtenus avec notre apprentissage à ceux qu'il serait possibles d'obtenir par apprentissage sur un problème analogue modélisé par un DEC-POMDP.

Les interactions sont ainsi remplacées par des zones communes dans lesquelles les agents peuvent entreposer et prendre des seaux (comme l'illustre la figure 6.10).

Le problème des pompiers a ainsi été modélisé par un DEC-POMDP $< S, A_i, T, \Gamma_i, O, R >$ pour lequel :

- L'état global du système est caractérisé par les positions des agents, le contenu de leurs seaux et la présence éventuelle de seaux dans les zones d'entrepôt.
- Les actions possibles pour un agent consistent à se déplacer, remplir un seau, éteindre un feu [18], prendre ou poser un seau dans la zone d'entrepôt la plus proche
- La matrice de transition se décompose en matrices de transition individuelles excepté pour les poses et prises de seau. Des interactions indirectes apparaissent dans ces situations.
- Les perceptions d'un agent sont limitées à son état individuel et les seaux situés dans les états d'entrepôt proches de lui.

De plus, afin de pouvoir guider les agents, nous avons été obligés de lever une contrainte de localité : les agents peuvent désormais percevoir la récompense globale du système. L'apprentis-

passées est intégrée avec un facteur (1-α).

[18] Ces actions dépendent du type d'agent considéré.

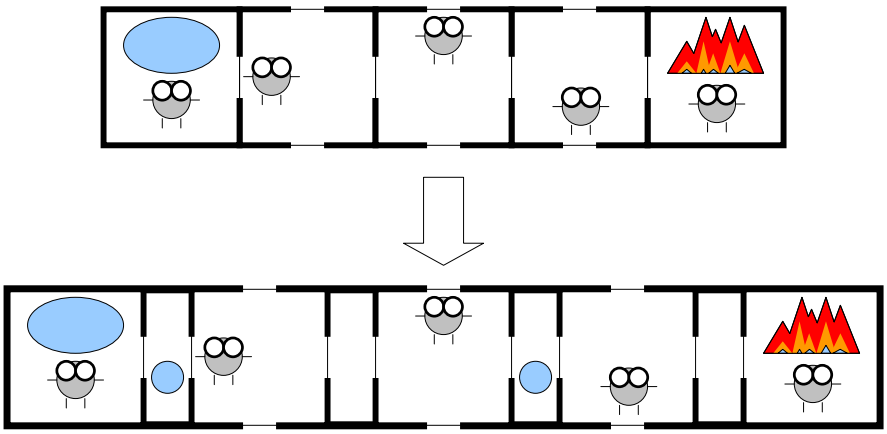


FIG. 6.10 – DEC-POMDP modélisant le problème de l'exemple 1

sage effectué sur ce système consiste pour chaque agent à effectuer un Q-learning décentralisé à partir de la récompense globale (avec pour paramètre d'apprentissage ϵ convergeant de 1 vers 0, pour valeur de α, 0.1 et pour valeur de γ, 0.9) pendant 50000 pas de temps.

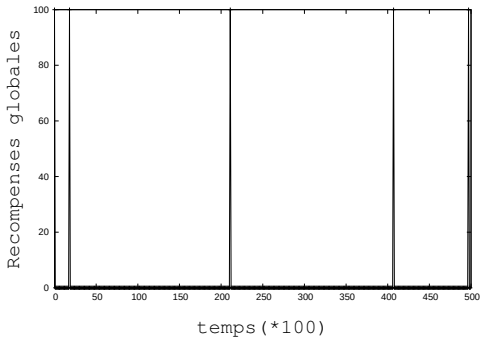


FIG. 6.11 – Récompenses reçues par exploitation au cours des 50000 pas de temps d'apprentissage dans le DEC-POMDP sans interaction directe

Observations :

Même pour ce cas très simple, on peut constater que les comportements des agents oscillent et qu'aucun comportement global n'émerge (cf fig 6.11 et 6.12). La figure 6.11 montre que les agents n'arrivent pas à éteindre les feux puisque les récompenses reçue par le système sont faibles. On peut constater en outre que même si les agents arrivent à adopter ponctuellement des comportements individuels permettant d'éteindre un feu, au cours de la suite de l'apprentissage ces comportements disparaissent.

Interprétation :

Les agents ne parviennent pas à construire des politiques individuelles utiles à cause du problème de *'crédit assignment'* : les récompenses sont attribuées aux agents lorsqu'un agent extincteur

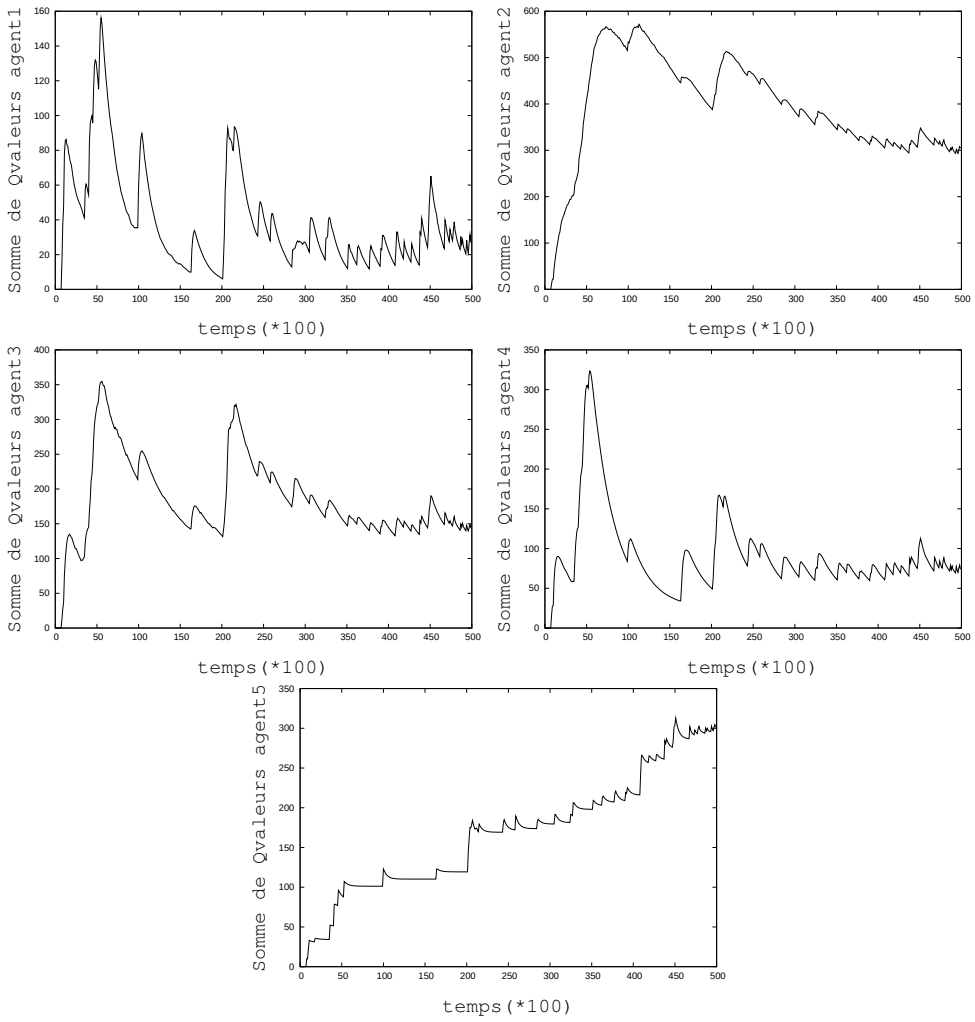


FIG. 6.12 – Évolution de la somme des Q-valeurs des agents dans le DEC-POMDP équivalent sans interaction

verse le contenu de son seau sur un feu, mais lorsqu'elles sont attribuées, les agents ne savent pas quel suite d'actions parmi celle qu'ils ont émises a pu permettre de générer cette récompense.

Conclusion :

Les interactions directes que nous avons introduites permettent donc de guider l'apprentissage des comportements individuels en explicitant les relations entre les agents et en effectuant automatiquement des transferts de récompense adéquats. Elles permettent de construire des systèmes permettant de résoudre des tâches que des apprentissages simples sur des DEC-POMDPs seuls ne permettent pas de résoudre.

6.2.3 Passage à l'échelle

6.2.3.1 Organisations concurrentes

Afin de voir quelle(s) organisation(s) peu(ven)t émerger de situations complexes, nous nous sommes intéressés à des problèmes dans lesquels plusieurs organisations permettant de générer des récompenses positives peuvent apparaître bien que toutes ne soient pas équivalentes. Ainsi, certains comportements collectifs permettent de générer plus de récompense globale que d'autres. De plus, dans certains cas, ces organisations ne peuvent pas apparaître simultanément puisqu'elles nécessitent des comportements différents pour un même agent.

Nous nous sommes ainsi intéressés à
- une situation pour laquelle il existe deux puits, l'un étant située plus près du feu que l'autre et pour laquelle les deux organisations potentielles nécessitent la participation d'agents communs (cf fig 6.13).
- une situation pour laquelle il existe deux incendies, l'un étant situé plus près d'un puit que l'autre et pour laquelle les deux organisations potentielles nécessitent la participation d'agents communs (cf fig 6.14).
- une situation pour laquelle il existe deux incendies situés aussi loin du puit mais pour laquelle un incendie rapporte des récompenses plus importantes que le second (cf fig 6.15).
- une situation pour laquelle il existe plusieurs puits et incendies (cf fig 6.16).

L'agent qui va être responsable de l'émergence d'une organisation plutôt qu'une autre est représenté sur ces schémas par un point d'interrogation.

Plusieurs incendies

Expérience :

Dans cette situation, il existe plusieurs sources de récompenses représentées par plusieurs pièces contenant des cases de feu. Deux organisations générant des récompenses sont possibles mais elle ne peuvent coexister car elles nécessitent toutes deux la participation d'un même agent.

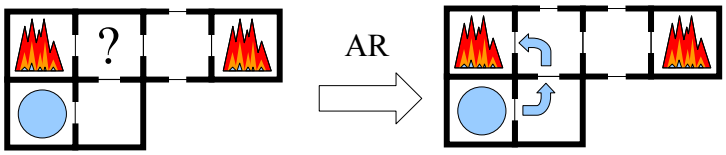


FIG. 6.13 – Problème constitué de deux incendies

Observations :

On peut constater l'apparition d'un comportement collectif consistant à amener le seau de l'agent ravitailleur vers l'agent extincteur qui en est le plus proche (cf figure 6.13).

Interprétation :

Le choix du comportement collectif repose sur le comportement de cet agent commun. Puisque les récompenses sociales sont décroissantes par rapport au nombre d'agents séparant l'agent du feu, de son point de vue, le feu le plus proche lui permet d'accéder à une récompense sociale

plus importante. C'est donc le comportement individuel consistant à donner son seau plein à l'agent le plus proche du feu qui va être sélectionné. Cette décision individuelle conditionne alors l'apparition d'organisation au sein du système.

De plus, au cours de l'exploration, plus la chaîne est importante en taille, plus elle est fragile puisqu'elle implique plus d'agents (aux comportements stochastiques). L'apprentissage a donc tendance à favoriser l'apparition des chaînes les plus courtes.

Conclusion :

Le comportement collectif correspondant est optimal puisque c'est celui qui permet d'amener le plus rapidement possible des seaux remplis au feu. Il est à noter cependant qu'en horizon infini, ces deux comportements ont des performances quasiment identiques [19] puisqu'une fois la chaîne établie, peu importe le nombre d'agents qui la composent, le flux de seau est constant.

Plusieurs puits

Observation :

Lorsque plusieurs puits sont présents dans le système, d'une manière similaire, la chaîne la plus courte émerge (cf figure 6.14).

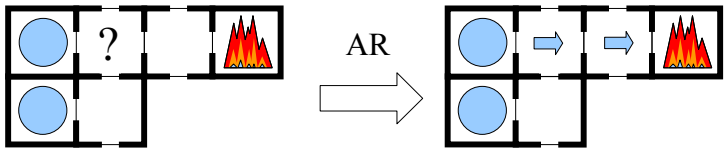


FIG. 6.14 – Problème constitué de deux puits

Interprétation :

Ce comportement collectif apparaît car si la chaîne plus longue est sélectionnée, un agent couloir supplémentaire va devoir recevoir des récompenses sociales. Cet agent supplémentaire doit lui aussi transmettre des récompenses sociales à l'agent ravitailleur. L'agent commun aux deux organisations potentielles va donc devoir lui transmettre des récompenses sociales plus importantes que s'il interagissait directement avec un agent ravitailleur. L'apprentissage va donc conduire l'agent commun aux deux organisations à se diriger directement vers l'agent ravitailleur puisque sa Q-valeur est plus importante dans ce cas.

L'apparition de ce comportement collectif est fortement renforcé par le fait que l'apprentissage a aussi tendance à favoriser les chaînes les plus courtes comme nous l'avons fait remarquer précédemment.

Plusieurs incendies distincts

[19] A un facteur ϵ prés

Expérience :

La situation décrite par la figure 6.15 consiste à mettre en présence plusieurs incendies de nature différente : un incendie génère des récompenses de +100 lorsqu'on y verse un seau rempli d'eau, l'autre une récompense de +200.

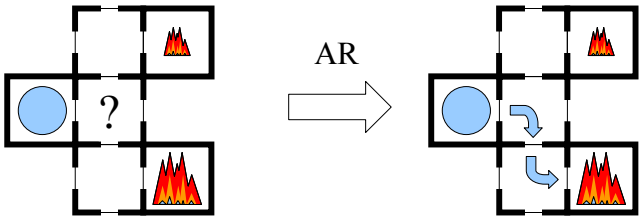


FIG. 6.15 – Problème constitué de deux incendies distincts

Observations :

Pour cette expérience, c'est effectivement la chaîne permettant d'amener des seaux au feu générant le plus de récompenses qui va apparaître (cf figure 6.15).

Interprétation :

Comme les récompenses sociales transmises dépendent directement des récompenses effectives reçues par les agents extincteurs, l'agent couloir sera plus incité à amener le seau vers l'incendie rapportant les récompenses les plus importantes. A l'exécution, c'est effectivement ce comportement collectif qui apparaît (et qui s'avère en outre être le comportement optimal).

Plusieurs sources, plusieurs incendies

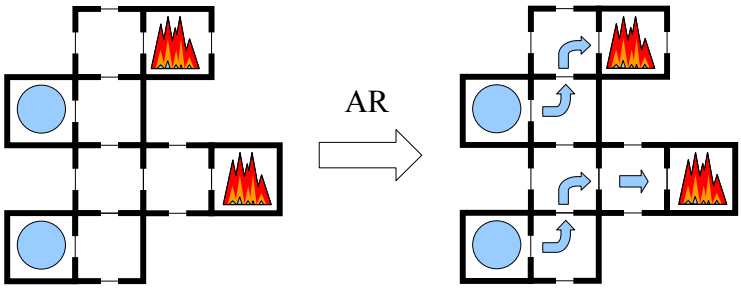


FIG. 6.16 – Problème constitué de deux sources et deux incendies

Observation :

Lorsque plusieurs organisations peuvent apparaître sans conflit, comme illustré par la figure 6.16,

le comportement collectif correspond à l'apparition de deux chaînes distinctes chacune amenant des seaux remplis vers un incendie.

Bilan Dans ces différentes situations, le fait que les transferts de récompenses sociales soient fonctions des récompenses finales et s'amenuisent au fur et à mesure que le nombre d'interactions augmente conduit à l'apparition spontanée d'organisations correspondant au comportement collectif optimal.

6.2.3.2 Coordination d'organisations

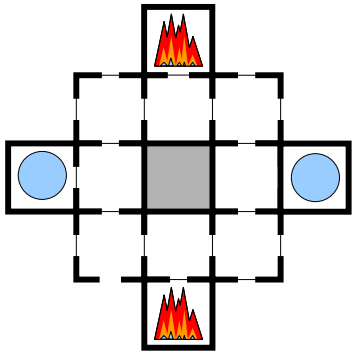


FIG. 6.17 – Expérience symétrique

Expérience :

Afin de voir comment le système pouvait s'organiser, nous nous sommes intéressés à une expérience pour laquelle la disposition des pièces est symétrique comme le montre la figure 6.17. Dans cette expérience plusieurs organisations utiles à la résolution de la tâche peuvent apparaître (celles si sont représentées sur la figure 6.18). Ces organisations ont des performances identiques vis à vis de la tâche puisqu'elles correspondent à des comportements collectifs générant les mêmes récompenses globales.

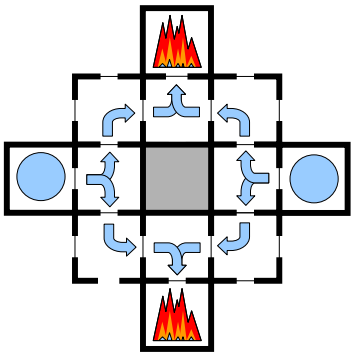


FIG. 6.18 – Organisations possibles

Le risque consiste à ce que les agents cherchent à éteindre le même feu et se gênent mutuellement.

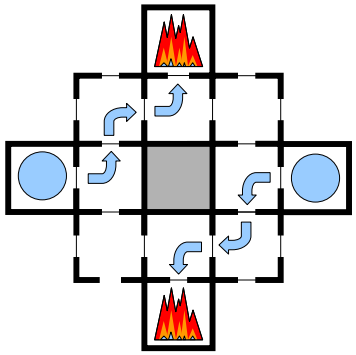


FIG. 6.19 – Brisure de symétrie

Observation :

Or à l'exécution du système, on observe une brisure de symétrie : plusieurs chaînes apparaissent, une permettant d'éteindre le feu au sud, l'autre permettant d'éteindre le feu au nord (cf fig 6.19).

Interprétation :

L'apparition de cette brisure de symétrie vient du fait que les agents au cours de leur apprentissage prennent en compte le comportement des autres agents de manière implicite. Ainsi, au cours de son apprentissage, un des agent-couloir proche d'un agent extincteur va émettre une légère préférence quant à l'agent à qui il va emprunter un seau plein. Cette préférence va avoir deux conséquences : tout d'abord, l'agent à qui il emprunte le seau va recevoir des récompenses et va être incité à reproduire la situation propice à l'interaction directe. Ensuite, l'agent du coté opposé va avoir tendance à apprendre qu'amener un seau ne constitue pas une action intéressante pour le système puisque l'autre agent ne sera pas amené à interagir avec lui. Ces légères différences initiales vont être amplifiées par apprentissage et se propager dans le système pour faire apparaître une chaîne d'un seul des deux cotés.

Inversement, une fois qu'une telle chaîne commence à apparaître, elle va avoir tendance à concentrer l'attention de certains agents, les récompenses reçues par l'autre agent extincteur vont se propager vers l'agent ravitailleur non impliqué dans la première chaîne.

Conclusion :

Ces expériences montrent ainsi non seulement que ces apprentissages permettent l'apparition d'organisation liées à la tâche, mais qu'en plus, ces organisations s'adaptent elles aussi aux autres organisations présentes et aux contraintes du système. Ces apprentissages permettent des adaptation à deux niveaux : au niveau des agents pour la génération d'organisation et de comportement collectif complexe et au niveau des organisations pour l'agencement des organisations les unes par rapport aux autres.

6.2.3.3 Augmentation du nombre d'agents

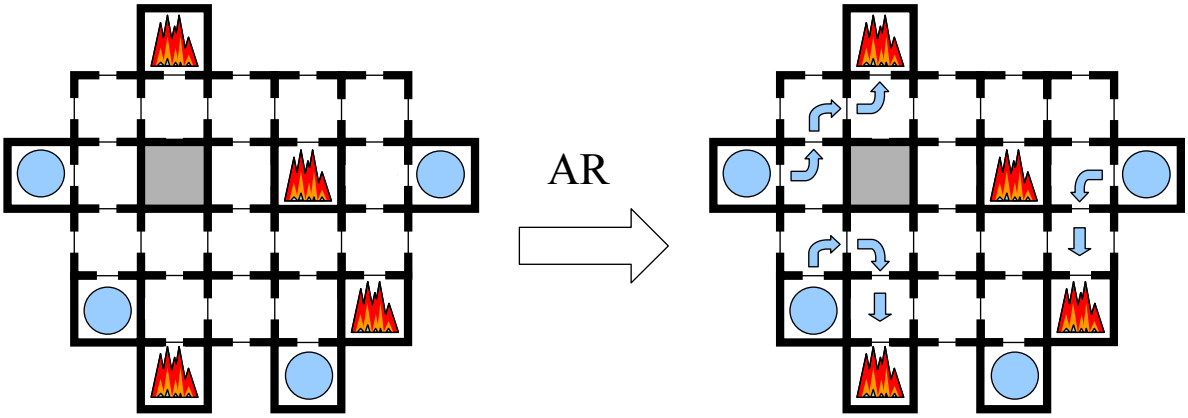


FIG. 6.20 – Système avec 24 agents

Expérience :

L'approche proposée du fait d'apprentissages décentralisés permet d'appréhender des problèmes avec un nombre d'agents élevé sans explosion combinatoire puisque chaque politique individuelle est construite à partir de Q-valeurs individuelles fondées sur des perceptions partielles.

Ainsi, des systèmes avec un nombre important d'agents (24 dans ce cas) sont facilement concevables (cf figure 6.20).

Observations :

Dans cet exemple, les résultats qualitatifs font apparaître la présence de plusieurs chaînes indépendantes. Les apprentissages permettent de mettre en interaction plusieurs agents uniquement en fonction de la tâche à résoudre et de générer simultanément plusieurs organisations si elles sont utiles. Les apprentissages effectués sur cet exemple permettent aux agents d'éteindre de manière optimale trois feux sur les quatre (cf figure 6.21).

Interprétation :

L'absence d'apparition d'une quatrième chaîne s'explique par l'aspect synchronisé des stratégies d'exploration. Au cours de la phase d'exploration, les comportements des agents sont stochastiques. Les chaînes ne sont pas alors totalement établies puisque le comportement des agents pouvant participer à une chaîne ne sont pas déterministes. Ainsi, d'autres agents proches vont dans certains cas transmettre un seau mais uniquement parce qu'un agent de la chaîne potentielle a exploré son environnement et cette exploration le conduit à interagir avec ces agents. Par contre, dés que la phase d'exploration s'achève, la chaîne est effectivement établie, les agents proches ne peuvent plus participer à cette chaîne mais ne cherchent plus à explorer leur environ-

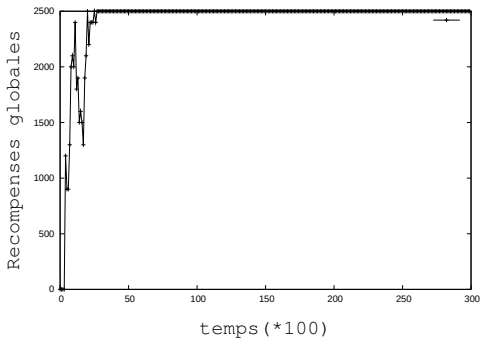


FIG. 6.21 – Récompense reçues

nement.

Conclusion :

Le fait que l'exploration soit déterminée par un critère global conduit à des problèmes insolubles :

- soit tous les agents explorent leur environnement, mais les chaînes ne sont alors pas établies et certains comportements apparaissent du fait de l'aspect stochastique des comportements.
- soit tous les agents exploitent leurs politiques, les chaînes s'établissent et bloquent certains agents mais ceux-ci n'explorent plus leur environnement pour chercher d'autres solutions.

Perspective :

Une piste qui nous semble intéressante à creuser dans des travaux futurs serait de doter les agents de coefficients d'exploration et d'apprentissage individuels et de modifier ce coefficient en fonction des relations qu'ils peuvent entretenir avec les autres agents.

6.2.4 Restructuration

6.2.4.1 Ajout d'un agent

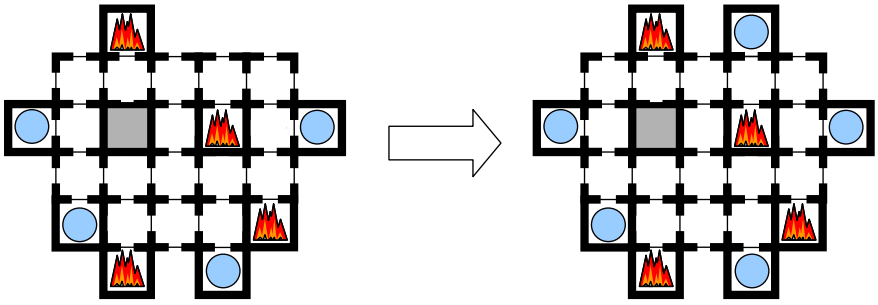


FIG. 6.22 – Ajout d'un agent en cours d'execution

Expérience :

Nous avons enfin cherché à analyser les capacités de réadaptation du système à l'ajout et au retrait d'agents en cours d'exécution (cf figure 6.22). Pour cela, nous avons maintenu un facteur d'exploration ϵ non nul dans le système.

Observations :

La figure 6.23 présente les performances du système lorsque ϵ est maintenu à 0.1. Afin de rendre compte de l'influence du coefficient ϵ sur le performances, cette courbe présente les récompenses globales reçues par le système lorsqu'au cours de la phase d'évaluation le système continue à explorer. L'exploration constante se traduit par une diminution des performances globales importantes [20].

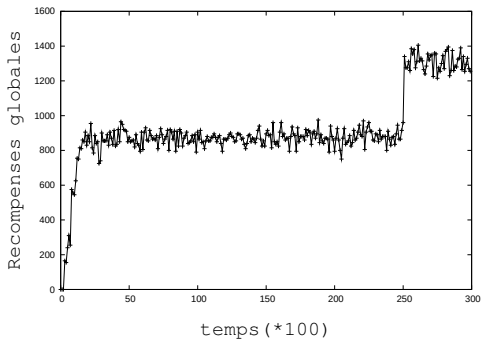


FIG. 6.23 – Récompense reçues avec exploration et adaptation

Cependant, maintenir une valeur de ϵ non nulle permet de prendre en compte l'ajout de nouveaux agents. Au pas de temps 25000, un agent ravitailleur est ajouté dans le système (cf figure 6.22 et 6.23). Les agents continuant à explorer leur environnement parviennent à prendre en compte la présence d'un nouvel agent et font converger le système vers un nouvel équilibre en temps réel.

Conclusion :

Ces expériences montrent ainsi qu'il est possible d'observer des re-configurations automatiques du système avec un apprentissage constant et d'envisager des utilisations de ce processus dans des systèmes ouverts pour lesquels des agents apparaissent et disparaissent à l'exécution du système.

6.2.4.2 Restructuration globale du système

Expérience :

Afin d'analyser des phénomènes de restructuration plus complexes, nous avons conduit des expériences sur l'exemple illustré figure 6.24. Cette expérience consiste à ajouter au cours de l'exécution un nouvel agent couloir qui permet aux agents d'atteindre un nouveau feu à éteindre et d'accéder à de nouvelles récompenses.

[20] Cette chute des performances de 75% s'explique par le fait qu'une valeur de 0.1 pour le coefficient ϵ est très importante. Chaque agent à 10% de choisir une action aléatoire à chaque pas de temps, ce qui a tendance à fortement briser les organisations qui sont apprises.

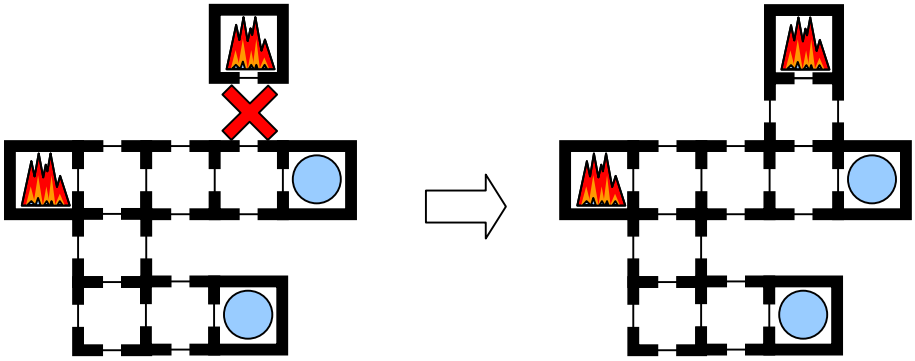


FIG. 6.24 – Expérience de re-structuration

Observations :

Dans cette expérience, on opère un premier apprentissage les 20000 premiers pas de temps (α vaut 0.02, et ϵ décroît linéairement de 1 vers 0). A la suite de cet apprentissage, une organisation émerge. Elle est caractérisée par la présence d'une chaîne qui permet d'éteindre le feu comme cela est illustré figure 6.25.

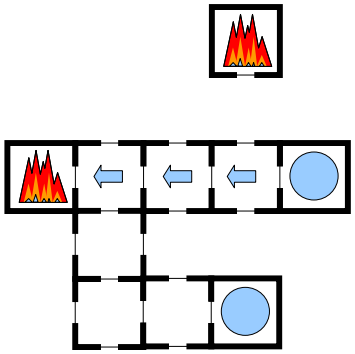


FIG. 6.25 – Première organisation

Observations :

On ajoute à partir du pas de temps 20000 un nouvel agent couloir dans le système comme cela est illustré figure 6.24. Cet ajout fait que désormais les agents ont plutôt intérêt à éteindre le feu le plus proche d'eux. Cependant, il reste un effet mémoire du aux transferts de récompenses : les Q-valeurs des agents font que la chaîne anciennement créée a tendance à se reconstruire. De plus, le nouvel agent couloir est incité à redonner le seau à chaque fois qu'il se le procure car, d'une part, les autres agents ont appris qu'ils avaient intérêt à disposer d'un seau pour pouvoir participer à l'organisation construite précédemment et car, d'autre part, ce nouvel agent n'a pas encore eu l'occasion d'explorer son environnement.

Néanmoins, à condition que le coefficient d'exploration reste non nul (ϵ vaut 0.1), une restruc-

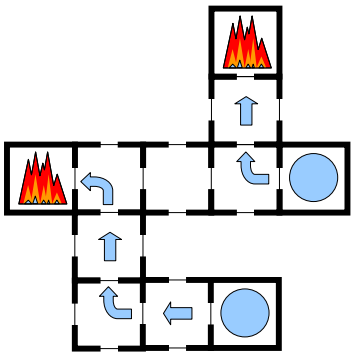


FIG. 6.26 – Après restructuration du système

turation à l'exécution du système est encore possible. Ainsi, jusqu'au pas de temps 100000, les coefficient ϵ est maintenu à 0.1, puis il prend pour valeur 0 afin de pouvoir exploiter et stabiliser la nouvelle organisation constituée de deux chaînes.

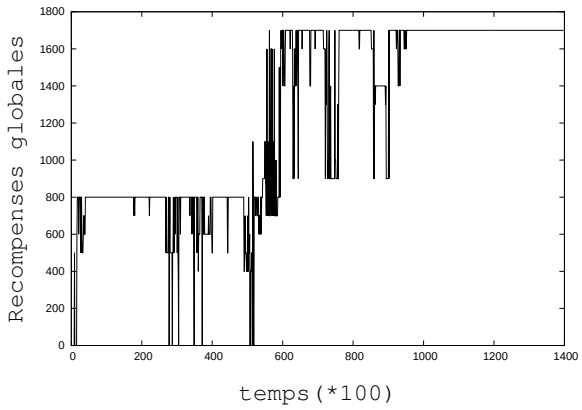


FIG. 6.27 – Récompenses globales reçues par le système au cours de l'expérience de restructuration

Tous les comportements des agents sont modifiés en conséquence de manière automatique pour prendre en compte la modification des structures relationnelles sur lesquelles peuvent s'appuyer les interactions comme cela est illustré figure 6.26. La figure 6.27 montre l'évolution des récompenses obtenues par exploitation des comportements au cours de l'apprentissage. Elle met en évidence la présence d'un temps de latence nécessaire pour réapprendre une nouvelle organisation (qui apparaît à partir du pas de temps 60000).

Conclusion :

Ces expériences montrent la capacité de restructuration des systèmes fondés sur notre apprentissage.

6.2.5 Limites de l'apprentissage

L'objectif de cette partie est de montrer au travers d'expériences la manière dont des organisations pouvaient apparaître dans des situations plus complexes. Elle nous permettra en outre de mettre en avant des pistes possibles pour la poursuite de travaux dans ce sens.

6.2.5.1 Récompense positive au contact de l'eau

Expérience :

Cette expérience consiste à introduire des récompenses positives individuelles afin d'évaluer la manière dont sont gérés les conflits entre des comportements collectifs et des comportements individuels. Le système sur lequel ces expériences ont été menées est constitué de trois agents : un agent extincteur, un agent ravitailleur et un agent couloir. Désormais, l'agent couloir reçoit une récompense positive lorsqu'il garde un seau plein qui lui a été confié.

En fonction de cette récompense positive, les comportements collectifs optimaux sont différents :

- Si la récompense reçue par l'agent est supérieure à 50, l'agent en conservant son seau parvient à apporter une récompense de 50 à chaque pas de temps. Cette récompense s'avère plus importante que la récompense de 100 reçue pour éteindre le feu puisque l'agent extincteur ne peut recevoir cette récompense qu'en deux pas de temps, le temps nécessaire pour récupérer un seau. Dans de telles circonstances, le comportement optimal consiste à ne pas générer de chaîne.
- Si la récompense reçue par l'agent est inférieure à 50, le comportement collectif optimal consiste à générer une chaîne.

Observations :

Les expériences qui ont été conduites montrent cependant que, dès que la récompense positive attribuée à l'agent couloir dépasse 25, la chaîne se brise et l'agent préfère garder son seau.

Interprétation :

Il s'agit d'un problème lié à l'exploration jointe des agents :

- lorsque les agents explorent leur environnement, les politiques d'actions et de déclenchement de l'agent extincteur deviennent stochastiques. L'agent couloir a alors peu de chances de transmettre un seau à l'agent extincteur et dans le cas où un échange a lieu, les récompenses sociales seront faibles car l'agent extincteur évalue encore mal le gain qu'il peut avoir à posséder un seau. l'agent couloir préfère donc recevoir une récompense individuelle qui ne dépend que de lui et qu'il est certain de contrôler plutôt que d'effectuer des interactions avec les autres agents.
- lorsque les agents exploitent leurs politiques, ils ne peuvent pas découvrir de nouveaux comportements collectifs. Ainsi, lorsque, après apprentissage, le comportement de l'agent extincteur est fixé, l'agent couloir préfère garder son seau et ne cherche plus à transmettre celui-ci à l'agent extincteur pour découvrir un meilleur comportement collectif.

Lorsque les différences de récompenses sont très importantes (récompense positive inférieure à 25), cette différence compense les incertitudes que peut avoir un agent sur les autres agents.

Lorsque ce n'est pas le cas, le système converge vers des minimas locaux.

Piste future

Une solution envisageable par la suite serait de désynchroniser certains apprentissages et de conditionner le facteur α d'apprentissage d'un agent par les facteurs d'exploration ϵ des autres agents.

6.2.5.2 Coût de déplacement

Expérience :

Toujours pour évaluer comment le système se comportait face à des conflits entre intérêts individuels et collectifs, nous avons ajouté des coûts de déplacement. Ces expériences consistent à attribuer des récompenses négatives aux agents couloirs lorsqu'ils se déplacent. Pour une chaîne constituée de trois agents, en fonction de la récompense attribuée à l'agent couloir, les comportements optimaux sont différents.

L'agent couloir doit faire un aller retour pour amener un seau à l'agent extincteur, amener un seau génère donc le double du coût de déplacement. Ainsi,

- si la récompense négative attribuée à l'agent est inférieure en valeur absolue à 50, le comportement collectif optimal consiste à apporter le seau à l'agent extincteur puisque cette action génère une récompense globale positive.
- si la récompense négative attribuée est supérieure en valeur absolue à 50, le comportement collectif consiste à ne pas générer de chaîne qui ne serait pas rentable.

Observations :

On observe effectivement l'apparition d'une chaîne lorsque la récompense négative est faible en valeur absolue, mais dés que cette récompense devient plus importante, la chaîne se brise (à partir de récompenses de -25).

Interprétations :

Ce comportement provient en partie du fait que les coûts de déplacements ne sont pas directement pris en compte dans l'interaction. En effet, les récompenses sociales sont calculées par rapport aux gains et aux pertes de l'agent dûs à l'exécution du résultat d'interaction choisi collectivement mais elles ne prennent pas en compte les coûts (ou gains) nécessaires pour la production d'une situation propice à l'interaction.

Les problèmes liés à l'exploration jointe évoqués précédemment s'ajoutent encore à ces considérations : tant que les agents explorent, l'agent couloir ne peut pas être sûr des réactions des autres et apprend à ne pas se déplacer. Quand les agents exploitent leur comportement, l'agent couloir ne cherche plus à se déplacer pour permettre l'apparition d'une chaîne dans le système.

Piste future :

Une solution serait de modifier l'heuristique de partage pour prendre en compte ces coûts. Des travaux ont été entrepris dans cette direction.

6.2.5.3 Influence de la distance

Enfin, au cours des interactions, le fait de répartir de manière équitable les récompenses conduit à plusieurs problèmes.

Le premier se produit lorsque deux feux à éteindre se situent à des distances différentes. Les récompenses sont dégradées de manière exponentielle au fur et à mesure des échanges. Ainsi, il peut arriver qu'un feu soit ignoré alors qu'il rapporte beaucoup plus de récompenses qu'un autre mais que cette récompense soit masquée du fait de la distance. Dans ces situations, des chaînes sous-optimales bloquant l'apparition de meilleures chaînes apparaissent dans le système.

Le second se produit lorsque un agent situé à une distance importante d'un agent extincteur est sollicité par d'autres récompenses individuelles comme présenté dans les parties précédentes. Les récompenses qui lui ont été transmises ont été fortement dégradées et empêchent l'agent de considérer un échange parce qu'il l'estime non profitable pour le système. Ce problème est particulièrement complexe puisque lui fournir des récompenses adéquates peut consister à remettre en cause une partie des récompenses déjà attribuées précédemment aux autres agents.

Piste future Une approche possible pour résoudre ces problèmes serait d'envisager d'autres heuristiques en particulier un apprentissage des récompenses à échanger (caractérisé par le coefficient λ). Par exemple lorsqu'un agent extincteur ne parvient pas à accéder à un seau, il peut augmenter les récompenses sociales qu'il a l'intention de donner pour inciter les autres agents à venir l'aider.

6.2.6 Synthèse des résultats

Les expérimentations que l'on a pu conduire montrent qu'il est possible de générer des comportements collectifs automatiquement sans qu'à aucun moment les agents ne disposent d'une vue globale du monde : les échanges de valeur sous forme de récompenses sociales couplés à des apprentissages simultanés et progressifs des agents parviennent à construire des comportements collectifs complexes.

Cette approche permet effectivement de construire le comportement d'agents réactifs adaptatifs, dans un système coopératif pour résoudre des problèmes collectifs de manière automatique et décentralisée. Le processus conduit en outre

- à des comportements stables
- à des processus de construction robustes
- à des processus de construction fondés sur des processus de construction mono-agents sans explosion combinatoire
- à des comportements adaptatifs et capables de s'adapter à l'ajout et au retrait d'agents dans le système.

Ces comportements s'expriment par l'apparition d'organisations caractérisées par l'émission d'actions, le déclenchement d'interaction directe et la résolution itérée de certaines interactions entre agents. Ces organisations apparaissent de manière spontanée dans le système uniquement à partir de la formalisation du problème à résoudre et possèdent des propriétés adaptatives (ajout d'agents, conflits entre organisation, etc...).

Ces résultats ont ainsi permis de mettre en évidence la force de la notion d'interaction en comparant les résultats les plus simples à une approche similaire sans interaction.

6.3 Positionnement

Maintenant que nous avons décrit et analysé les résultats qu'il est possible d'obtenir, il est possible de positionner notre approche mêlant actions et interactions aux autres travaux que nous avons présenté précédemment.

6.3.1 Satisfaction altruisme

Comme nous l'avons déjà mentionné, le modèle satisfaction altruisme [Sim01] est très proche de nos considérations (cf partie 2.2.4.3). Ce modèle est fondé sur
- des interactions directes
- des interactions indirectes entre les agents

Les interactions directes sont basées sur la notion d'insatisfaction : dés qu'un agent est dans une situation bloquante, il émet un signal d'insatisfaction croissant qui modifie le comportement des agents voisins.

Cependant le modèle satisfaction-altruisme utilise des règles données à priori permettant d'intégrer les signaux émis par les agents au niveau des comportements individuels. En cela, le modèle est plus proche du modèle Hamelin qui décrit des lois de fonctionnement et d'adaptation que du formalisme Interac-DEC-POMDP qui cherche à construire automatiquement les comportements des agents à partir de la tâche à résoudre. Ainsi, même si le modèle satisfaction altruisme et des modèles similaires sont utilisables dans de nombreux problèmes comme les problèmes d'optimisation combinatoire (cf [LSC03] qui présente des techniques similaires pour résoudre des problèmes d'affectation de plages horaires dans des emplois du temps), les lois d'adaptation à mettre en œuvre doivent être déterminées par le concepteur du système.

Notre approche se voulait plus générique en ce sens. Nous effectuons un apprentissage automatique des interactions ce qui permet de produire et d'adapter les comportements des agents à partir d'un problème donné a priori et représenté sous la forme d'une fonction de récompense distribuée.

6.3.2 Weakly coupled MDP

Dans [MHK+98], le problème traité est un problème d'allocation de ressources : plusieurs tâches indépendantes doivent être effectuées mais les ressources globales disponibles sont limitées(cf 3.4.3.9).

L'approche consiste
- à évaluer dans une première phase les valeurs associées aux tâches en fonction des ressources restantes allouées à la tâche.
- à repartir une partie des ressources à partir d'une heuristique fonction des valeurs individuelles.

A l'exécution, une répartition est faite ressource par ressource selon les utilités marginales associées aux différentes tâches. L'article [MHK+98] repose sur les mêmes principe que ceux qui nous ont guidés :

- la capacité à décomposer un problème global en sous-problèmes locaux consistant à exécuter une tâche
- la construction des politiques optimales et leur valeurs pour ces sous problèmes
- le fait que les fonctions de valeur calculées peuvent guider les interactions entre les tâches
- l'utilisation à l'exécution d'heuristiques pour effectuer les actions collectives consistant en une répartition des ressources.

Il est possible de représenter le problème d'attribution de ressources de [MHK+98] dans le formalisme Interac-DEC-POMDP. Il suffit pour cela d'introduire une interaction impliquant tous les agents du système et consistant à répartir les ressources entre les agents. Les résultats possibles pour une telle interaction sont constitués par les distributions particulières de ces ressources.

Certaines heuristiques utilisées pour distribuer les ressources correspondent aux heuristiques que nous avons mises en place. La principale différence avec notre approche réside dans le fait que l'interaction n'est pas formalisée explicitement. Notre approche en la définissant permet de nombreuses variations de cette approche :

- elle permet des interactions impliquant des agents en nombre moins important
- elle permet de limiter les communications à des communications locales
- elle permet une restructuration des interactions

6.3.3 Algorithme de Co-evolution

Les travaux de Chadés (cf [Cha02]) s'intéressent aussi à la construction de systèmes multi-agents. Ces travaux utilisent des techniques de planification pour construire les comportements des agents. Afin de contourner le problème de non-stationnarité des environnements subjectifs, l'approche proposée dans [Cha02] consiste à construire itérativement l'ensemble des politiques des agents (cf partie 3.4.3.7)

Une telle approche ne peut fonctionner sur le problème des pompiers représenté dans le formalisme DEC-POMDP : il faut dans ce cas considérer l'ensemble des agents pour effectuer une planification correcte.

Considérons un agent A situé prés d'un puit et un agent B situé prés d'un feu. Si on essaie de construire les politiques des agents en fixant successivement la politique de l'un d'entre eux, le comportement collectif consistant à éteindre le feu ne peut pas apparaître. En effet, pour que l'agent A puisse apprendre à aller chercher un seau à un agent B, il est nécessaire que l'agent B ait déjà adopté le comportement consistant à éteindre le feu. Inversement, pour que l'agent B puisse apprendre un comportement consistant à aller chercher un seau et à éteindre un feu, il est nécessaire que l'agent A ait adopté un comportement consistant à lui apporter un seau.

L'interaction en considérant des actions jointes permet de sortir de certains sous-optimaux caractérisés par des équilibres de Nash pour lesquels les agents doivent modifier simultanément leurs politiques. De plus, ces sous-optimaux n'ont souvent pas lieu d'être puisque le concepteur du système peut déjà avoir une réponse à ces problèmes de coordination dans les interactions qu'il envisage d'implémenter et qu'il ne pouvait faire directement dans le formalisme DEC-POMDP.

6.3.4 Guestrin

Les travaux de Guestrin consistent à construire des systèmes multi-agents en utilisant une représentation factorisée d'un MMDP.

Il représente la fonction de valeur globale de manière factorisée et utilise des techniques d'élimination de variables pour choisir l'action optimale. Ces techniques d'élimination de variables utilisent des communications locales entre les agents à chaque pas de temps et se fondent sur un réseau fixe de relations entre agents. (cf partie 3.4.3.6)

Nos travaux utilisent le même genre d'approche consistant à structurer le système en utilisant la notion d'interaction et en s'intéressant à des maximisations locales.

Cependant, les contraintes que respecte notre système sont plus importantes :

- Nous avons supposé que les agents ne peuvent échanger leur Q-valeurs que lorsqu'ils sont proches : il n'est pas possible de propager à tout instant les fonctions de valeurs de tous les agents pour reconstituer la fonction de valeur globale.
- Le nombre d'agents peut évoluer au cours du temps à condition d'ajouter les interactions correspondantes à l'exécution.

Ces contraintes font qu'une approche comme celle envisagée dans Guestrin n'est plus possible puisque les agents n'ont pas la possibilité de communiquer tous ensemble. Notre approche permet néanmoins de répondre à ces problèmes en intégrant au niveau d'un agent une partie des fonctions de valeurs des autres agents transmises sous la forme de récompense sociale au cours des interactions.

6.4 Bilan et Perspectives

6.4.1 Bilan du chapitre

Dans ce chapitre, nous nous sommes intéressés à la construction de comportements collectifs à partir de processus de construction de comportement individuel et de la formalisation de l'interaction directe effectuée dans le chapitre précédente. Nous nous sommes concentrés sur une sous-classe de l'Interac-DEC-POMDP dans laquelle l'interaction est centrale afin de voir de quelle manière il est possible de construire les interactions.

Nous avons pour cela proposé un algorithme permettant de construire les politiques d'actions, de déclenchement et d'interaction des agents à partir de signaux de récompenses perçus localement. Cet algorithme original se fonde sur plusieurs propositions jointes :

- nous nous sommes basés sur des techniques d'apprentissage individuel permettant à un agent de construire et d'adapter sa fonction de comportement à partir des interactions qu'il entretient avec son environnement
- nous avons construit les politique de résolution d'interaction directe en utilisant les Q-valeurs construites au cours des apprentissages individuels afin de construire ces politiques à moindre coût tout en permettant de prendre des décisions collectives utiles pour les agents impliqués dans l'interaction
- nous avons introduit des récompenses sociales au cours des interactions pour inciter des agents à reproduire des situations propices aux interactions utiles. En faisant ces transferts, la fonction de performance globale du système est répartie entre les agents et permet de répondre au problème du 'credit assignement'. Certains agents sont alors guidés par la

tâche à résoudre alors qu'ils n'ont aucun moyen d'accéder directement à des récompenses. La force de l'interaction directe dans ce cadre est qu'elle a permis en structurant le système d'identifier les agents en interaction et d'effecteur des échanges entre eux.

Notre approche n'a pas cherché l'optimalité mais elle avait pour objectif de montrer qu'il est possible de construire de manière automatique à partir d'heuristiques relativement simples des comportements collectifs qui n'étaient pas atteignables sans la notion d'interaction à partir d'un niveau de simplicité identique.

Cet algorithme parvient à construire automatiquement et de manière entièrement distribuée des comportements collectifs malgré toutes les contraintes que nous nous sommes fixées, c'est à dire pour des systèmes caractérisés par des communications locales limitées et par des perceptions partielles des agents concernant l'état du système et de la tâche à résoudre.

Nous avons prouvé la force de l'interaction. Nous avons montré qu'il est possible de construire des systèmes collectifs de manière entièrement décentralisée en se reposant sur l'interaction directe et sa formalisation qui offre de nombreuses nouvelles possibilités aux agents. De plus ces systèmes sont caractérisés par l'apparition spontanée d'organisation permettant à la collectivité de résoudre le problème posé.

Cette approche originale est ainsi à envisager comme une première tentative pour
- intégrer des prises de décision liées à la présence de plusieurs agents dans le système tout en respectant **des contraintes de localité**
- de guider **la construction décentralisée** des comportements individuels pour générer de manière des organisations liées à la tâche
- proposer des mécanismes d'apprentissage collectif tirant parti **à moindre coût** de mécanismes d'apprentissage individuels

6.4.2 Perspectives

Au cours de nos expérimentations, nous avons mis en évidence un certain nombre de points à développer ultérieurement.

6.4.2.1 Nouvelles heuristiques

L'introduction de récompenses individuelles dans le système conduit les comportements globaux vers des sous-optimaux : ainsi, il peut arriver qu'aucune chaîne d'agents n'émerge alors qu'elle pourrait permettre d'obtenir une récompense globale positive (cf partie 6.2.5).

D'autre heuristiques comme celle des récompenses adaptées sont envisagées pour répondre à ce problème. Cette heuristique consiste à intégrer les récompenses données par l'environnement au pas de temps précédent pour évaluer les récompenses sociales à transmettre au cours d'une interaction. Ce type d'heuristique permet de répondre au problème de l'introduction de coût dans les déplacements mais ne répond pas au problème dû à la synchronisation des apprentissages.

Un apprentissage des partages des récompenses sociales à l'exécution peut aussi être envisagé pour permettre d'adapter les récompenses transmises si elles ne sont pas assez importantes. Ce partage de récompense pourrait alors se faire à partir d'un facteur de répartition adaptatif qui

caractérise la partie du gain reçu par un agent.

Une autre approche que nous envisageons par la suite serait de trouver des processus de construction à partir desquels on pourrait prouver certaines propriétés sur les Q-valeurs des agents qui peuvent être construites. Ce problème reste difficile puisque les Q-valeurs associées aux agents n'ont pas un sens précis et que les échanges d'information sont conditionnés par leur comportement (contrairement aux approches proposées par Guestrin [Gue03]et Szer [Sze04]).

6.4.2.2 Réduction au niveau individuel

Nos travaux sont proches des travaux effectués sur les MDP faiblement couplés. Nous avons montré qu'il est possible de construire l'ensemble des comportements individuels potentiellement optimaux. Mais nous avons par contre négligé les problèmes de synchronisation pouvant apparaître dans le système.

Par exemple, en fonction des pièces dans lesquelles les agents se déplacent, traverser un couloir peut prendre plus de pas de temps que traverser une autre pièce. Ceci est d'autant plus vrai que l'exécution d'une politique peut être stochastique. Ainsi, lorsque des agents décident de se rejoindre, ils n'arrivent pas forcément au même instant dans les états permettant d'interagir.

Dés lors, une idée consisterait à rajouter des temps d'attente pour synchroniser les politiques
- soit de manière centralisée, en synchronisant toutes les politiques. C'est l'hypothèse que nous avons fait de manière implicite dans le problème des pompiers : atteindre un état d'interface prenait exactement un pas de temps.
- soit de manière distribuée. Le problème dans ce cas consiste pour un agent à savoir si l'autre agent a effectivement l'intention d'atteindre l'état d'interface pour exécuter une interaction (auquel cas il est utile d'attendre) ou s'il a effectué une politique le conduisant vers d'autres états (auquel cas l'attente est inutile). Or, comme l'exécution d'une politique prend un certain temps, l'agent doit pouvoir estimer un délai d'attente à partir duquel il considérera que l'agent n'atteindra pas cet état d'interaction.

Il est possible d'envisager ces problèmes dans le futur en utilisant plusieurs niveaux de représentation au sein des agents :
- un niveau proche du problème des pompiers pour lequel l'exécution d'une politique individuelle prend un pas de temps, ce niveau de représentation permettra à un agent de savoir s'il a intérêt à attendre
- un niveau modélisant les temps d'attentes espérés pour pouvoir savoir quand abandonner et considérer d'autres actions.

Afin de prendre en considération les problèmes d'optimisation qui se posent dans cette sous-classe des Interac-DEC-POMDP, nous envisageons d'utiliser un autre formalisme qui intègre explicitement la notion de temps : les SMDP habituellement utilisés dans les approches hiérarchiques pour les mêmes raisons (comme dans [WM99]).

Chapitre 7

Conclusion

7.1 Résumé du travail présenté

7.1.1 Objectifs fixés initialement

Les systèmes distribués prennent une place de plus en plus importante en informatique, que ce soit dans les réseaux, la robotique collective, ... Il s'agit de systèmes ouverts pour lesquels les composants peuvent s'ajouter et se retirer à l'exécution. De plus, chaque composant ne dispose que d'une vue partielle du système et doit néanmoins prendre une décision.

Ces systèmes constitués de composants en interaction posent la problématique de l'intelligence collective. Il s'agit de savoir dans quelle mesure et comment il est possible de construire des systèmes qui répondent à un problème posé globalement à partir de composants simples dotés de perceptions partielles et en interaction.

Pour répondre à cette question, nous avons cherché à construire des systèmes multi-agents **réactifs** (car nous souhaitons évaluer ce qu'il est possible d'obtenir en combinant des comportements d'agents simples) **coopératifs** (car nous souhaitons répondre à un problème posé à la collectivité et tirer parti de la multiplicité des prises de décision) **rationnels** (car nous souhaitons répondre à un problème posé sous la forme d'un problème global d'optimisation), **adaptatifs** (car ces systèmes ne peuvent accéder à une vision globale et doivent donc faire face à des situations imprévues et changeantes) de manière **décentralisée** (puisque les agents ne peuvent pas disposer d'une vue globale du système).

Les constructions que nous envisagions étaient guidées par deux souhaits :

- Nous voulions que ces approches soient suffisamment génériques pour pouvoir traiter plusieurs problèmes grâce aux mêmes outils.
- Nous voulions réduire le rôle du concepteur dans le processus de construction du système et nous concentrer sur des constructions automatiques.

Ces deux considérations nous ont orientés vers l'utilisation de cadre formel pour exprimer un problème global, caractériser les comportements des agents et les manipuler pour produire une réponse collective à ce problème.

7.1.2 Démarche suivie

7.1.2.1 Formalismes markoviens

Afin de proposer des processus génériques, nous avons été ainsi amenés à considérer les formalismes issus des processus de décisions markoviens dans lesquels :
- il est possible d'exprimer un problème sous la forme d'un problème d'optimisation
- il est possible de formaliser des comportements réactifs
- il est possible d'exprimer des systèmes constitués de plusieurs agents en interaction
- il est possible de prendre en compte les incertitudes liées à des lois d'évolution non déterministes, des perceptions partielles ou des méconnaissances des comportements des autres agents
- il est possible de construire des comportements réactifs adaptatifs dans certains circonstances (mono-agent, observabilité individuelle totale)

Les DEC-POMDPs qui font partie de cette famille permettent en outre de représenter des problèmes de prises de décision multi-agents. Cependant, ils constituent une simple extension des Processus de Décision Markovien mono-agent caractérisés par une exécution décentralisée :
- la construction des politiques des agents est un problème NEXP
- au niveau d'un agent, la présence d'autres agents n'est pas explicitement représentée
- ces cadres ne spécifient rien quant à la construction des politiques et ne s'intéressent pas à proposer des entités manipulables par un agent

L'avantage des DEC-POMDP réside dans le fait qu'il est possible lorsque l'on dispose d'une vue globale du système de ramener la résolution d'un DEC-POMDP à un MDP de grande taille sous certaines conditions et d'avoir des preuves de convergence. Cependant, ce genre d'approches se heurte à l'explosion combinatoire du nombre d'états et d'actions.

De plus, les DEC-POMDPs ne considèrent pas :
- la présence d'interactions locales entre les agents qui peuvent constituer un élément permettant de résoudre de nombreux problèmes
- la structure du système inhérente au fait qu'il s'agisse d'un système multi-agents

Les différentes approches de résolution ont cherché à compenser ce manque de représentation de la collectivité en réintroduisant du collectif
- au niveau des algorithmes (comme le principe de co-évolution [Cha02]) mais en s'accompagnant d'une forme de centralisation (synchronisation globale des différents agents dans ce cas)
- au niveau du cadre formel en modifiant l'expressivité du formalisme (introduction de communications) mais en ne respectant pas certaines de contraintes de localités inhérentes à ces systèmes (communications globales entre tous les agents)

7.1.2.2 Introduction d'interaction et inspiration biologique

Notre proposition a consisté à reconsidérer la notion de système et la manière dont celui-ci est représenté pour :
- introduire explicitement la présence d'autres agents au niveau individuel (et pas uniquement dans la description du système global) afin de pouvoir raisonner individuellement sur la présence d'autres entités

- vérifier si l'introduction de l'aspect social au niveau de l'agent permet de tirer parti de principes de construction de comportements individuels pour bâtir des comportements collectifs.
- concevoir des algorithmes permettant de construire des systèmes collectifs sur ce nouveau formalisme à partir des nouvelles possibilités offertes aux agents.

Pour partir sur de nouvelles bases, nous avons reconsidéré l'apprentissage par renforcement à ses sources d'inspiration : les phénomènes biologiques. Nous nous sommes concentrés sur des systèmes collectifs existants dans la nature pour essayer de comprendre la manière par laquelle les individus parviennent à s'organiser et à prendre en compte la présence d'autres individus pour répondre à un problème posé au groupe. Notre objectif en faisant cette étude était d'en extraire des principes permettant d'envisager des apprentissages collectifs.

Nous avons donc cherché des phénomènes biologiques pour lesquels des mécanismes d'adaptation locale ont pu être mis en évidence ainsi que des processus permettant de tirer parti de ces mécanismes d'adaptation individuels pour produire des réponses collectives. Les groupes de rats soumis à des contraintes environnementales d'accés à la nourriture exhibent des comportements différents en fonction du nombre d'agents ce qui suggère des processus d'adaptation et la prise en compte au niveau d'un individu de ces contexte social.

Nous avons ainsi cherché à comprendre les phénomènes de régulation mis en œuvre et comment le contexte social et la présence d'autres individus pouvait être intégrés au niveau des prises de décision d'un individu. Aidés par les éthologues qui se posaient des questions identiques, nous avons proposé un modèle rendant compte de ce phénomène : le modèle Hamelin. Ce modèle utilise la notion d'interaction directe pour prendre des décisions collectives locales à partir de confrontation de valeur de dominance. La particularité de ce modèle réside dans la manière dont l'interaction est construite et résolue : celle-ci permet de prendre en compte au niveau individuel la notion d'environnement social sans nécessiter de modèle complexe des autres agents du système.

7.1.2.3 Proposition d'un nouveau formalisme

Nous avons formalisé les principes découverts en éthologie et avons proposé un formalisme original : l'Interac-DEC-POMDP fondé sur le modèle Hamelin et issu des DEC-POMDP. Ce cadre formel permet de représenter des agents, les interactions directes possibles, les lois d'évolution du système et un problème posé à la collectivité par l'intermédiaire de fonctions de récompense.

Toujours en s'inspirant du modèle Hamelin, nous avons proposé un algorithme permettant de résoudre certains problèmes collectifs. Dans Hamelin, la résolution d'une interaction se fait en confrontant des valeurs de dominance des agents. Les valeurs de dominance constituent la capacité qu'a un agent à imposer le résultat d'une interaction. Dans notre résolution, cette capacité à imposer un résultat donné correspond à l'évaluation locale de l'avancement de la tâche si ce résultat est décidé. Ainsi, comme Hamelin, notre approche pour construire des comportements est basée sur

- des mécanismes d'adaptation individuels
- et des mécanismes d'adaptation collectifs fondés sur des confrontations de variables individuelles
- sans nécessiter de représentation complexe des autres

7.1.2.4 Manipulation du formalisme

Afin de valider notre approche, nous l'avons appliquée à des systèmes basés sur des observabilités partielles, dans lesquels sont posés des problèmes collectifs et qui nécessitent d'intégrer l'environnement social pour résoudre la tâche. Un problème adapté était le problème des pompiers :

- Les problèmes posés dans ce cadre restent complexes parce qu'ils nécessitent un **apprentissage collectif** : chaque agent doit intégrer à un instant ou un autre les autres agents du système pour effectuer une action pertinente. La présence d'un autre agent même situé loin peut modifier totalement le comportement à émettre.
- Ces problèmes peuvent s'exprimer sous la forme d'une sous-classe du problème Interac-DEC-POMDP

Nous avons enfin pu montrer que notre approche permet de modéliser le problème des pompiers et qu'il est possible d'effectuer des apprentissages collectifs de manière entièrement décentralisée pour construire une réponse collective à plusieurs instanciations de ce problème.

7.2 Contributions

Les travaux développés selon cette démarche ont donné lieu à plusieurs contributions : deux contributions majeures qui constituent une réponse à notre objectif premier : le formalisme Interac-DEC-POMDP et une algorithmique permettant de manipuler une sous-classe du formalisme Interac-DEC-POMDP ainsi qu'une contribution qui est un effet de bord de la démarche qui a été suivie : le modèle Hamelin.

7.2.1 Cadre formel Interac-DEC-POMDP

Le cadre formel Interac-DEC-POMDP est fondé sur le couplage de deux modules :

- un module d'action qui permet de modéliser les actions que peuvent émettre les agents et les conséquences que celles-ci peuvent avoir sur l'environnement
- un module d'interaction qui permet de formaliser la notion d'interaction directe impliquant des décisions jointes entre plusieurs agents.

L'Interac-DEC-POMDP représente dans un cadre homogène des actions et des interactions de manière explicite. Ce cadre formel permet ainsi de représenter un élément fondamental des systèmes multi-agents qui est négligé dans les DEC-POMDPs : la notion d'interaction directe entre agents qui constitue l'un des moyens par lequel les agents peuvent s'influencer.

La représentation explicite de l'interaction permet de structurer le système multi-agents. Elle définit de nouvelles entités qu'il va être possible de manipuler comme les ensembles d'agents en interaction.

En outre, l'Interac-DEC-POMDP offre une structure permettant de prendre des décisions collectives locales. Une des caractéristiques du formalisme Interac-DEC-POMDP réside dans le fait que les actions et les interactions sont forcément dues à des initiatives individuelles : une action est décidée par un agent et une interaction est déclenchée par un agent. Il permet ainsi de représenter des mécanismes de prise de décision collective tout en se basant sur des agents autonomes et assure l'exécution du système sans nécessiter de contrôleur global.

De plus, une des caractéristiques de l'Interac-DEC-POMDP réside dans la capacité donnée aux agents de restructurer leur réseau de relations en choisissant les agents avec lesquels ils veulent agir.

Enfin, l'introduction de la notion d'interaction directe au niveau individuel permet à des agents de pouvoir raisonner explicitement sur la présence d'autres agents dans le système et de pouvoir prendre en compte les caractéristiques comportementales des autres agents avec lesquels il est possible d'interagir.

7.2.2 Processus d'apprentissage

Nous avons aussi proposé un processus d'apprentissage permettant de tirer parti de la notion d'interaction pour construire automatiquement des systèmes fournissant une réponse collective à des problèmes complexes pour lesquels :

- les agents sont régis par des contraintes de localités (communications limitées)
- les agents ne disposent que d'observations partielles

L'algorithmique proposée se fonde sur l'idée que des échanges d'une partie des fonctions de valeur des agents au cours des interactions permettent de transmettre implicitement des informations concernant :

- les comportements des agents vis à vis de la tâche
- les perceptions dont disposent les agents
- les perceptions des agents sur l'avancement local de la tâche

Ce processus d'apprentissage original que nous avons proposé est constitué par

- des apprentissages par renforcement individuels décentralisés correspondant à nos contraintes de localité
- des heuristiques fondées sur des échanges des fonctions de Q-valeurs pour prendre des décisions collectives locales à partir des apprentissages individuels
- des échanges de récompenses sociales permettant de distribuer la tâche au sein de la collectivité.

Nous avons montré qu'il est alors possible de tirer parti des processus d'apprentissage par renforcement pour construire des réponses collectives à un problème global sans que les agents n'aient une vue globale du système.

Les expériences que nous avons effectuées ont ainsi pu montrer que notre proposition :

- permet de construire des comportements collectifs sans explosion combinatoire
- permet d'obtenir des résultats qualitatifs supérieurs à ce qu'il est possible d'obtenir avec une formalisation du même problème avec des DEC-POMDP
- permet de construire des solutions adaptatives : les organisations qui apparaissent parviennent à se restructurer en fonction des modifications de l'environnement et les agents n'ont pas besoin de l'ensemble des informations sur le système pour prendre de bonnes décisions.
- ne nécessite pas de modèle du monde ni de modèle complexe des autres agents.
- parvient à limiter les échanges d'informations aux échanges utiles et qu'il est possible de limiter les interactions déclenchées dans le système

Ces algorithmes ont mis en évidence qu'il est possible de construire des comportements collectifs à partir de processus de construction d'agents égoïstes (guidés par des signaux de récompenses individuels) et de prise en compte d'autres agents dans le système. Ils constituent selon nous une piste importante pour envisager des apprentissages collectifs dans des cas plus généraux d'Interac-DEC-POMDP dans le futur.

7.2.3 Modèle Hamelin

Au cours de notre démarche, nous avons été amené à développer le modèle Hamelin.

Il s'agit d'un modèle fondé sur un module de sélection d'action et un module d'interaction. A partir de règles de renforcement locales, ce modèle parvient à construire des comportements collectifs adaptatifs caractérisés par l'apparition d'une spécialisation au sein de groupes d'agents initialement homogènes.

En plus de nous avoir guidé pour proposer le cadre Interac-DEC-POMDP et un algorithme d'apprentissage distribué, ce modèle présente des intérêts pour :

- la compréhension des phénomènes biologiques de spécialisation observés dans des groupes de rats : il a permis de prouver que les rats n'avaient pas besoin de disposer de capacités cognitives complexes (comme la reconnaissance des autres) pour produire ce comportement collectif. Par exemple, il a mis en évidence la nécessité de reconsidérer certaines hypothèses biologiques pour expliquer les phénomènes de redifférenciation.
- la résolution de systèmes complexes : bien que par souci de généricité, nous ayons proposé de nous concentrer sur un cadre formel, il est envisageable d'utiliser le modèle Hamelin et ses lois d'évolution pour résoudre des problèmes posés en informatique comme la répartition de charge dans un réseau. De telles approches ont donné lieu à des stages (stage de DEA [Diz03]) et seront poursuivies par la suite.

7.3 Perspectives

La mise en œuvre que nous avons proposée constitue une première approche pour l'utilisation du formalisme Interac-DEC-POMDP : nous nous sommes limités à un type de problème, celui des pompiers, en nous concentrant sur une heuristique particulière.

Cependant, l'introduction au niveau individuel de la présence explicite d'autres agents dans le système ouvre de nombreuses pistes de recherche que nous souhaitons explorer dans le futur.

7.3.1 Perspectives à court terme

7.3.1.1 Formalisme : Coût d'interaction

Pour le moment, effectuer une interaction directe ne génère pas de récompense. Les interactions directes constituent un moyen de prendre des décisions collectives, d'effectuer certaines actions jointes.

Nous pensons par la suite nous intéresser à des problèmes plus complexes pour lesquels l'interaction directe peut avoir un coût ou rapporter des récompenses à la collectivité. Ces coûts peuvent être liés aux dépenses d'énergie nécessaires à effectuer des échanges de signaux ou au

fait que l'on souhaite interagir uniquement lorsque la situation l'exige.

L'introduction de récompenses lors des exécutions des résultats d'interaction conduit à des prises de décision plus complexes pour lesquelles un agent doit non seulement évaluer l'intérêt qu'il a à déclencher une interaction directe mais aussi le coût occasionné par l'utilisation de cette interaction.

7.3.1.2 Applications : Autres problèmes

Pour le moment, nous nous sommes limités à un seul type de problème : celui des agents pompiers pour lesquels les interactions directes constituent les seules influences possibles entre agents. Maintenant que nous avons montré qu'il est effectivement possible de construire des comportements collectifs sur ces problèmes, nous nous pencherons sur d'autres problèmes plus complexes pour lesquels les agents peuvent s'influencer en outre par des interactions indirectes.

Un problème analogue à celui des pompiers pour lequel les agents peuvent se déplacer dans l'intégralité de l'environnement nous semble adéquat dans ce cadre. Deux agents ne peuvent pas se trouver sur une case identique, et si deux agents sont cote à cote ils peuvent effectuer des interactions consistant à échanger des seaux.

Nous envisageons de traiter ce problème à l'aide d'interacteurs locaux :
- chaque agent connaît a priori l'environnement dans lequel la collectivité évolue
- par contre, il ne connaît pas la position des autres agents ni leurs comportements qui peuvent évoluer au cours de l'exécution du système
- il peut par contre observer localement son environnement pour voir les autres agents à une certaine distance de sa position.

Nous pensons que ce problème constitue un problème intéressant car :
- les agents sont confrontés à des choix entre actions ou interactions : un agent doit choisir entre contourner les agents qui le gênent ou tenter d'interagir avec eux
- les agents ne perçoivent que partiellement les autres agents du système ce qui nous semble une des principales sources d'incertitude sur le monde qui les entoure et qui constitue une contrainte présente dans beaucoup de systèmes.
- le fait que les agents connaissent la structure de leur environnement a priori ne constitue pas une contrainte forte
- ce type d'approche nous semblent envisageable dans beaucoup de problèmes impliquant des agents situés dotés de perceptions partielles.

L'objectif serait de vérifier s'il est possible de générer une organisation spécifique : dans ce cas, ce serait une spécialisation conséquences des positions des agents permettant de résoudre la tâche.

7.3.1.3 Formalisme : Semi-Markov Decision Process

Pour le problème des pompiers, nous avons supposé que toutes les politiques individuelles potentiellement optimales prennent le même temps d'exécution et nous avons négligé un certain nombre de problèmes de synchronisation entre les agents.

Afin d'envisager de résoudre ces problèmes, nous pensons nous intéresser à d'autres cadres formels qui intègrent explicitement la notion de temps nécessaire à exécuter une action dans le

modèle : les Semi Markov Decision Process (cf [WM99]).

7.3.1.4 Algorithmique : Autres heuristiques

Enfin, notre approche est fondée sur une heuristique de partage consistant à distribuer de manière équitable le gain collectif dû à une interaction entre les agents.

Cette heuristique n'est pas suffisante pour construire le comportement optimal dans le cadre du problème des pompiers. De nombreuses autres heuristiques sont envisageables et nous semblent intéressantes.

Une idée que nous proposons de mettre en place serait de permettre aux agents d'apprendre au fur et à mesure de l'exécution du système la manière dont ils peuvent répartir leurs récompenses individuelles au sein de la collectivité ou de voir comment les approches proposées par Guestrin peuvent s'appliquer lorsque les communications sont limitées à des communications locales au cours des interactions.

7.3.2 Perspectives à moyen terme

7.3.2.1 Algorithmique : Détection des interactions utiles

Pour le moment, bien que la formalisation de la notion d'interaction permette de représenter de nombreuses interactions dans la système, c'est toujours au concepteur de proposer des interactions pertinentes, utiles à la résolution de la tâche.

Une question à se poser dans un futur proche serait de savoir s'il est possible de transformer un DEC-POMDP en Interac-DEC-POMDP en cherchant les actions jointes pour lesquelles la représentation sous forme d'interaction permettrait de structurer le système et de construire de meilleures solutions.

7.3.2.2 Formalisme : Meta-interaction

Une autre approche très proche de l'approche précédente consisterait à ne pas partir d'interactions a priori mais à définir des meta-interactions. Ces meta-interactions seraient des modèles permettant de construire des interactions à partir des actions définies dans un DEC-POMDP.

Par exemple, un concept de méta-interaction possible serait celui d'obligation. Un agent A peut tenter de forcer par une interaction directe un agent B à effectuer une certaine action.

Ces méta-interactions peuvent alors constituer un moyen pour représenter un DEC-POMDP sous la forme d'un Interac-DEC-POMDP et tirer parti de cette représentation en permettant aux agents de s'influencer pour des actions ponctuelles.

7.3.2.3 Formalisme : Interactions plus complexes

Il nous semble aussi intéressant d'envisager des cas d'interaction directe plus complexes.

Les interactions directes peuvent être plus complexes de deux manières :
- soit parce que ces interactions impliquent plus d'agents.
- soit parce que ces interactions peuvent s'exécuter de manière séquentielles en impliquant des agents communs (quand A souhaite interagir avec B qui souhaite interagir avec C).

Ces interactions sont représentables dans le formalisme Interac-DEC-POMDP tel que nous l'avons présenté dans ce manuscrit, mais pour le moment, aucun problème utilisant cette capacité du formalisme n'a été implémenté. Il s'agit d'une voie que nous souhaitons aussi explorer par la suite.

7.3.3 Perspectives à long terme

7.3.3.1 Formalisme : Interaction dans le temps et rôles

Un des problèmes qui est posé dans la construction des DEC-POMDP réside dans le fait qu'un agent dispose de perceptions partielles et qu'il doit disposer de l'ensemble de ses observations passées pour prendre l'action optimale.

Dans des situations où un agent doit amener successivement des objets à un autre agent qu'il ne voit pas, le problème de l'absence de mémoire à court terme pour l'agent se pose. En effet, il ne peut donc pas se souvenir des objets qu'il a déjà amenés et de celui qu'il faut désormais apporter.

Un moyen serait d'introduire des mémoires internes à court terme. Des interactions possibles entre agents consisteraient à modifier les mémoires des autres agents. Chaque agent disposerait de mémoire individuelle modifiable au cours d'une interaction par les autres agents.

De cette manière, il serait possible :

d'inciter un agent à effectuer des actions dans le long terme :

Un agent A en modifiant la mémoire d'un agent B modifie l'état interne de B et les actions que B pourra déclencher par la suite. On peut très bien imaginer par exemple que par apprentissage, B apprenne à exécuter dans le futur une certaine action [21] parce que A lui a appris à coupler les récompenses sociales qu'il recevra avec les actions qu'il peut faire et le contenu de sa mémoire modifiée par A.

de transmettre des informations d'un agent à un autre :

Dans l'exemple consistant à transmettre des cubes dans un certain ordre, l'interaction consistant à modifier la mémoire de l'agent transporteur peut permettre de faire émerger une forme de langage. Une approche de ce type est présentée dans [PWD02] : lorsqu'un agent reçoit un objet, il envoie un message à un autre agent, le comportement de l'autre agent est alors modifié. Dans certains cas, un apprentissage permet de synchroniser ces politiques et de faire émerger

[21] comme appuyer sur une manette dés qu'il en verra une

une communication et un langage guidé par le but.

d'introduire des notions de rôles et d'organisation explicite

Le modèle aalaadin [GF98] s'est intéressé à la notion d'organisation. Une organisation est caractérisée par des rôles attribués aux agents, un agent pouvant avoir plusieurs rôles dans des organisations différentes. L'interaction telle que nous l'avons définie ressemble à cette notion d'organisation : chaque agent prend un rôle au sein de l'interaction (émetteur/receveur pour deux agents) et les actions jointes correspondant au résultats possibles sont déterminées par ces rôles. Un introduisant la notion de mémoire, ces rôles pourraient avoir une certaine durée et modifier le comportement d'un agent.

Il nous semble ainsi intéressant de poursuivre dans cette direction pour voir dans quelle mesure le formalisme pourrait operationnaliser la notion de groupe et de rôle et permettre de calculer l'attribution d'organisations et de rôles explicites dans un système de manière distribuée.

7.3.3.2 Algorithmique : Niveau de centralisation

Nous avons vu que l'interaction permet de prendre des décisions collectives et d'intégrer explicitement la présence d'autres agents dans ses décisions.

Les interactions permettent ainsi de considérer le système à des degrés divers : au niveau individuel, au niveau collectif (lorsqu'il est possible d'avoir une vision globale du système) et à des niveaux intermédiaires. Il peut être intéressant d'explorer cette piste pour s'intéresser à des interactions impliquant de plus en plus d'agents pour résoudre les situations de conflits.

Ainsi dans certaines situations, les interactions impliquant deux agents sont insuffisantes pour pouvoir répondre convenablement à un problème. Des interactions impliquant un plus grand nombre d'agents sont alors nécessaires. L'interaction permet alors d'aborder plusieurs niveaux de centralisations, éventuellement jusqu'à prendre des décisions jointes impliquant tous les agents comme cela peut être le cas avec l'interprétation des approches de [MHK+98]

7.4 Conclusion finale

Dans ce manuscrit, nous avons réhabilité la notion d'interaction en tant que brique de base dans des cadres formels manipulables. Nous avons mis en évidence le concept d'interaction directe, nous l'avons intégré dans un cadre formel et avons prouvé sa force sur un problème particulier.

La notion d'interaction permet d'appréhender les systèmes à des niveaux de décision divers. Cette caractéristique permet alors d'évaluer localement les conséquences de certaines actions jointes en prenant en compte la présence d'autres agents dans le systèmes. Il est ainsi possible de décider collectivement d'actions jointes ponctuelles tout en respectant des contraintes de localité.

En proposant un cadre formel inspiré des DEC-POMDP intégrant la notion d'interaction directe, nous avons souhaité :

- introduire plus de rationalité dans les systèmes multi-agents par la notion de récompense qui permet de définir des problèmes multi-agents sous la forme de problèmes d'optimisation.

- introduire plus de collectif dans les problèmes de décision par la notion d'interaction directe dans les modèle markoviens qui permet d'intégrer au niveau individuel la présence de plusieurs agents pour produire des comportements collectifs

Ce cadre formel a permis de développer des approches de construction de comportements décentralisées. Ces approches parviennent

- à résoudre des problèmes purement collectifs à partir d'individus égoïstes n'ayant pas de vue globale du système
- à construire de manière entièrement décentralisée et à moindre coût des organisations utiles à la tâche en tirant parti de la multiplicité des individus et de leur représentation explicite au niveau individuel.
- à générer des comportements collectifs qualitativement différents de comportements individuels qui nécessite d'intégrer/de considérer la présence d'autres agents dans le système.

L'interaction a donc permis de construire des systèmes tels que nous les avons souhaités :

- réactifs : car les agents sont régis par des règles de type stimulus-réponse
- automatiquement : car le processus de construction des comportements ne nécessite pas d'intervention extérieure
- coopératif : car l'objectif est de résoudre un problème défini globalement
- avec des communications limitées : car les agents ne peuvent échanger de l'information que lorsqu'ils sont proches selon une topologie définie dans le cadre du problème
- avec des perceptions partielles : car les agents ne perçoivent pas l'état global du système ni les comportements des autres agents

Ces travaux ouvrent de nombreuses perspectives aussi bien au niveau des systèmes sur lesquels ils peuvent s'appliquer, au niveau des développements possibles du formalisme qu'au niveau des algorithmes à mettre en œuvre.

La représentation explicite de l'interaction directe en permettant aux agents d'évaluer de manière plus précise les conséquences des actions sur les autres agents présents dans le système constitue selon nous la brique de base permettant d'envisager à long terme des apprentissages collectifs décentralisés.

Bibliographie

[BC01] C. Bourjot and V. Chevrier. Multi-agent simulation in biology : application to social spiders case. *Agent Based Simulation Workshop II, Passau*, pages 18–23, 2001.

[BCT03] C. Bourjot, V. Chevrier, and V. Thomas. A new swarm mechanism based on social spiders colonies : from web weaving to region detection. *Web Intelligence and Agent Systems : An International Journal - WIAS*, 1(1) :47–64, Mar 2003.

[BDC98] F. Le Ber, A. Dury, and V. Chevrier. Un modèle multi-agents pour la simulation en agronomie : usages et comparaisons. *Actes des 6èmes Journées Francophones IAD-SMA*, 1998.

[BDT99] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence : From Natural to Artificial Systems*. Oxford University Press., 1999.

[BGIZ02] D. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4) :819–840, 2002.

[Bou99] C. Boutilier. Sequential optimality and coordination in multiagent systems. In *IJCAI '99 : Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 478–485, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[BP04] L. Bisognin and S. Pesty. Emotions et systèmes multi-agents : une architecture d'agents émotionnels. *Journées francophones des Systèmes multi-Agents JFSMA 2004*, pages 307–320, 2004.

[Bro91] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47 :139–159, 1991.

[BT94] E. Bonabeau and G. Theraulaz. *Intelligence Collective*. Hermes, 1994.

[BT99] E. Bonabeau and G. Theraulaz. *Swarm intelligence*. Oxford university press, 1999.

[BTD96] E. Bonabeau, G. Theraulaz, and J. L. Deneubourg. Mathematical models of self-organizing hierarchies in animal societies. *Bulletin of Mathematical Biology*, 58 :661–717, 1996.

[Buf03] O. Buffet. *Une double approche modulaire de l'apprentissage par renforcement pour des agents intelligents adaptatifs*. PhD thesis, Université Nancy I, 2003.

[BZL04] R. Becker, S. Zilberstein, and V. Lesser. Decentralized markov decision processes with event-driven interactions. In *AAMAS '04 : Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, 2004.

[BZLG03] R. Becker, S. Zilberstein, V. Lesser, and C. Goldman. Transition-independent decentralized markov decision processes. In *AAMAS '03 : Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 41–48, New York, NY, USA, 2003. ACM Press.

[CB98] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI*, pages 746–752, 1998.

[CCF+99] R.S. Cost, Y. Chen, T. Finin, Y. Labrou, and Y. Peng. Modeling agent conversations with colored petri nets. *Proc of the Workshop on Specifying and Implementing Conversation Policies*, pages 59–66, 99.

[CFS+01] S. Camazine, N. R. Franks, J. Sneyd, E. Bonabeau, J. L. Deneubourg, and G. Theraulaz. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, USA, 2001.

[Cha02] I. Chades. *Planification distribuée dans les systèmes multi-agents à l'aide de processus décisionnels de Markov*. PhD thesis, Université Nancy I, 2002.

[CKK96] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien. Acting under uncertainty : Discrete bayesian models for mobile robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[CKL94] A. R. Cassandra, L. P. Kaelbling, and M. 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.

[CTB+05] M. C. Cotel, V. Thomas, C. Bourjot, D. Desor, V. Chevrier, and H. Schroeder. Processus cognitifs et differenciation sociale de groupes de rats : interet de la modelisation multi-agents. *6e colloque des jeunes chercheurs en sciences cognitives*, Mai 2005.

[DC99] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In David Corne, Marco Dorigo, and Fred Glover, editors, *New Ideas in Optimization*, pages 11–32. McGraw-Hill, London, 1999.

[Dem97] Y. Demazeau. Steps towards multi-agent oriented programming. *1st International Workshop on Multi-Agent Systems(IWMAS'97)*, 1997.

[DF93a] A. Drogoul and J. Ferber. From tom-thumb to the dockers : Some experiments with foraging robots. *From Animals to Animats II*, pages 451–459, 1993.

[DF93b] A. Drogoul and J. Ferber. From tom-thumb to the dockers : Some experiments with foraging robots. *From Animals to Animats II*, 1993.

[DG89] J. L. Deneubourg and S. Goss. Collective patterns and decision making. *Etholgy, ecology and evolution*, 1 :295–311, 1989.

[Diz03] A. St Dizier. Les processus décentralisés d'organisation et leurs applications potentielles. *Diplome de DEa, universite Henri Poincare, Nancy*, 2003.

[dKL98] M. d'Inverno, D. Kinny, and Michael Luck. Interaction protocols in agentis. *ICMAS'98, Third International Conference on Multi-Agent Systems*, pages 112–119, 1998.

[DKTD91] D. Desor, B. Krafft, A. M. Toniolo, and P. Dicked. Social cognition in rats : incentive behaviour related to food supply. *XXIInd Int. Ethologica conference*, 1991.

[DLC89] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1) :63–83, 1989.

[DMC96] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system : Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26 :29–41, 1996.

[DS03] A. Dutech and M. Samuelides. Apprentissage par renforcement pour les processus décisionnels de markov partiellement observés. *Revue d'Intelligence Artificielle, RIA*, 17, 2003.

[DT92] D. Desor and A. M. Toniolo. Incentive behaviour in structure groups of rats : about the possible occurence of socio-cognitive processes. *Comparative approach in sciences cognitives*, 1992.

[Dur00] A. Dury. *Modélisation des interactions dans les systèmes multi-agents*. PhD thesis, Université UHP Nancy-1, 2000.

[DVB+01] A. Dury, G. Vakanas, C. Bourjot, V. Chevrier, and B. Krafft. Using multi-agent system to model prey capture in social spiders. *ESS01 13th European Simulation Symposium*, pages 831–833, 2001.

[Fer95] J. Ferber. *Les systèmes multi-agents. Vers une intelligence collective*. InterEditions, 1995.

[Fer97] J. Ferber. Les systèmes multi-agents : un apercu general. *Technique et science informatique*, 16 :979–1012, 1997.

[FFMM94] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an Agent Communication Language. In N. Adam, B. Bhargava, and Y. Yesha, editors, *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463, Gaithersburg, MD, USA, 1994. ACM Press.

[FJ91] J. Ferber and E. Jacopin. The framework of eco-problem solving. In Y. Demazeau and J.-P. Müller, editors, *Decentralized A.I. 2 : Proc. of the 2nd European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 181–193. North-Holland, Amsterdam, 1991.

[FM96] J. Ferber and J. P. Müller. Influences and reaction : A model of situated multiagent systems. *International Conference on Multi-Agent Systems, 1996*, 1996.

[Foi98] R. Foisel. *Modèle de réorganisation de systèmes multi-agents : une approche descriptive et opérationnelle*. PhD thesis, Université UHP Nancy-1, 1998.

[GAZ04] C. V. Goldman, M. Allen, and S. Zilberstein. Decentralized language learning through acting. *Proc. 3rd Intl. Joint Conf. on Autonomous Agents and Multi Agent Systems*, pages 1006–1013, 2004.

[GF98] O. Gutknecht and J. Ferber. Un meta-modèle organisationnel pour l'analyse, la conception et l'exécution de systèmes multi-agents. *actes de journées francophones sur les systèmes multi-agents 98*, 1998.

[GKP01] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. *Advances in Neural Information Processing Systems (NIPS 2001)*, pages 1523 – 1530, 2001.

[GKPV03] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research (JAIR)*, 19 :399–468, 2003.

[GLP02] C. Guestrin, M. Lagoudakis, and R. Parr. Coordinated reinforcement learning. *Nineteenth International Conference on Machine Learning(ICML 2002)*, pages 227 – 234, 2002.

[Gra59] P. P. Grasse. La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp., la théorie de la stigmergie : essais d'interprétation du comportement des termites constructeurs. *Ins. Soc.*, 6 :41–84, 1959.

[Gue03] C. Guestrin. *Planning Under Uncertainty in Complex Structured Environments*. PhD thesis, Stanford University, August 2003.

[GZ03] C. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems, 2003.

[GZ04] C. Goldman and S. Zilberstein. Decentralized control of cooperative systems : Categorization and complexity analysis. *J. Artif. Intell. Res. (JAIR)*, 22 :143–174, 2004.

[Har68] G. Hardin. the tragedy of the commons. *Science*, pages 1243–1248, 1968.

[HBH88] E. J. Horvitz, J. S. Breese, and M. Henrion. Decision theory in expert systems and artificial intelligence. *Journal of Approximate Reasoning,Special Issue on Uncertainty in Artificial Intelligence*, pages 247–302, 1988.

[HBZ04] E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, pages 709–715, 2004.

[Hem96] C. K. Hemelrijk. Dominance interactions, spatial dynamics and emergent reciprocity in a virtual world. *fourth international conference on simulation of adaptive behavior*, pages 545–552, 1996.

[Hem99] C. K. Hemelrijk. An individual-oriented model on the emergence of despotic and egalitarian societies. *Proceedings of the Royal Society London B : Biological Sciences*, pages 361–369, 1999.

[Hem00] C. Hemelrijk. Towards the integration of social dominance and spatial structure. *Animal Behaviour*, pages 1035–1048, 2000.

[Jea97] M. R. Jean. Emergence et sma. *JFSMA97*, pages 323–342, 1997.

[JJD+02] C. Jost, R. Jeanson, JL. Denebourg, C. Rivault, and G. Theraulaz. Self-organized aggregation in cockroaches : sensitivity to model structure. *International Workshop on Self-Organization and Evolution of Social Behaviour*, 2002.

[Joh02] N. L. Johnson. the development of collective structure and its response to environmental change. *International workshop on self-organization and evolution of social behaviour*, pages 215–237, 2002.

[Joz01] J. Jozefowiez. *conditionnement opérant et problèmes décisionnels de markov, partie : Reinforcement learning and conditionning : an overview*. PhD thesis, Université Lille III, 2001.

[JSW98] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1) :7–38, 1998.

[Jud94] O. P. Judson. The rise of the individual-based model in ecology. *Trends in Ecology and Evolution*, 9 :372–377, 1994.

[KD93] J. R. Krebs and N. B. Davies. *An introduction to behavioral ecology (3rd ed.)*. Oxford : Blackwell Science, 1993.

[KG03] D. Keil and D. Goldin. Modeling indirect interaction in open computational systems. *1st Int'l workshop on Theory and Practice of Open Computational systems (TAPOCS)*, 2003.

[LF94] E. D. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. *From Animals to Animats 3*, pages 501–508, 1994.

[Lit94a] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, pages 157–163, New Brunswick, NJ, 1994. Morgan Kaufmann.

[Lit94b] M. L. Littman. Memoryless policies : Theoretical limitations and practical results. In *Simulation of Adaptive Behaviour (SAB-94)*, pages 238–245, 1994.

[Lit94c] M. L. Littman. The witness algorithm : Solving partially observable markov decision processes. Technical report, Brown University, Providence, RI, USA, 1994.

[LSC03] P. De Loor, C. Septseault, and P. Chevaillier. Les émotions : une métaphore pour la résolution de problèmes dynamiques distribués. *Déploiement des systèmes multi-agents, vers un passage à l'échelle, JFSMA'2003 Revue des sciences et technologies de l'information*, pages 331–344, November 2003.

[MAI02] MAIA. Rapport d'avant projet maia. Technical report, 2002.

[MHK+98] N. Meuleau, M. Hauskrecht, K. Kim, L. Peshkin, L. Kaelbling, T. Dean, and C. Boutilier. Solving very large weakly coupled markov decision processes. In *AAAI/IAAI*, pages 165–172, 1998.

[MV03] P. Mathieu and M. H. Verrons. Ants : an api for creating negotiation applications. *Proceedings of the 10th ISPE International Conference on Concurrent Engineering : Research and Applications (CE2003)*, 2003.

[Par97] H. Van Parunak. Go to the ant : Engineering principles from natural agent systems. *Annals of Operations Research*, pages 69–101, 1997.

[Par98] R. Parr. Flexible decomposition algorithms for weakly coupled Markov decision problems. *Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 422–430, 1998.

[Pic01] S. Picault. *Modèles de comportements sociaux pour les collectivités de robots et d'agents*. PhD thesis, Université Paris 6, 2001.

[PT02] D. Pynadath and M. Tambe. The communicative multi-agent team decision problem : analyzing teamwork : theories and models. *Journal of Artificial Intelligence Research*, 16 :389–423, 2002.

[Put94] M. L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.

[PWD02] S. Papendick, J. Wellner, and W. Dilger. Society first and then minds : Self organisation of a social symbol system by learning agents. *International Workshop on Self organisation and evolution of social behaviour*, pages 323–333, 2002.

[RD98] A. Ribeiro and Y. Demazeau. A dynamic interaction model for multi-agent systems. *Proceedings of the 2d Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent Systems*, pages 27–36, 1998.

[Rey87] C. Reynolds. Flocks, herds, and schools : A distributed behavioral model. *SIGGRAPH '87*, 1987.

[RG95] A. S. Rao and M. P. Georgeff. BDI-agents : from theory to practice. In *Proceedings of the First Intl. Conference on Multiagent Systems*, San Francisco, 1995.

[RN95] S. Russel and P. Norvig. *Artificial Intelligence : a modern approach*. Prentice Hall, New York, 1995.

[RW72] R. Rescorla and A. Wagner. A theory of pavlovian conditioning : Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II : Current research and theory*, pages 64–99, 1972.

[SB98] R. Sutton and A. Barto. Reinforcement learning : An introduction, 1998.

[SCZ05] D. Szer, F. Charpillet, and S. Zilberstein. Maa*, a heuristic search algorithm for solving decentralized decpomdps. *UAI*, 2005.

[SGP03] Y. Shoham, T. Grenager, and R. Powers. Multi-agent reinforcement learning : A critical survey. Technical report, Stanford University, 2003.

[Sha01] C. Shalizi. *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*. PhD thesis, University of Wisconsin, 2001.

[Sig03] O. Sigaud. Comportements adaptatifs pour les agents dans des environnements informatiques complexes. Technical report, LIP6, Paris, 2003.

[Sig04] O. Sigaud. *Comportements adaptatifs pour des agents dans des environnements informatiques complexes*. Habilitation à Diriger des Recherches de l'Université PARIS 6, 2004.

[Sim01] O. Simonin. *Le modèle staisfaction-altruisme*. PhD thesis, Université Montpellier II, 2001.

[Smi88] R. G. Smith. The contract net protocol : High-level communication and control in a distributed problem solver. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 357–366. Kaufmann, San Mateo, CA, 1988.

[STND98] H. Schroeder, A.M. Toniolo, A. Nehlig, and D. Desor. Long-term effects of early diazepam exposure on social differentiation in adult male rats subjected to the diving-for-food situation. *Behavioural Neurosciences*, 112 :1209–1217, 1998.

[SWMR99] J. Schneider, W. Wong, A. Moore, and M. Riedmiller. Distributed value functions. In *Proceedings of the 16th International Conference on Machine Learning*, pages 371–378. Morgan Kaufmann, San Francisco, CA, 1999.

[Sze04] D. Szer. communication et apprentissage par renforcement pour une équipe d'agents. *Journées Francophones des Systèmes Multi-Agents, JFSMA2004*, pages 175–186, 2004.

[TBC04a] V. Thomas, C. Bourjot, and V. Chevrier. Interac-dec-mdp : Towards the use of interactions in dec-mdp. In *Third International Joint Conference on Autonomous Agents and Multi-Agent Systems - AAMAS'04, New York, USA*, pages 1450–1451, Jul 2004.

[TBC04b] V. Thomas, C. Bourjot, and V. Chevrier. Un formalisme pour la construction automatique dínteractions dans les sma réactifs. In *Journées Francophones sur les Systèmes Multi-Agents - JFSMA 2004, Paris, france*, Nov 2004.

[TBC06] V. Thomas, C. Bourjot, and V. Chevrier. Heuristique pour l'apprentissage automatique décentralisé d'interactions dans un système multi-agents réactif. *Reconnaissance de Forme et Intelligence Artcificielle, RFIA 06 (accepte)*, 2006.

[TBCD02] V. Thomas, C. Bourjot, V. Chevrier, and D. Desor. Mas and rats : Multi-agent simulation of social differentiation in rats groups. In *International Workshop on Self-Organization and Evolution of Social Behaviour, Monte Verita, Ascona, Switzerland*, Sep 2002.

[TBCD04] V. Thomas, C. Bourjot, V. Chevrier, and D. Desor. Hamelin : A model for collective adaptation based on internal stimuli. In Stefan Schaal, Auke Ijspeert, Aude Billard, Sethu Vijayakumar, John Hallam, and Jean-Arcady Meyer, editors, *From animal to animats 8 - Eighth International Conference on the Simulation of Adaptive Behaviour 2004 - SAB'04, Los Angeles, USA*, pages 425–434, Jul 2004.

[TCC05] V. Thomas, C.Bourjot, and V. Chevrier. Un formalisme pour la construction automatique dínteractions dans les smas réactifs(version longue). rapport interne, INRIA n°550, 2005.

[Thr92] S. B. Thrun. Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1992.

[TW00] K. Tumer and D. Wolpert. Collective intelligence and braess' paradox. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 104–109. AAAI Press / The MIT Press, 2000.

[WD92] C. J. C. H. Watkins and P. Dayan. Technical note q-learning. *Machine Learning*, 8 :279–292, 1992.

[Wei99] G. Weiss, editor. *Multiagent systems : a modern approach to distributed artificial intelligence*. MIT Press, Cambridge, MA, USA, 1999.

[Wil91] S. W. Wilson. The animat path to ai. *From Animals to Animats*, pages 15–21, 1991.

[WJ95] M. Wooldridge and N. R. Jennings. Intelligent agents : Theory and practice. *Knowledge Engineering Review*, 10(2) :115–152, 1995.

[WM99] G. Wang and S. Mahadevan. Hierarchical optimization of policy-coupled semi-markov decision processes. In *ICML '99 : Proceedings of the Sixteenth International Conference on Machine Learning*, pages 464–473, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[Woo01] M. Woolridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

Annexe A

Approches de résolution pour les DEC-POMDPs

Cette annexe propose l'éventail des principales approches utilisées dans des cadres markoviens pour résoudre des DEC-POMDPs. Elle constitue une extension des approches présentées dans la partie 3 en ajoutant des approches qu'il n'était pas nécessaire de présenter dans le déroulement de l'argumentation mais qui sont au coeur de nombreux travaux de recherche.

A.1 Approches centralisées

A.1.1 Résolution Directe

Objectif	Construire une politique jointe optimale
Contraintes	Environnement accessible, récompense globale, règles de l'environnement connues
Modèle	DEC-POMDP
Moyen	Utiliser des heuristiques de recherche

Certains travaux cherchent néanmoins à construire directement une politique jointe optimale de manière centralisée. Même si cette approche est forcément contrainte par la complexité du problème, il est possible de trouver des heuristiques pour limiter le parcours de l'espace de recherche.

[SCZ05] propose une recherche heuristique appelée MAA*. L'heuristique consiste à calculer la fonction de valeur du problème DEC-POMDP lorsque les agents peuvent s'échanger gratuitement leurs observations. Lorsque cette hypothèse est faite, il est possible de se ramener à un POMDP et de le résoudre à moindre coût. La fonction de valeur trouvée par cette résolution constitue alors une heuristique supérieure à la fonction de valeur réelle de la politique jointe optimale sans communication et peut être utilisée dans des algorithmes de recherche heuristique du genre A*.

Inconvénients Ces approches sont forcément fortement contraintes par la difficulté du problème qu'elles cherchent à résoudre (NEXP).

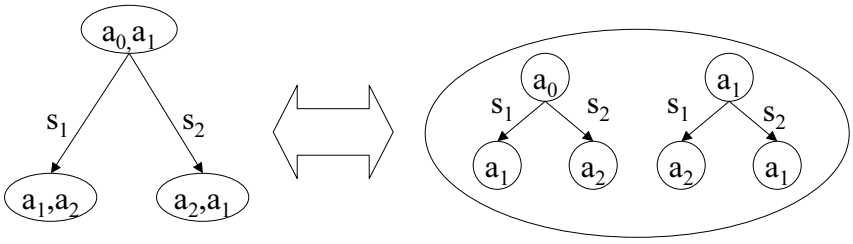


FIG. A.1 – MMDP et politiques jointes

A.1.2 Multi-agents Markov Decision Problem

Objectif	Construire une politique jointe optimale
Contraintes	Environnement accessible par chaque agent, récompense globale, lois d'évolution de l'environnement connues
Modèle	MMDP (Multi-agents Markov Decision Problem)
Moyen	Nouveau cadre formel, perception globale des agents, éventuellement capacités de communication pour se coordonner

Boutilier dans [Bou99] propose un modèle permettant de représenter des problèmes de décision multi-agents dans des systèmes entièrement coopératifs pour lesquels chaque agent perçoit l'état global du système : les MMDPs (Multiagent Markov Decision Problem).

Description Un MMDP est défini par un tuple $< \alpha, A_{ii\in\alpha}, S, Pr, R >$:
- α désigne le nombre d'agents du système
- chaque agent $i \in \alpha$ dispose d'actions individuelles décrites par A_i
- On définit $A = \times A_i$ l'espace des actions jointes.
- S désigne l'espace d'état
- Pr la matrice de transition. $Pr : S \times A \times S \rightarrow [0, 1]$
- R la fonction de récompense $R : S \times A \rightarrow \mathbb{R}$

Ce cadre peut être compris comme un DEC-POMDP pour lequel les agents ont une perception totale de leur environnement$(\forall i, s, O_i(s) = s)$. Cette propriété permet d'assimiler la recherche de la politique jointe à la recherche d'une politique sur les actions jointes. Une politique globale sur les actions jointes pourra en effet facilement être distribuée parmi les agents puisqu'ils connaissent l'état global du système (cf fig A.1).

Résolution On suppose en outre que tous les agents connaissent le modèle du monde et des récompenses. Chaque agent peut alors résoudre individuellement ce problème par planification en supposant qu'il contrôle le comportement de l'ensemble des agents du système. Résoudre un MMDP est équivalent à résoudre un MDP $< S', A', T', R' >$ pour lequel :
- S' est égal à S
- A' correspond à l'espace des actions jointes
- T' correspond à Pr
- R' correspond à R

En utilisant les algorithmes présentés dans la partie 3.1.2, chaque agent peut donc construire une politique optimale $\pi^* : S \rightarrow A'$ comme s'il contrôlait intégralement les actions émises par tous les agents. Si tous les agents disposent du même modèle du monde, ils peuvent alors décider à chaque instant quelle action émettre en supposant que les autres agents aient effectués la même planification.

Néanmoins, puisque l'exécution est décentralisée, un problème de coordination se pose quand deux actions jointes ont la même valeur. Dans ce dernier cas, chaque agent hésite entre plusieurs actions jointes et la réponse individuelle qu'il doit fournir. Si les agents choisissent des actions jointes différentes, cela peut grandement nuire à la performance du système.

Par exemple, considérons deux agents qui doivent traverser un pont qui ne peut supporter que le poids d'un seul. On suppose en outre que pour ce problème, la politique jointe pour laquelle l'agent A avance avant l'agent B est autant avantageuse que celle pour laquelle l'agent B avance avant l'agent A. L'action que l'agent A doit émettre dépend de la politique jointe optimale que B va choisir de suivre et réciproquement. Il est donc nécessaire d'introduire des mécanismes de coordination pour répondre à la problématique de répartition de la politique jointe en politiques individuelles.

Boutilier avance plusieurs propositions pour effectuer le choix d'une action jointe commune quand il y a ambiguïté. Il propose :

d'utiliser des synchronisations par apprentissage A l'exécution, chaque agent évolue selon une certaine politique individuelle et apprend les réactions des autres agents lors des situations de conflit jusqu'à ce que l'ensemble des agents se synchronise sur une action jointe optimale qu'ils répéteront dans le futur.

d'utiliser des conventions Les agent connaissent des règles qui permettent de lever l'ambiguïté sur l'action jointe à choisir en cas de conflit.

d'utiliser de la communication Les agents peuvent s'échanger des messages pour se mettre d'accord sur l'action jointe à choisir dans les situations de conflit.

Avantages L'approche proposée par Boutilier permet effectivement de construire des politiques jointes optimales en s'appuyant sur les techniques développées dans le cadre mono-agent.

Inconvénients Néanmoins, utiliser de telles approches se heurte à l'explosion combinatoire de la taille de l'espace d'états et de la taille de l'espace des actions jointes par rapport au nombre d'agents, ainsi qu'à la nécessité de pouvoir observer l'état global de l'environnement (ce qui s'oppose au principe de localité présenté dans la partie 2)

Par rapport à nos travaux Une des propositions de Boutilier nous intéresse au plus haut point pour la suite de nos travaux : il montre que les approches centralisées ne sont pas suffisantes pour résoudre l'ensemble des problèmes de coordination même si chaque agent perçoit l'ensemble du système. Il propose surtout une approche consistant à résoudre ces problèmes en synchronisant les comportements des agents par apprentissage.

A.1.3 Communication explicite

Objectif	Proposer un nouveau cadre pour chercher des politiques sous-optimales
Contraintes	Observabilité partielle, récompense perçue globalement
Moyen	Un nouveau cadre formel COMMTDP permettant de représenter des communication entre agents : les agents sont dotés de nouvelles capacités sur lesquelles ils peuvent raisonner.
Modèle	COMMTDP ou DECPOMDP-COM

[PT02] propose de représenter explicitement la communication entre agents. Pour cela, les travaux présentés dans [PT02] décrivent un nouveau cadre formel appelé COM-MTDP : Communicative Multiagent Team Decision Process.

Un COM-MTDP (communicative Multiagent Team Decision Problem) est défini par un tuple $< S, A_\alpha, P, \Gamma_\alpha, O_\alpha, B_\alpha, R >$:

- un ensemble d'états S
- des ensembles d'actions individuelles A_i
- P une matrice de transition dépendant de l'action jointe
- des ensembles d'observations individuels Γ_i
- des fonctions d'observation $O_i(s, a, w) = Pr(\Gamma_i^t = w | S^t = s, A_i = a)$
- B_α des belief states qui constituent l'originalité du modèle.
- R une fonction de récompense

Les politiques des agents sont basées sur une mémoire à court terme représentée par leur belief state. Une politique individuelle est donc une fonction $\pi_i : B_i \rightarrow A_i$. Les belief-state correspondent dans ce cas à l'état interne de l'agent et permettent à celui-ci d'intégrer des informations sur son environnement pour sa prise décision.

Ce modèle propose en outre l'ajout de communications Σ_α de manière séparée des actions. Ces communications ont pour objectif de modifier les états de croyances des agents. Un agent a peut envoyer un message x aux autres agents. Les autres agents perçoivent ses signaux et mettent à jour leur belief state en conséquence. Ainsi, l'action qu'ils émettront par la suite pourra être modifiée par les messages reçus des autres agents.

L'exécution d'un tel système se fait en deux phases : dans la première phase, les agents échangent simultanément des messages. Dans la seconde, chaque agent émet une action en fonction de sa politique et de son belief state. Les actions émises simultanément conditionnent l'évolution du système.

Construire un système multi-agents dans ce cadre consiste alors à définir les états internes des agents, leurs politiques d'action et de communication.

Avantages Les auteurs prouvent l'intérêt de ce modèle en établissant que lorsque les communications sont gratuites, la complexité de résolution en horizon fini d'un tel modèle est réduite. Il suffit aux agents de se communiquer leurs observations pour se ramener à un POMDP. La

recherche de la politique jointe est alors équivalente à la recherche d'une politique dans les espace-joint comme cela peut être fait pour un MMDP puisque les agents peuvent disposer à l'exécution de l'ensemble des informations connues par les autres agents.

L'intérêt de ce modèle consiste avant tout à proposer un nouveau cadre formel définissant une nouvelle classe de problèmes pour lequel les agents ont des capacités supplémentaires pour résoudre les problème de coordination.

Inconvénients Pour le moment, il s'agit d'un modèle et aucun algorithme n'a été proposée permettant d'utiliser ce modèle. En particulier, les fonctions de mises à jour des belief state en fonction des messages reçus n'ont pas été explicitées. Par exemple, une sélection des messages est nécessaire pour éviter une explosion combinatoire du nombre d'états (beliefstate) en entrée de la politique. De plus, les communications sont supposées être diffusées à l'ensemble des agents et les récompenses observables par tous, ce qui s'oppose au principe de localité que nous avons mis en avant.

Travaux connexes D'autres travaux se concentrent sur ce genre d'approche [GAZ04] et un autre modèle propose le même genre de représentation : les DEC-POMDP-com [GZ03]. Dans ces approches, la forme des politiques individuelles est donnée a priori : l'action d'un agent est fonction de l'historique de ses perceptions et des messages qu'il a reçu. Tous ces travaux sont encore récents et il n'y a pour le moment pas d'algorithme tirant parti explicitement de la notion de communication explicite entre agents.

A.1.4 Théorie des jeux

Objectif	atteindre les états d'équilibre dans un système constitué d'agents rationnels
Contraintes	Observabilité partielle, récompenses individuelles
Moyen	Algorithmique pour converger vers des situations d'équilibre
Modèle	jeux de markov, POIPSG

La théorie des jeux est initialement fondé sur la notion de jeu de matrice (matrix game). Il s'agit d'une matrice spécifiant les différents gains des agents en fonction de l'action jointe émise. Dans ces jeux, chaque agent a pour objectif de maximiser ses gains individuels et sait que les autres agents ont le même objectif.

Les jeux stochastiques constituent une extension des jeux de matrice et permettent de représenter des systèmes dynamiques : un jeu stochastique est défini comme [SGP03] un tuple $< N, S, A, R, T >$

- N désigne l'ensemble d'agents
- S un ensemble d'états
- $A = A_1, ..., A_n$ avec A_i l'ensemble des actions possibles pour l'agent i
- $R = R_1, ..., R_n$ avec $R_i : S \times A \rightarrow \Re$ l'ensemble des récompenses immédiates
- $T : S \times A \rightarrow \Pi(S)$ la matrice de transition.

Il s'agit d'un modèle proche des DEC-POMDPs, seule la manière dont les récompenses sont définies diffère. Plusieurs familles de sous-modèles ont intéressé le domaine de la théorie des jeux :

- les jeux à somme nulle [Lit94a] qui correspondent à des agents compétitifs : chaque agent essaie de maximiser sa récompense aux détriments des autres
- les jeux d'équipe ([CB98]) où les agents reçoivent les mêmes récompenses.

Le problème se pose alors en terme de rationalité individuelle : chaque agent à des récompenses individuelles propres et non en terme d'objectif collectif. La notion d'optimalité telle que nous l'avons abordée jusqu'à présent est désormais à revoir et s'exprime dorénavant sous la forme d'équilibre dans lequel chaque agent maximise son critère de performance tout en sachant que les autres agents maximise le leur. Résoudre un jeu stochastique consiste à trouver des algorithmes permettant d'atteindre une position d'équilibre dans lequel aucun agent n'a intérêt à modifier son comportement en supposant qu'un tel équilibre existe.

[Lit94a] propose l'algorithme minimax-Q comme extension du Qlearning pour des jeux stochastiques à somme nulle.

$$V_i(s) = max_{P_1 \in \Pi(A_1)} min_{a_2 \in A_2} \Sigma P_1(a_1) Q_1(s,(a_1,a_2))$$

L'agent apprenant cherche à trouver la stratégie P_1 maximisant ses récompenses escomptées sachant, que de son coté, l'autre agent va choisir l'action minimisant cette récompense.

[HBZ04] propose un algorithme pour trouver la politique jointe optimale dans les jeux stochastiques. Cette approche se fonde sur l'élimination itérée des stratégies dominantes et la programmation dynamique.

Inconvénients La théorie des jeux cherche à répondre à un autre problème que celle qui nous intéresse : alors que l'on cherche à construire des systèmes, la théorie des jeux cherchent à trouver les états qui peuvent constituer des états d'équilibre et à construire des processus permettant de les atteindre. Le problème est que de manière générale, plusieurs équilibres existent et rien ne justifie chercher ces équilibres dans un cadre coopératif mais plutôt de trouver un moyen de s'en sortir. [SGP03] remet ainsi en cause la notion d'équilibre comme finalité et cherche à définir un programme de recherche avec des objets d'études plus précis :

- quelle stratégie adopter face à une stratégie fixe
- comment expliquer les décisions humaines
- comment faire collaborer une machine avec un être humain dans ce cadre

Certaines de ces finalités permettent d'aborder les problématiques de conception proches de celles que nous avons suivies.

A.1.5 Sous-Classes de DEC-POMDP

Objectif	Recherche de politique optimale
Contraintes	Perceptions partielles, récompense globale
Modèle	'transition independent DEC-MDP', 'reward independent DEC-MDP'
Moyen	Se limiter à des classes particulières de DEC-MDP

Plusieurs travaux se sont intéressés à définir des sous-classes de DEC-POMDP à partir desquelles il devient possible de chercher des politiques optimales de manière efficace [GZ04]. Deux

classes de DEC-POMDPs sont au coeur des travaux présentés dans [BZLG03] et [BZL04].

Ces travaux s'intéressent à des systèmes constitués de deux agents et supposent que l'espace d'état peut se factoriser à l'aide de deux sous-espaces $S = S_1 \times S_2$ et que chaque agent i peut observer s_i. Le système est donc un DEC-MDP, puisque les observations des agents permettent de reconstituer l'état global.

Un DEC-MDP factorisé $< S, A_i, T, \Gamma_i, O, R >$ à deux agents est dit 'transition indépendant' s'il existe T_1 et T_2 vérifiant :

$$\forall s_2, a_2 : T(s'_1|(s_1, s_2), (a_1, a_2), s'_2) = T_1(s'_1|s_1, a_1)$$

$$\forall s_1, a_1 : T(s'_1|(s_1, s_2), (a_1, a_2), s'_2) = T_2(s'_1|s_1, a_1)$$

Un DEC-MDP factorisé $< S, A_i, T, \Gamma_i, O, R >$ est dit 'reward-independant' s'il existe R_1 et R_2 vérifiant :

$$R((s_1, s_2), (a_1, a_2), (s'_1, s'_2)) = R_1(s_1, a_1, s'_1) + R_2(s_2, a_2, s'_2)$$

Une flotte de robots devant explorer un environnement peut être modélisée par un DEC-POMDP 'transition-indépendant' : les actions entreprises par un agent n'ont pas d'influence sur le résultat des actions d'un autre agent et la matrice de transition T peut se décomposer en deux matrices T_1 et T_2. Un tel système n'est par contre pas 'reward-independant' : si les robots effectuent les mêmes tâches et explorent la même partie de l'environnement, leurs récompenses seront moins importantes que s'ils partent explorer l'environnement dans des directions différentes.

[BZLG03] propose de résoudre des DEC-MDP transition-independant. La fonction de récompense est représentée comme la résultante de récompenses additives et d'un terme supplémentaire correspondant aux dépendances entre les actions des agents. Il propose un algorithme permettant de trouver la politique optimale jointe pour deux agents. Cet algorithme consiste à déterminer dans un premier temps, l'ensemble des politiques optimales π_A de l'agent A pour n'importe quel politique de l'agent B (nommé Coverage Set) en utilisant les propriétés de la fonction de valeur. Ensuite, à chaque politique π_A fixée du 'Coverage Set', est associée la politique optimale de B $\pi^*_B(\pi_A)$ qu'il est possible de calculer facilement. La politique jointe optimale se trouve alors parmi l'ensemble des couples $\pi^*_A, \pi_B(\pi^*_A)$.

[BZL04] propose d'autres types de résolution fondés sur le même genre d'approche : dans les transition independant DECMDP, la seule influence possible d'un agent envers un autre agent résidait dans les fonctions de récompense. Dans les DEC-MPD 'with event-driven interaction', les transitions effectuées par un agent sont la résultante de transitions locales et de transitions dues aux événements effectués par les autres agents. Un agent peut ainsi faciliter les transitions des autres agents. L'algorithme repose sur le même principe que l'algorithme utilisé pour résoudre des DEC-MDP transition-independant : définir un ensemble de politiques optimales pour l'agent A pour toutes les politiques de l'agent B, associer à chacune de ces politiques, la politique optimale de B, chercher dans l'ensemble de ces politiques jointes, la politique jointe optimale.

Avantages En se focalisant sur des sous-problèmes particuliers des DEC-POMDP, ces travaux ont pu proposer les premiers algorithmes permettant de construire des politiques jointes de manière optimale à moindre coût.

Inconvénients Cette approche se heurte à l'explosion combinatoire du nombre d'états par rapport au nombre d'agents. Elle permet de construire des politiques optimales pour des groupes de deux agents en considérant des couples de politiques mais elle n'est plus envisageable dès que le nombre d'agents est supérieur à deux ce qui limite fortement son intérêt.

Par rapport à nos travaux Cette approche est extrêmement intéressante car elle utilise une structuration explicite du problème dû à la présence de plusieurs agents dans le système pour construire des résolutions à moindre coût. Elle isole ainsi les interactions possibles entre les agents et raisonne explicitement sur les couplages qui peuvent exister entre les comportements des agents.

A.1.6 MDP factorisés

Objectif	Recherche de politique sous-optimale
Contraintes	Observabilité partielle
Moyen	MDP factorisés
Modèle	Réseaux bayésiens

L'approche proposée par [Gue03] consiste à décrire le MDP sous la forme d'un réseau bayésien dynamique et à utiliser cette structure pour résoudre des MDP de grande taille de manière approchée en décomposant la fonction de valeur comme combinaison linéaire de fonction de valeurs élémentaire définies sur cette structure. Cette approche permet de construire des systèmes de manière décentralisée et sera présentée de manière plus détaillée dans la partie suivante.

A.2 Approches décentralisées

A.2.1 Utilisation de réseaux bayésiens

Objectif	Construire des politiques sous-optimales de manière peu coûteuse
Moyen	Exprimer la structure du MDP et en tirer parti pour construire des politiques sous-optimales
Contraintes	Observabilité partielle
Modèle	MDP factorisés et réseaux bayésiens dynamiques

Dans sa thèse, Guestrin [Gue03] propose de tirer parti d'une décomposition d'un MDP afin de construire à moindre coût une politique sous-optimale. Dans de nombreux problèmes, les états du systèmes correspondent à la concaténation de différentes caractéristiques définies dans le cadre du problème. De plus, ces caractéristiques n'ont pas forcément toutes des influences directes entre elles.

Afin de tirer parti de la structure du problème, Guestrin propose d'isoler ces caractéristiques pour obtenir une représentation factorisée de MDP inspirée des réseaux bayésiens dynamiques. Les états du système sont définis en fonction de variables X_i et la fonction de récompense et la matrice de transition peuvent s'exprimer de manière factorisée à partir des relations de dépendances entre ces variables.

[Gue03] propose d'utiliser des fonctions de valeur élémentaires définies sur des ensembles réduits de variables. Il s'intéresse particulièrement aux fonctions de valeurs $v_1, ...v_n$ définies sur les supports des relations de dépendance entre variables.

L'approche proposée par [Gue03] consiste alors à chercher une approximation de la fonction de valeur optimale définie sur l'espace d'état parmi les combinaisons linéaires de ces fonctions élémentaires $v(s) = \sum_i \alpha_i.v_i(s)$. En se plaçant dans ce sous-espace, la recherche de la fonction de valeur sous-optimale consiste à chercher les coefficients α_i.

Des techniques basées sur la programmation linéaire peuvent être utilisées pour effectuer cette recherche. En ayant choisi de décomposer la fonction de valeur dans l'espace défini par ces fonctions élémentaires, il est possible de représenter de manière compacte les équations d'optimalité de Bellman à résoudre. Enfin, en utilisant des techniques d'élimination de variables dans les réseaux bayésiens, la complexité de la résolution peut être réduite.

Cette approche parvient alors à trouver des solutions approchées (dont il est possible de borner l'erreur) dans des MDPs avec un nombre d'états très important [GKPV03]

Comme la résolution centralisée de systèmes multi-agents correspond à la résolution de MDP de grande taille et que les SMAs sont adaptés à une représentation factorisée, Guestrin propose d'utiliser ce même type de techniques pour effectuer de la planification [GKP01].

En tirant parti de la décomposition du problème et des techniques d'élimination de variables, il propose en outre des techniques d'apprentissage par renforcement distribuées [GLP02]. Ces techniques se fondent sur la présence du réseau de relation entre les variables modifiables par les agents. Ainsi il est possible, grâce aux techniques d'élimination de variables, de faire une maximisation globale à partir d'un transfert des fonctions de valeurs locales dans les réseau des relations inter-agents.

L'intérêt de ces approches résident dans le fait que la représentation factorisée du MDP correspondant explicite les relations entre les agents et les variables de l'environnement qu'ils peuvent modifier. De ce fait, les techniques d'élimination de variables peuvent être réalisées grâce à des communications locales entre agents.

Avantages Ces techniques permettent ainsi de limiter les communications entre agents aux communications utiles et de trouver la politique sous optimale à moindre coût à partir d'une structuration du problème tout en conservant des bornes concernant la qualité du comportement collectif construit.

Inconvénients Ces approches ne correspondent néanmoins pas entièrement à nos attentes :

- Les communications ne sont pas locales : bien que les canaux de communications utilisés soient toujours des canaux impliquant peu d'agents, il est supposé que les agents peuvent toujours échanger des informations et donc qu'il est toujours possible de reconstituer par communication l'approximation de la fonction de valeur globale
- Ces approches se heurtent au problème de l'expressivité des réseaux bayésiens dynamiques : le système est censé pouvoir se représenter sous la forme d'un réseau entre les différentes variables du système, en supposant cette représentation compacte. Cependant, si on s'intéresse à des systèmes complexes, une telle représentation est toujours possible mais n'est plus forcément compacte. Ainsi, si on s'intéresse à des système dans lesquels certaines actions (comme des échanges d'objets entre agents) nécessitent le respect de certaines distances entre les agents et des interactions impliquant des agents divers en fonction de leur position, ceci peut ajouter de nombreuses relations entre les noeuds du réseau bayésien puisque toutes les variables sont potentiellement en interaction les unes avec les autres.

Par rapport à nos travaux L'intérêt de ces travaux est de proposer une approche utilisant la structuration du système pour construire de manière distribuée les politiques des agents. Guestrin propose une approche consistant à partir d'un MDP factorisé pour lequel les interactions entre agents sont définies de manière explicite. Ceci permet alors de considérer les influences à long terme d'une action par un agent en évaluant l'ensemble des conséquences que peut avoir cette action sur le système, y compris sur les autres agents. Cependant, l'absence de localité dans les communications s'oppose au principe de localité que nous avons mis en avant.

A.2.2 Notifications réciproques

Objectif	Calcul de la politique optimale
Contraintes	environnement déterministe, observabilité totale, récompenses locales
Moyen	partager des gains locaux, communication entre agents
Modèle	MMDP

Description Les travaux de [Sze04] s'intéressent à des agents dotés de capacités de communication. Ils se concentrent sur des systèmes coopératifs pour lesquels chaque agent perçoit localement une partie de la récompense.

Au fur et à mesure de leurs apprentissages indépendants, les Q-valeurs des agents sont mises à jour : ils reçoivent de nouvelles récompenses et arrivent dans un nouvel état en fonction de leur action mais aussi de celles des autres agents. [Sze04] propose une approche basée sur la notion de notification : dés qu'un agent perçoit une augmentation locale de ses performances au cours de l'exploration, il en notifie les autres agents qui a leur tour communiquent à tous leurs gains locaux. Il est alors possible de déterminer si l'action jointe qui vient d'être émise est meilleure ou non pour le système dans sa globalité et si on décidera d'émettre à nouveau cette action dans les mêmes circonstances.

Avantage Une telle approche dispose de preuve de convergence vers la politique optimale tout en se limitant à des constructions de politiques entièrement décentralisées.

Inconvénients Par contre, l'état du système doit être perçu ce qui s'oppose à nos contraintes de localité et l'environnement doit être non stochastique pour être sur que les variation de récompense reçues proviennent effectivement des modifications des actions des agents et non pas de la présence d'une variable aléatoire. Les communications au cours de l'apprentissage sont non limitées dans l'espace (tous les agents peuvent communiquer ensemble), dans le temps (les communications sont émises à un taux constant même si on dispose de la politique jointe optimale) même si celles-ci disparaissent à l'execution du système une fois la politique jointe apprise.

A.2.3 Empathie

Objectif	Construire les politiques sous-optimales
contrainte	Récompense globale, Perceptions partielles
Moyen	Communication implicite permettant de connaître les politiques des autres agents, processus de synchronisation global
Modèle	DEC-POMDP

Un des problèmes rencontrés dans les systèmes multi-agents provient du fait qu'un agent doit connaître les comportements des autres agents et en être sûr pour pouvoir décider de la meilleure action à entreprendre. Une solution possible consiste à fixer les comportements des autres agents pendant qu'un agent planifie ses actions.

Les travaux présentés dans [Cha02] utilisent cette idée et se fondent sur la notion d'empathie. Chades propose de doter les agents de capacités de communication leur permettant de s'échanger leur politique. Chaque agent peut alors prendre en compte le comportement des autres agents dans sa prise de décision. De plus, si les comportements des autres agents restent fixes, un agent peut facilement planifier ses actions puisque les hypothèses de convergence des algorithmes classiques sont respectées étant donné que l'environnement perçu par un agent est alors stationnaire.

L'algorithme de co-evolution qui a été proposé dans [Cha02] consiste à choisir un sous-groupe d'agents et à fixer les politiques des autres agents. Il consiste ensuite à résoudre un MMDP pour trouver les politiques optimales du sous-groupe d'agents sélectionné. A l'itération suivante, un autre groupe d'agents est choisi et leur politique optimale calculée. Et ainsi de suite jusqu'à ce qu'aucun agent ne modifient leur politique au cours d'une itération. A chaque itération, les récompenses reçues par l'ensemble des agents augmentent puisque la phase de planification trouve au pire les politiques dèjà possédée par les agents au début de l'itération, ce qui prouve la convergence de l'algorithme.

Avantages L'avantage d'une telle approche est de disposer d'un algorithme de construction entièrement décentralisé et basé sur des planifications individuelles peu coûteuses pour construire

des comportements permettant de résoudre un problème collectif.

Inconvénient Cependant, cet algorithme converge vers des équilibres de Nash. Il devient alors nécessaire de choisir le bon sous-groupe d'agents à faire évoluer pour résoudre cet équilibre, ce qui peut éventuellement en provoquer d'autres. De plus, ces approches nécessitent la communication des politiques entières entre agents et utilise un processus de synchronisation consistant à autoriser ou bloquer les phases de planification chez les agents, ce qui se heurte à notre principe de localité.

Par rapport à nos travaux Cette approche nous semble extrêmement intéressante car elle se fonde sur l'idée que des planifications individuelles peuvent permettre de construire des comportements collectifs. Néanmoins, le fait de synchroniser ces planifications, bien que permettant d'avoir des preuves de convergence des politiques construites, ne conduit qu'à des sous-optimaux et se heurte aux systèmes qui doivent utiliser des interactions complexes entre agents, comme l'émission d'actions synchronisées.

A.2.4 COIN

Objectif	Répondre au problème de la tragédie des communs
Contrainte	Perception partielles, Récompenses partielles
Moyen	Communication implicite permettant de connaître les politiques des autres agents, processus de synchronisation globale.

[TW00] s'intéressent à la tragédie des communs et tentent de proposer une solution à des problèmes proches. Ils essaient de trouver comment à partir d'un problème donné il est possible de définir les fonctions de valeur locales des agents afin de s'assurer que de bons comportements individuelles induisent un bon comportement collectif.

Pour cela, ils définissent un cadre formel afin de pouvoir déterminer si des fonctions individuelles sont "alignées" avec la fonction de performance collective. Ils définissent ainsi la notion de "factored systems" pour lesquels les changements décidés par un agent ne peuvent que faire augmenter la fonction de performance globale. Dans ce cas, le comportement global optimal correspond à un équilibre de nash de fonction d'utilité locales et il est possible de construire la solution globale optimale à partir d'apprentissages individuels simples.

Inconvenients Malheureusement, dans un système multi-agents complexe, il n'est pas possible de faire de factorisation : les actions des agents s'influencent mutuellement et ce sont ces influences qui permettent de résoudre correctement la tâche globale ce qui réduit considérablement les champs d'applications de l'approche COIN

A.2.5 Fonctions de valeurs distribuées

Objectif	Trouver une politique sous-optimale
Manière	Apprentissages décentralisés
Moyen	Introduire des communications entre agents pour échanger des fonctions de valeurs
Modèle	DEC-POMDP factorisé avec récompenses locales

Description [SWMR99] présente une approche pour effectuer de l'apprentissage par renforcement décentralisé (Decentralized Reinforcement Learning). Le point de départ de ces travaux consiste à s'intéresser tout d'abord à des apprentissages entièrement distribués. Chaque agent dispose de ses Q-valeurs et essaie d'apprendre une fonction de valeur comme s'il était seul dans l'environnement.

$$V_i(x) = max_{a \in A_i}(R_i(x,a) + \gamma \Sigma_{x' \in X} p(x'|a,x) V(x'))$$

Afin de prendre en compte les récompenses reçues par les autres agents, Schneider et al définissent une topologie et ajoutent un terme supplémentaire à cette équation : les récompenses pondérées (par un facteur $f(i,j)$) que transmettent les agents voisins à chaque agent.

$$V'_i(x) = max_{a \in A_i}(\Sigma_j f(i,j) R_j(x,a_j) + \gamma \Sigma_{x' \in X} p(x'|a,x) V'_i(x'))$$

.

Désormais, lorsqu'un agent apprend à maximiser cette nouvelle fonction V', il prend en compte de manière implicite les récompenses reçues par les autres agents et aura tendance à adopter un comportement utile pour les agents voisins. Simplement, la prise en compte des récompenses reste limitée aux voisins immédiats. L'approche proposée consiste alors à échanger non pas les récompenses mais les fonctions de valeurs.

$$V''_i(x) = max_{a \in A_i}(R_i(x,a_i) + \gamma \Sigma_j f(i,j) \Sigma_{x' \in X} p(x'|a,x) V''_i(x'))$$

Il est à noter que cette fonction V'' n'a plus le sens ni les propriétés des fonctions de valeur classique puisque elle ne vérifie plus l'équation de Bellman initiale.

Cette heuristique permet aux agents de prendre en compte la satisfaction des agents voisins qui intègrent à leur tour la satisfaction des voisins de rang supérieur. De cette manière, un agent considère de manière implicite l'ensemble des agents du système au moment de prendre sa décision. Appliquée à des problèmes de distribution, une telle approche fournit de bons résultats en se limitant à des communications entre agents voisins. Des perspectives sont envisagées concernant l'adaptation des paramètres $f(i,j)$ à l'exécution du système.

Avantages Cette approche présente l'avantage de construire des politiques collectives sous-optimales à partir d'apprentissages entièrement individuels et décentralisés. Elle se fonde sur l'idée qu'un simple échange des fonctions de valeur entre agents suffit pour qu'un agent puisse considérer la présence d'autres agents dans le système et leurs capacités par rapport à la tâche à résoudre. Cette approche constitue donc un moyen simple de doter les agents de compétences sociales au cours de leur apprentissage.

Inconvénients Les communications sont statiques et impliquent toujours les mêmes ensembles d'agents, or les interactions possibles entre les composants du systèmes doivent pouvoir se reconfigurer. De plus, il est supposé qu'il est toujours possible pour les agents de communiquer entre eux ce qui se heurte à nos contraintes de localité.

Par rapport à nos travaux L'approche employée par Schneider et al propose un moyen simple et économique d'intégrer une composante sociale au niveau de l'agent pour construire des comportements collectifs à partir d'apprentissages décentralisés. L'échange de fonctions de valeur est un moyen permettant aux agents d'adopter une attitude coopérative les uns avec les autres et de rendre compte des interactions à long terme entre les composants du système. Comme nous le verrons par la suite, l'approche que nous proposerons reposera sur le même postulat.

A.2.6 MDP faiblement couplés

L'article [MHK+98] porte sur la résolution de MDP faiblement couplé. Néanmoins, l'approche proposée peut être interprétée d'un point de vue multi-agents comme en témoigne l'exemple présenté et est trés proche du formalisme que nous développerons par la suite. Nous nous permettons donc de la représenter dans cette partie.

Objectif	L'objectif est de construire une politique sous-optimale dans des problèmes à contraintes de ressources
Contrainte	Perception partielles, récompenses locales
Manière	Résolution décentralisée
Moyen	Communication implicite pour réajuster les politiques locales au sein du groupe.

[MHK+98] s'intéresse à un problème d'allocation de ressources entre plusieurs tâches. A chaque tâche est associée une fonction de performance indépendante et la fonction de performance globale du système est définie comme la somme des fonctions de performance locales. En outre, le MDP se décompose en états indépendants propres à chaque tâche. La résolution de chaque tâche peut ainsi être vue comme un sous-processus indépendant des autres (mais dépendant des ressources attribuées).

Interprétation multi-agents possible : Chaque tâche peut être vue comme assignée à un agent. Cet agent accède à la fonction de performance locale et cherche en fonction des ressources qui lui sont attribuées à maximiser cette fonction.

Résoudre un tel problème est non trivial puisque :
- les politiques locales dépendent des ressources attribuées
- les ressources attribuées dépendent des politiques locales qui en tirent parti et des perfomances associées.

L'exemple proposé est un problème de distribution d'armement dans une flotte aérienne pour laquelle chaque avion a une cible fixée a priori. Les contraintes sont de deux ordres : des contraintes globales concernant le stock global d'armement et des contraintes locales liées à la capacité de chaque avion.

[MHK+98] propose de calculer les politiques locales individuellement (comme pourrait le faire chaque agent de manière décentralisée si on garde à l'esprit notre interprétation) et d'utiliser des heuristiques pour distribuer les ressources entre les tâches :

- Tout d'abord en cherchant à maximiser la somme des performances locales attendues tout en respectant la contrainte globale. Chaque ressource est attribué de manière itérative en fonction de son utilité marginale, c'est à dire du gain que peut apporter cette ressource pour la performance de la tâche considérée. Cette ressource est attribué à la tâche par laquelle le gain marginal est le plus important.
- En vérifiant ensuite si les contraintes locales sont respectées et en redistribuant les excès lorsqu'une contrainte locale n'est pas respectée.

Avantage Chaque agent détermine sa politique de manière individuelle entièrement décentralisée et une phase de centralisation fondée sur une heuristique permet d'ajuster les ressources attribuées entre agents.

Inconvénient Il y a échange d'information entre toutes les 'entités du système' pour déterminer comment distribuer les ressources.

Par rapport à nos travaux Cette approche se fonde sur l'explicitation des interactions entre les composants du système : le système est décrit sous la forme de processus locaux chacun associé à une tâche particulière et les interactions possibles entre ces processus résident dans l'attribution de ressources communes. En structurant le système de cette manière, il est possible de construire des réponses collectives (attribution des ressources et comportement associé à chaque tâche) à partir de résolutions locales et d'heuristiques permettant de décider de l'attribution des ressources en fonction des performances locales qui ont pu être calculées.

A.2.7 Apprentissage incrémental

Objectif	L'objectif est de construire les politiques sous-optimales
Contraintes	Perceptions partielles, Récompense globale
Manière	Résolution décentralisée, Exécution décentralisée
Moyen	guider les agents et proposer des situations de plus en plus complexes aux agents.

Cette approche proposée par Buffet [Buf03] cherche à construire de manière décentralisée des politiques individuelles d'un ensemble d'agents dotés d'observations partielles. Il existe deux types d'agents et leur tâche consiste à fusionner des blocs de couleurs différentes (jaune et bleu). Chaque agent perçoit localement le bloc jaune et le bloc bleu les plus proches ainsi que l'agent de l'autre couleur. La précision des perceptions dépend de la distance de l'agent aux autres objets : si l'agent est à une case de l'objet, il peut percevoir sa position exacte. Si celui-ci est plus éloigné, il en perçoit la direction. Chaque agent reçoit une récompense lorsqu'il participe à la fusion de bloc.

Bien qu'il ne soit pas possible de générer par apprentissage les politiques optimales des agents, l'apprentissage incrémental cherche à construire "de bonnes politiques". Il consiste à proposer à deux agents des situations de plus en plus complexes et à leur faire apprendre ces situations à

partir de leurs apprentissages précédents. Les agents sont guidés par le concepteur de l'application et parviennent :

- à se coordonner à partir des signaux de récompenses reçus et des perceptions partielles
- à construire des politiques permettant de résoudre des situations élémentaires

Ces solutions possèdent en outre de propriétés de passage à l'échelle. Une fois les politiques individuelles apprises, on peut disposer dans l'environnement un certain nombre de cubes et d'agents et parvenir à des fusions de blocs. Le fait de disposer de politiques stochastiques et de perceptions partielles constitue un moyen de généralisation des comportements des agents.

Inconvénients Cependant, la présence d'une entité extérieure est nécessaire pour guider les apprentissages en donnant des situations de plus en plus complexes. En outre, la récompense du système est globale et observée par l'ensemble des agents.

Par rapport à nos travaux Cette approche se fonde sur un apprentissage des interactions possibles entre agents. Elle n'utilise pas de structuration a priori du système mais permet aux agents d'apprendre à résoudre les situations d'interactions en leurs présentant des situations de plus en plus complexes. Un apprentissage simultanée (contrairement à [Cha02]) permet de construire des solution collectives du fait de la présence d'un signal de récompense global qui renforce les actions jointes utiles qui ont pu être émises.

Annexe B

Exécution de quelques interac-DEC-POMDP

B.1 Description

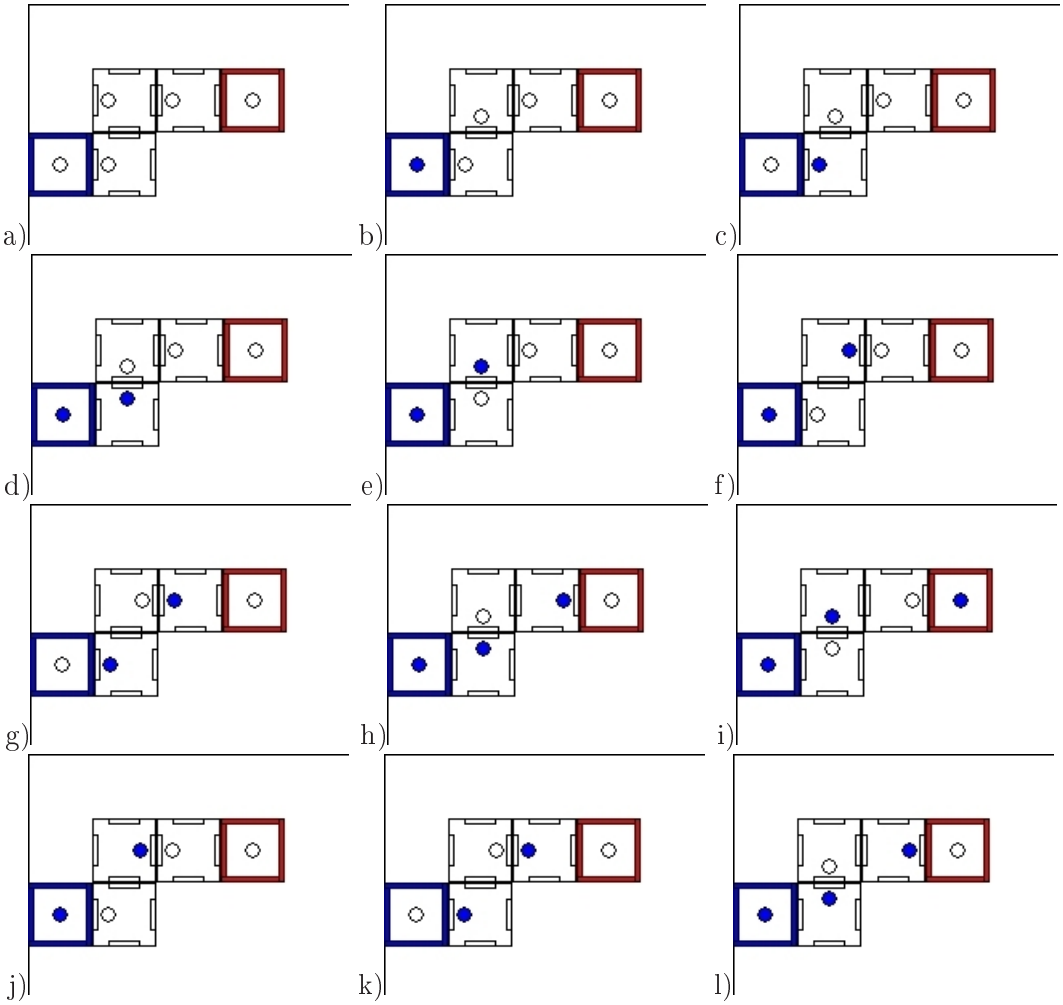
Cette annexe contient les résultats obtenus par exécution des politiques construites par apprentissage.

Chaque image présentée correspond à une situation avant l'exécution d'un module donné (d'action ou d'interaction). Ainsi l'image a) correspond à l'état initial, l'image b) à l'état après exécution des actions, l'image c) à l'état après exécution de l'ensemble des interactions, l'image d) à l'état après exécution des actions et ainsi de suite.

Les salles du système sont représentées par des cubes et les interactions possibles par des portes. Les agents sont représentés dans ces salles par un cercle.

- le type d'agent est représenté par la couleur des murs de la pièce : si ceux ci sont bleus, il s'agit d'un agent ravitailleur capable d'accéder à un puit, s'ils sont rouge, il s'agit d'un agent extincteur enfin, s'ils sont blanc, cela signifie qu'il s'agit d'un agent couloir
- lorsque le cercle est plein, cela signifie que son seau est rempli. Au contraire lorsque le cercle est vide, celui signifie que le seau est vide.
- la position des agents couloir est représentée par la position du cercle dans la pièce

B.2 Chaîne constituée de 5 agents



B.3 Système constitué de 24 agents

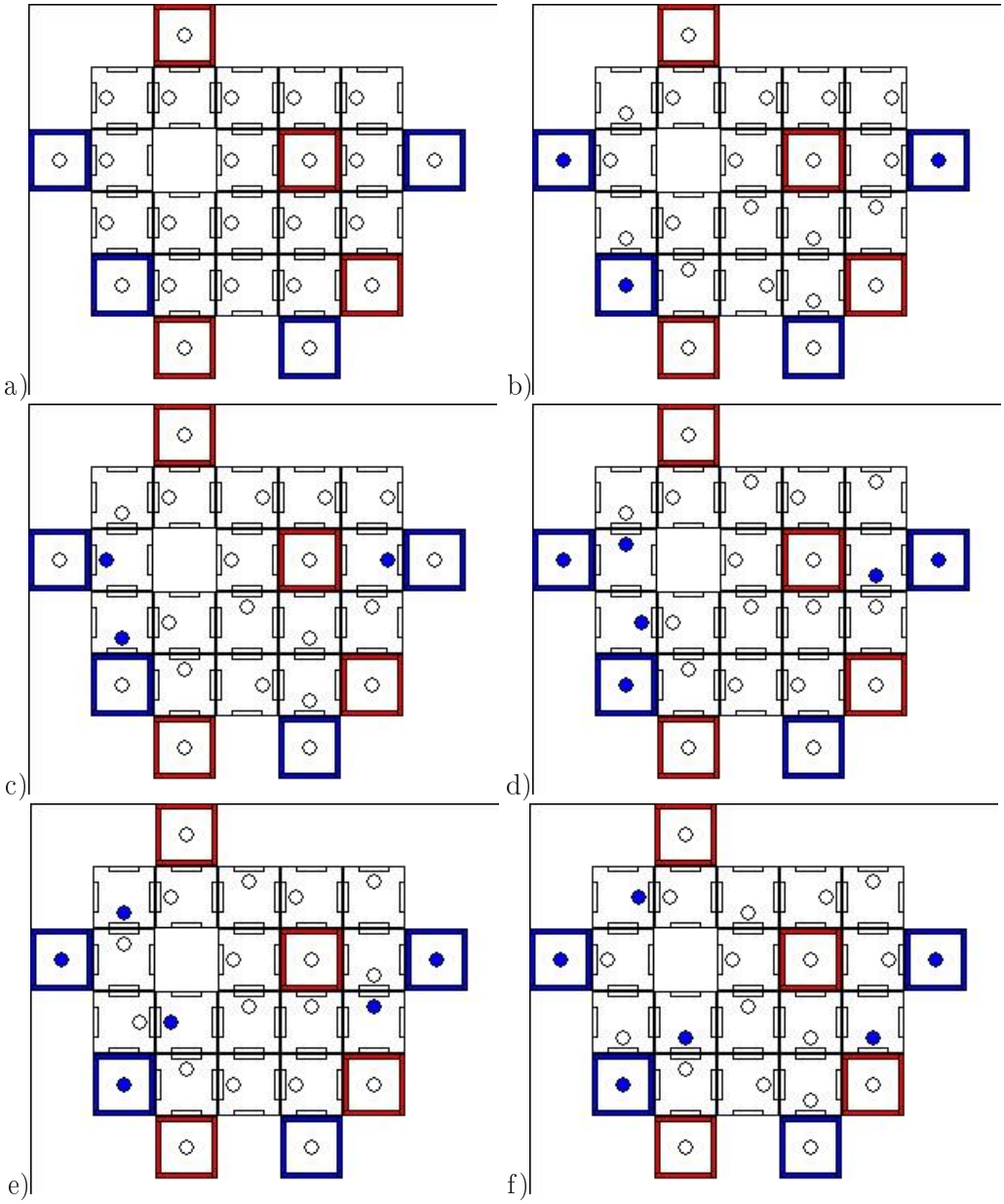


FIG. B.1 – Execution Partie I

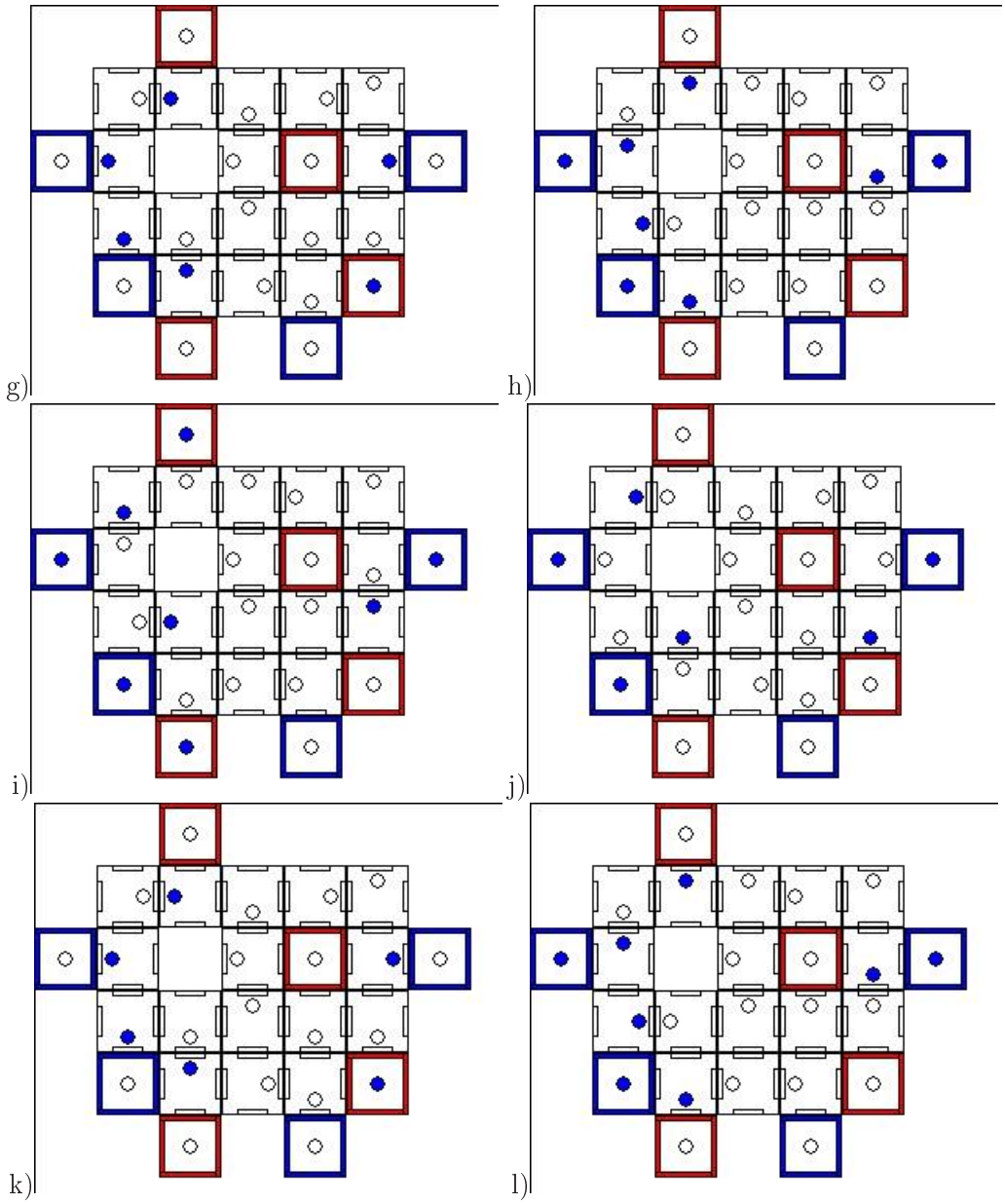


FIG. B.2 – Execution Partie II

Résumé

Cette thèse traite de la conception de système multi-agents. Elle se focalise sur des approches formelles et s'est donné pour objectif à long terme de construire de manière automatique et décentralisée les comportements d'agents coopératifs devant résoudre collectivement un problème. Ce travail a cherché à proposer des méthodes pour construire les comportements d'agents sociaux, capables de prendre en compte à l'exécution la présence d'autres agents dans le système.

Les formalismes existants comme les DEC-POMDPs parviennent à représenter des problèmes multi-agents mais ne représentent pas au niveau individuel la notion d'interaction fondamentale dans les systèmes collectifs. Ceci induit une complexité algorithmique importante dans les algorithmes de résolution. Afin de donner aux agents la possibilité d'appréhender la présence d'autres agents et de structurer de manière implicite les systèmes multi-agents, cette thèse propose un formalisme original, l'interac-DEC-POMDP inspiré des DEC-POMDPs et d'Hamelin, une simulation développée au cours de cette thèse et issue d'expériences conduites en éthologie. La spécificité de ce formalisme réside dans la capacité offerte aux agents d'interagir directement et localement entre eux. Cette possibilité permet des prises de décision à un niveau intermédiaire entre des décisions globales impliquant l'ensemble des agents et des décisions purement individuelles.

Nous avons proposé en outre un algorithme décentralisé basé sur des techniques d'apprentissage par renforcement et une répartition heuristique des gains des agents au cours des interactions. Une démarche expérimentale nous a permis de valider sa capacité à produire pour des restriction du formalisme des comportements collectifs pertinents adaptatifs sans qu'aucun agent ne dispose d'une vue globale du système.

Mots-clés: Système multi-agents, Interaction , Processus décisionnel de Markov, Apprentissage par renforcement, inspiration biologique

Abstract

This thesis deals with the design of multi-agent systems. It focuses on formalism based approach and aims in the long run to build, automatically and in a decentralized way, the behaviours of cooperative agents which must solve a collective problem. The goal of this work was to propose new techniques to build the behaviour of social agents, able to consider the presence of other agents in the system.

Existing formalism like DEC-POMDPs manage to formalize multi-agents problem but they don't represent at the agent level the concept of interaction which is fundamental in collective systems. It induces a important complexity in the algorithms used to build the behaviours of the agents. In order to give the agent the ability to consider the presence of other agents in the system and to structure implicitely multi-agents systems, this thesis proposes an original formalism the Interac-DEC-POMDP inspired by the DEC-POMDP formalism and Hamelin, a simulation developped during this thesis and inspired by collective biological phenomenon. The specificity of this new formalism lies in the ability given to the agents to interact directly and

locally among them. It allows them to make decision at a level between global level and individual level.

Furthermore, we have proposed a decentralized algorithm based on reinforcement learning techniques and on distribution of individual rewards among agents during interactions. We have conducted experiments and validated our proposal : this algorithm manage to produce adaptive collective behaviour without the need for the agents to have a global vision of the system.

Keywords: Multi-agent system, Interaction, Markov Decision Process, Reinforcement learning, Biological inspiration