Contrôle d'un Système Multi-Agents Réactif par Modélisation et Apprentissage de sa Dynamique Globale

François Klein

Contrôle d'un SMA Réactif par Modélisation et Apprentissage de sa Dynamique Globale

THÈSE

présentée et soutenue publiquement le [date]

pour l'obtention du

Doctorat de l'université Nancy 2

(spécialité informatique)

par

François KLEIN

Composition du jury

Rapporteurs : Marie-Pierre GLEIZES, Professeur, Université Paul Sabatier – Toulouse 3.
Salima HASSAS, Professeur, Université Claude Bernard – Lyon 1.

Examinateurs : Abderrafiâa KOUKAM, Professeur, Université de Technologies de Belfort-Montbéliard.
Anne BOYER, Professeur, Université Nancy 2.
Vincent CHEVRIER, Maître de conférences HDR, Université Henri Poincaré – 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

Table des matières

TABLE DES ILLUSTRATIONS

INDEX DES TABLEAUX

Remerciements

Résumé

Dans un système multi-agent (SMA) réactif, le lien entre le comportement collectif et celui des individus qui composent ce système est difficile à établir. Obtenir un comportement particulier est donc également difficile.

Nous défendons le principe de maîtriser le comportement d'un SMA par une approche de contrôle. Pour cela, nous agissons sur le SMA à partir d'informations relatives à ses comportements globaux.

Pour y parvenir, nous proposons tout d'abord de modéliser la dynamique globale du SMA sous forme d'un graphe d'états. Des outils d'apprentissage par renforcement permettent de construire ce graphe et de calculer une politique qui indique quelle action effectuer en fonction de l'état courant et d'un comportement cible à atteindre. Ensuite, cette politique est exploitée pour contrôler le SMA.

L'originalité de notre proposition est de s'appuyer sur la dynamique du SMA décrite à son niveau global. Ainsi, les différents comportements du SMA sont exprimés dans notre proposition au même niveau de description que celui du comportement à atteindre.

La proposition est appliquée au contrôle d'un SMA inspiré du déplacement de piétons dans un couloir. Nous la comparons à d'autres approches destinées à maîtriser le comportement d'un SMA.

Nous vérifions que le principe du contrôle au niveau global fonctionne. Nous montrons que notre proposition fournit de bonnes performances de contrôle et permet d'atteindre un comportement cible plus fréquemment que les autres approches testées.

Nous posons ainsi les premières pierres d'un cadre paradigmatique pour le contrôle au niveau global des systèmes multi-agents.

Mots-clefs : Système multi-agents, contrôle, comportement global, apprentissage, processus de décision markovien.

Abstract

In a reactive multi-agent system (MAS), the link between the collective behaviour and the behaviours of the individuals who make up this system is difficult to set up. So to reach a particular behaviour is also difficult.

We support the concept of driving the behaviour of a MAS by a control approach. In order to obtain this control, we act on the MAS by using information about its global behavior.

To achieve this, we first propose to model the global dynamics of the MAS as a graph of states. Reinforcement learning tools help to build the graph and to compute a policy. This policy indicates which action to choose based on the current state and a target behaviour. Then this policy is used to control the MAS.

The originality of our proposal lies in the global level description of the MAS's dynamics. Thus the different behaviours of the MAS are expressed, in our proposal, at the same description level as the one of the target behaviour.

The proposal is applied to control a MAS based on the movement of pedestrians in a corridor. It is compared to other approaches whose goal is to drive the behaviour of a MAS.

We verify that the concept of global level control works. We show that our proposal provides good control performance and achieves a target behaviour more frequently than other tested approaches.

Thus we lay the foundations of a new framework for the global level control of multi-agent systems.

Keywords : Multi-agent system, control, global behaviour, learning, Markov decision process.

CHAPITRE I - INTRODUCTION

« Une bonne partie du travail de la science est consacrée à la détection des permanences sous-jacentes aux changements apparents, à la mise en évidence du constant sous le variable. »

« Souvent, le système est trop complexe, par exemple, s'il comprend un grand nombre de particules [...], pour que l'on puisse suivre en détail son évolution. Mais un nouveau type de constance apparaît souvent au coeur de la variabilité, en aval cette fois, si l'on peut dire, et non plus en amont comme dans les équations du mouvement elles-mêmes. Ces "constantes du mouvement" sont des grandeurs collectives impliquant les diverses grandeurs d'état particulières, qui ont la propriété de ne pas changer de valeur numérique, alors même que les grandeurs particulières varient. »

« Dès que l'on a affaire à des systèmes tant soit peu élaborés, [...] l'incertitude sur l'état du système croît exponentiellement avec le temps, de façon que toute possibilité de prévision sur son comportement, même qualitative, disparaît rapidement. » - Jean-Marc Lévy-Leblond [Lévy-Leblond 96]

La plupart des systèmes naturels – biologiques, physiques ou sociologiques – sont formés de nombreux composants élémentaires en interaction. Ils présentent un comportement global, comme la température d'un gaz ou des mouvements de foule, collectif à la majeure partie des composants du système. Il peut être identifié par un observateur extérieur. On a souvent recours à des modèles pour expliquer ce comportement à partir des propriétés élémentaires des composants du système.

Mais il existe des systèmes dont le comportement est qualifié de complexe. Dans ce cas, aucune simplification ne permet de rendre compte de l'origine de ce comportement, qui échappe à la compréhension humaine.

Les systèmes multi-agents (SMA) partagent avec ces systèmes naturels la propriété de voir un comportement global émerger des interactions entre composants, sans qu'il soit possible de l'expliquer facilement. Les SMA forment un outil qui permet de simuler ces systèmes, non pas au niveau de leur comportement global, sous forme d'équations par exemple, mais au niveau de leurs composants élémentaires : on dit qu'il s'agit d'une approche individu-centrée. Ils peuvent également être construits pour assurer une tâche en présentant un comportement souhaité.

L'une des difficultés liées aux SMA est d'assurer un comportement particulier au système, car ce comportement est difficilement expliqué. Le but de ce travail est de mieux comprendre et de diriger le comportement d'un SMA, en particulier lorsqu'il n'est pas possible d'assurer un comportement souhaité lors de la construction du système.

Nous défendons le principe de maîtriser le comportement d'un SMA en le contrôlant à son niveau global. Pour cela, nous agissons sur le SMA à partir d'informations sur ses comportements globaux. Pour y parvenir, nous proposons :

- De modéliser la dynamique globale du SMA par un processus de décision Markovien.
- D'effectuer le contrôle en considérant les différents comportements du système comme les états du modèle markovien. Ainsi, seul le comportement courant est pris en compte pour agir sur le système.
- D'automatiser l'apprentissage d'une politique de contrôle, en se servant d'outils d'étude des processus de décision Markovien. Le nombre limité de comportements différents du SMA, donc d'états du modèle, doit permettre de réduire la durée de cet apprentissage.

Pour appuyer ces propositions, nous montrons que ce contrôle offre un cadre riche pour améliorer la capacité du SMA à atteindre un comportement désiré, avec un temps d'apprentissage acceptable.

Notre objectif est d'évaluer à la fois cette solution de contrôle particulière et le principe même du contrôle global d'un SMA. Nous étudions l'utilité d'une approche de contrôle en fonction du contexte. Nous posons ainsi les premières pierres d'un cadre paradigmatique pour le contrôle au niveau global des systèmes multi-agents.

Le présent manuscrit est organisé en quatre parties principales. En voici une description succincte :

Chapitre II : Dans cette partie, nous nous attachons d'abord à définir ce que sont les systèmes multi-agents et la problématique de maîtriser leur comportement global. A la lumière d'autres travaux traitant de cette problématique, nous montrons sa difficulté. La classification de ces différentes approches permet d'identifier des tendances pour répondre à la problématique. L'une d'entre elles est l'objectif de la thèse : maîtriser le comportement du SMA par un contrôle au niveau global.

Chapitre III : Nous expliquons dans cette partie ce qu'est le contrôle au niveau global d'un SMA. Nous proposons de construire un modèle markovien de sa dynamique globale, en assimilant les états du modèle aux comportements globaux du système. Nous montrons les différentes décisions à prendre pour y parvenir. Elles sont réparties en étapes, et un ensemble de pistes pour y répondre est proposé.

Chapitre IV : Nous entrons dans les détails de la proposition en montrant comment l'appliquer concrètement sur un SMA particulier, qui modélise des piétons dans un couloir. Notre objectif est de présenter des problèmes de contrôle qui se posent sur ce système, ainsi que les premiers choix effectués lors des étapes de la proposition.

Chapitre V : La proposition est validée de façon expérimentale. Nous montrons sur plusieurs exemples que les outils et les choix proposés permettent mieux d'assurer un comportement global du SMA que d'autres approches. Nous vérifions aussi l'importance des étapes de la proposition, ainsi que l'existence de régularités dans le SMA qui permettent à la politique apprise d'être robuste.

Chapitre II - Maîtriser le comportement d'un système multi-agents

Dans cette partie, nous nous attachons d'abord à définir ce que sont les systèmes multi-agents et leur évolution, le concept d'émergence dans un tel système, et celui de comportement global. Nous identifions les problèmes fondamentaux soulevés lorsqu'un utilisateur cherche à maîtriser ce comportement. Pour mettre en évidence ces difficultés, des travaux qui s'apparentent à cette question sont exposés. Nous identifions deux catégories d'approches.

La première se situe à la conception du système et permet de construire un SMA qui présente un comportement désiré. S'il existe une incertitude sur l'évolution de ce SMA, même minime, elle risque d'être amplifiée jusqu'à modifier le comportement de l'ensemble du système. Ces approches sont donc insuffisantes pour assurer le comportement global d'un SMA dans le cas général.

La seconde catégorie se concentre sur le contrôle du comportement global lors de l'évolution du SMA. Ces approches amènent le système à passer d'un comportement non souhaité à un comportement souhaité, appelé cible.

A. Définitions autour des SMA

Le but de ce chapitre est de donner une définition des objets étudiés dans cette thèse, les systèmes multi-agents. Un premier sous-chapitre s'attache à les présenter de façon générale, et à définir leur évolution, ainsi que les caractéristiques d'ouverture et de décentralisation sur lesquelles nous reviendrons à la fin du document. L'apparition d'un comportement propre au système à son niveau global fait l'objet d'un second sous-chapitre, et en particulier la difficulté d'appréhender ce comportement par rapport au niveau de définition local du SMA.

A.1 Les systèmes multi-agents

Un SMA est un système composé d'entités informatiques, appelées des *agents*, qui évoluent et interagissent dans un *environnement* commun. La notion d'interaction entre agents est essentielle car chacun d'eux est impliqué dans une dynamique commune, au lieu d'évoluer parallèlement et indépendamment aux autres.

Nous nous intéressons particulièrement à des systèmes appelés *réactifs*, et les définitions que nous proposons sont orientées vers ce type de SMA, bien que notre travail puisse être adapté à d'autres systèmes multi-agents.

[Demazeau 95] propose une décomposition d'un SMA en quatre dimensions qui correspondent aux quatre voyelles A, E, I et O, et qui est développée dans [Demazeau 01] :

- Agent : définition des modèles ou des architectures des composants du système.
- Environnement : milieu dans lequel sont plongés les agents, composé d'objets qui sont perçus et manipulés par les agents, et qui obéit à des lois physiques.
- Interactions : ensemble des infrastructures, langages et protocoles d'interaction entre agents.
- Organisation : structure des agents en groupes, hiérarchies, relations, etc.

On y trouve souvent associé une cinquième dimension *Utilisateur* qui représente un humain extérieur au système mais qui possède une influence sur lui et peut l'observer. Ces cinq notions sont à la base de notre réflexion, nous nous en servons pour définir plus en détails certaines notions propres aux SMA.

Les SMA peuvent être différenciés en trois familles d'application ([Boissier 04]) :

- la simulation de systèmes, par exemple biologiques ([Thomas 02]) ou sociaux ([Amblard 03], [Gaud 08]),
- la résolution de problèmes ([Ferber 89, Picard 06, Gleizes 04]),
- l'intégration de l'informatique avec les êtres humains et les systèmes mécaniques, c'est-à-dire des SMA ayant une raison d'être propre (réseaux P2P [Siebert 08], systèmes multi-robots [Simonin 02]).

Les caractéristiques communes aux SMA sont plus ou moins importantes en fonction de la famille à laquelle le système étudié appartient.

A.1.1 Comportement d'un agent informatique

L'une des définitions d'un agent qui fait consensus dans la communauté des SMA est donnée par [Ferber 95] et [Ferber 06]. Un agent est une entité informatique qui possède un comportement individuel, caractérisé principalement par quatre propriétés :

- Autonomie ou proactivité : capacité à agir sans intervention extérieure, prise d'initiative.
- Sensibilité : capacité à percevoir l'environnement ou les autres agents.
- Localité : limitation de la perception et des actions.
- Flexibilité : réaction aux changements perçus.

Nous verrons plus loin que l'ensemble des comportements individuels contribue à définir la dynamique d'un SMA. La caractéristique de réactivité présentée ci-dessous se répercute sur l'ensemble du système, et la notion de comportement individuel est nécessaire pour définir la dynamique du SMA.

A.1.1.1 Agents réactifs et cognitifs

Toujours d'après [Ferber 95], un agent possède une sociabilité plus ou moins importante, qui représente sa capacité à interagir avec d'autres agents ou avec un utilisateur humain. On dit qu'un agent est *réactif* s'il n'a qu'une faible capacité de communication et une représentation interne sommaire du système auquel il appartient : il ne possède que peu ou pas de modèle de lui-même, des autres agents ou de l'environnement. Un agent réactif a donc un comportement de type stimulus-réponse face à ce qu'il perçoit. Ce type d'agents est typiquement utilisé pour la simulation de systèmes et la résolution de problèmes.

Un agent *cognitif* possède une capacité de mémoire, de raisonnement ou de communication importante. On le rencontre principalement dans la troisième famille d'application, celle des systèmes physiques ou intégrés. La frontière entre les deux catégories d'agents est floue dans la mesure où il existe de nombreuses nuances d'agents, et l'on rencontre parfois des catégories intermédiaires.

A.1.1.2 Formalisation du comportement individuel

La notion de *comportement individuel* est formalisée de manière à la fois simple et générale dans [Wooldridge 02]. Il s'agit des règles qui déterminent l'action que cet agent effectue en fonction de ce qu'il perçoit. Les quatre propriétés du comportement individuel se trouvent dans la définition de ces règles. Par exemple, la localité des agents se traduit par une limitation de la portée des règles de comportement, à la fois en amont pour la perception et en aval pour la capacité d'action.

Le comportement d'un agent peut être défini soit en extension, lorsque l'action qu'il effectue est donnée explicitement pour chaque situation qu'il peut rencontrer, soit en intention, s'il s'agit d'une fonction analytique qui fournit une action par un calcul sur l'état de l'agent et de son environnement.

A.1.2 Environnement d'un SMA

Avec celle des comportements individuels, la caractérisation de l'environnement permet de définir la dynamique d'un SMA.

Un rapport très complet sur l'utilité de l'environnement dans un SMA peut être trouvé dans [Weyns 04]. Selon lui, les différents rôles de l'environnement sont de permettre la communication entre agents, d'être le support des actions des agents en définissant les règles et en renforçant ces actions, d'être observable par les agents, et enfin de prendre en charge l'activité propre des objets et des ressources présents en son sein.

Selon [Boissier 04], un système multi-agents possède un environnement dans lequel plusieurs agents évoluent, communiquent, perçoivent et agissent. Il peut être essentiel au système, par exemple si les agents se déplacent spatialement en son sein – on dit alors que les agents sont *situés* – ou s'il possède une dynamique propre, par exemple s'il contient des objets qui évoluent selon des lois physiques, indépendamment des agents. Il peut au contraire être très peu présent, jusqu'à se limiter aux messages échangés par des agents cognitifs.

Lorsque les agents sont réactifs, l'environnement détient une importance capitale car il est le médiateur de leurs interactions. En effet, comme ces agents ne peuvent communiquer directement entre eux, ils s'influencent mutuellement soit par leur position s'ils sont situés, soit par l'intermédiaire d'objets qu'ils perçoivent et modifient.

Dans les systèmes destinés à la résolution de problèmes, l'environnement possède un rôle supplémentaire : il définit souvent le problème à résoudre. Par exemple, dans la résolution du problème du voyageur de commerce par un algorithme fourmis [Dorigo 92], l'environnement est défini par les sites à visiter, donc par l'instance de problème.

A.1.3 SMA réactifs et cognitifs

Un SMA composé d'agents réactifs (respectivement cognitifs) est lui-même appelé réactif (resp. cognitif). Un système réactif comporte souvent une ou quelques populations composées de nombreux agents identiques, tandis qu'un système cognitif comporte quelques agents souvent hétérogènes. Nous avons vu que la frontière entre les deux est ténue. La distinction réside avant tout dans l'objectif lié à l'étude du système, comme indiqué dans [Parunak 99] :

- L'étude des SMA cognitifs cherche à améliorer les comportements individuels des agents en s'intéressant à leur intelligence individuelle, leur modèle cognitif, et aux communications. Elle met l'accent sur l'agent et ses capacités.
- L'étude des SMA réactifs cherche à comprendre le fonctionnement du système comme un tout, en se focalisant sur les interactions et la dynamique qui en résulte, donc sur les aspects collectifs du système.

Nous nous intéressons dans cette thèse aux aspects collectifs d'un SMA. Nous prenons donc en considération prioritairement des systèmes réactifs. Toutefois, un SMA cognitif peut lui aussi présenter des aspects collectifs intéressants, éventuellement indépendants de la richesse des comportements individuels. L'étude présentée dans cette thèse s'applique également à ces systèmes.

A.1.4 Dynamique d'un SMA

Un système multi-agents en fonctionnement évolue au cours du temps, sous l'influence des comportements individuels des agents et de la dynamique propre à l'environnement. D'un point de vue calculatoire, le modèle influences-réaction de [Ferber 96] permet de rendre compte de cette évolution, particulièrement dans le cas d'un SMA réactif. Il donne en effet une grande importance à l'environnement et à ses interactions avec les agents. Cette formalisation a l'avantage d'exprimer et de résoudre les conflits entre des actions contradictoires des agents.

Dans ce modèle, l'*état dynamique* d'un SMA est défini par un couple <σ, γ> dont les membres appartiennent respectivement à un ensemble d'états Σ de l'environnement et à un ensemble d'influences Γ des agents sur l'environnement. Une fonction opérateur $op : \Sigma \rightarrow \Gamma$ génère cette influence, tandis qu'une fonction $Laws : \Sigma \times \Gamma \rightarrow \Sigma$ traduit les lois qui régentent les réactions de l'état de l'environnement σ à une influence γ[1]. Des fonctions d'exécution et de réaction prennent en compte ces données pour mettre à jour le système.

La dynamique du SMA se traduit par une fonction $Cycle : \Sigma \times \Gamma \rightarrow \Sigma \times \Gamma$ qui met à jour son état dynamique <σ, γ> de façon régulière. À tout instant, le SMA se trouve dans un état, et son *évolution* est l'ensemble des états successifs.

Les notions d'état et de règles d'évolution nous permettront de comparer par la suite un SMA et un système dynamique, lorsque nous aborderons le contrôle d'un système.

A.1.5 Caractéristiques d'un SMA

Il existe des caractéristiques propres aux SMA, par rapport aux autres systèmes informatiques. Nous en fournissons une liste issue de la littérature, proposée par O. Boissier, S. Gitton et P. Glize dans [Boissier 04]. Nous développons ensuite deux d'entre elles dont nous aurons besoin par la suite : l'ouverture du système qui sous-entend l'existence de perturbations, et sa décentralisation liée à l'autonomie et la localité des agents. Les caractéristiques liées à l'émergence nous intéressent particulièrement, et font l'objet du chapitre suivant.

Pour [Boissier 04], un SMA possède la plupart des caractéristiques suivantes :

- Distribution : le système est modulaire, l'élément de base étant l'agent.
- Autonomie : un agent est en activité permanente et prend ses propres décisions en fonction de ses objectifs et de ses connaissances.
- Décentralisation : les agents sont indépendants, il n'y a pas de décisions centrales valables pour tout le système.
- Échange de connaissances : les agents sont capables de communiquer entre eux, selon des langages plus ou moins élaborés.
- Interaction : les agents ont une influence localement sur le comportement des autres agents, généralement sur un pied d'égalité (il n'y a pas d'ordres, seulement des requêtes).

1 Ce qui fait toute l'originalité de ce modèle : des actions simultanées peuvent ainsi être traitées.

- Organisation : les interactions créent des relations entre les agents, et le réseau de ces relations forme une organisation qui peut évoluer au cours du temps.
- Situation dans un environnement : les agents sont ancrés dans un environnement, source de données, de contraintes et d'incertitude, lieu d'actions et d'influences entre agents. L'évolution du SMA est la combinaison des évolutions des agents et de l'environnement.
- Ouverture : le système échange des informations avec l'extérieur, des agents peuvent entrer et sortir du SMA ou encore être modifiés en cours d'évolution.
- Émergence : « Dans tous les SMA, une fonction globale est attendue à partir d'un ensemble de spécifications au niveau local de chacune des entités. Cette propriété du niveau global n'est pas programmée dans les agents et n'existe que par leurs interactions conduisant à des processus permanents de réorganisation. »
- Adaptation : il est impossible de spécifier le but global et d'organiser les agents pour l'atteindre, ou même de prouver que le SMA réalise effectivement une fonction globale adéquate. Mais le système adapte son comportement à l'environnement en cours de fonctionnement, et offre une robustesse de ce comportement, à défaut d'une optimisation.
- Délégation : l'utilisateur accepte de ne pas maîtriser le comportement de l'application globale, à défaut de pouvoir supporter la complexité liée à l'ensemble des décisions prises par les agents dans le système. Il délègue une partie du contrôle de l'application globale aux agents.
- Personnalisation : lorsqu'un agent représente un utilisateur, typiquement dans un SMA appartenant à la famille des systèmes intégrés dans un contexte plus large, il s'adapte à lui.
- Intelligibilité : les SMA proposent une manière naturelle de modéliser d'autres systèmes ou de mettre en oeuvre des applications, ce qui les rend simples à appréhender pour un utilisateur extérieur.

Avant de développer la notion d'émergence en nous appuyant sur plusieurs de ces caractéristiques, nous revenons sur l'ouverture et la décentralisation d'un SMA.

A.1.5.1 Ouverture et perturbations

L'ouverture d'un système, qu'il soit physique ou informatique, représente la possibilité qu'il échange de l'information ou de la matière avec l'extérieur, et que son environnement possède une dynamique propre avec des évolutions imprévisibles. En informatique, cela signifie généralement que les composants du système sont conçus et évoluent séparément [Sichman 95].

Pour un système multi-agents, l'ouverture désigne la capacité d'ajouter ou de retirer dynamiquement dans le système des agents [Sichman 95], ou des fonctionnalités et des services [Vercouter 01, Vercouter 04] de ces agents. Pour [Wooldridge 01], c'est l'impossibilité de savoir lors de la conception quels seront les composants du SMA ni comment ils vont interagir les uns avec les autres. Nous venons de voir que [Boissier 04] indique également que l'utilisateur joue un rôle en termes de perturbations du système.

Nous résumons la notion d'ouverture d'un SMA comme l'existence potentielle de perturbations exogènes au système. Leur origine est facile à expliquer pour des SMA intégrés dans un environnement plus large, qui contient des utilisateurs humains ou une dynamique physique propre. Pour les SMA utilisés en résolution de problèmes, [Gechter 05] présente les contraintes du problème comme des perturbations de l'environnement. Pour la simulation de systèmes, enfin, les perturbations sont celles qui peuvent intervenir sur le système modélisé.

A.1.5.2 Décentralisation

La dynamique d'un SMA est définie au niveau des comportements individuels des agents, et son évolution découle de leurs interactions. Cela lui confère une robustesse justement recherchée lorsqu'il est utilisé pour résoudre des problèmes. Son fonctionnement ne dépend en effet d'aucun agent particulier. Une légère modification du comportement d'un agent (une panne par exemple) n'aura la plupart du temps aucune conséquence sur le fonctionnement du SMA dans son ensemble, à partir du moment où les agents sont interchangeables.

Cette décentralisation impose également au SMA de fortes contraintes. Elle peut être vue comme le fait qu'un intervenant extérieur au SMA est soumis à des contraintes de localité similaires à celles des agents : il ne peut percevoir et agir que sur une partie du système. L'état courant du SMA n'est pas nécessairement connu intégralement, seuls des indicateurs limités sur son état sont disponibles, et permettent son observation partielle. De la même manière, une action destinée à influer sur le SMA pourra ne pas s'appliquer à l'ensemble du système, ou pas de manière synchronisée.

C'est la cas particulièrement pour la famille des systèmes intégrés : par exemple, toute action ou observation globale sur l'Internet est déjà obsolète le temps d'être effectuée. Pour les SMA destinés à la résolution de problème ou à la simulation d'autres systèmes, en revanche, ce problème est moins contraignant car il est souvent possible d'observer et de modifier l'ensemble du SMA à tout instant.

A.2 Comportement global d'un SMA

Maintenant que les notions essentielles sur les SMA sont posées, nous définissons notre problématique. L'observation de l'évolution d'un SMA permet souvent d'identifier différents comportements qui impliquent la majorité des agents. Ces comportements jouent un rôle-clef dans notre étude, d'une part parce que ce sont eux que nous voulons maîtriser, et d'autre part parce qu'ils seront l'un des piliers de notre proposition. Les comportements du SMA sont mal connus, car ils ne sont pas définis explicitement au sein du système comme le sont les agents, mais émergent des interactions lors de l'évolution du système.

Nous venons de citer [Boissier 04] en mettant en avant l'existence de tels comportements non programmés dans les agents (caractéristique d'*émergence*), et la difficulté de les maîtriser (*adaptation*) en raison, entre autres, de leur complexité (*délégation*). Ce chapitre a pour but de développer la notion d'émergence et ses implications dans un SMA.

Le groupe de travail COLLINE[1] et avant lui le collectif IAD/SMA ([Jean 97]) ont proposé une définition synthétique de l'émergence dans un SMA :

1 Site web : http://www.irit.fr/COLLINE/Accueil.html

- Au niveau micro du système un ensemble d'agents sont en interaction, selon une dynamique rapide, au sein d'un environnement qui sert à la fois de médium de connaissance et d'ensemble de contraintes.
- Au niveau macro, un phénomène collectif est produit, suivant une dynamique plus lente donc plus observable que le niveau micro. L'émergence correspond à la description de ce phénomène global, soit par l'observateur, soit par les agents.

Le phénomène émergent n'est pas défini explicitement, mais résulte des comportements des agents et de leurs interactions. Il a lui-même une influence sur les agents. Le lien entre un comportement collectif et les individus qui le composent est difficile à établir. Obtenir un comportement collectif particulier en influant le niveau des agents est donc aussi difficile.

A.2.1 Caractérisation d'un phénomène émergent

Un récent travail de bibliographie sur la notion d'émergence, en particulier dans les SMA, est fourni par [Deguet 08]. Ce travail est axé sur ce qui caractérise un phénomène émergent et conclut sur l'importance primordiale des interactions par rapport aux comportements individuels pour leur apparition. Il définit l'émergence comme un « avis plus ou moins consensuel, sans fondement formel, regroupant tout ou partie des critères » :

- Sens commun : c'est l'apparition soudaine et importante d'un chose qui était cachée ou inexistante.
- Causalité descendante : l'émergent, ou épiphénomène, est perçu indépendamment des entités qui composent le système, même s'il survient d'elles, et a une causalité de son niveau global sur le niveau local de ces entités.
- Observation et prédiction : l'émergence facilite la description de l'état du système ou de son évolution future. Mais il n'y a pas de réduction possible : on ne peut pas déduire le niveau global du niveau local. L'observation du système est nécessaire, et le meilleur moyen de prévoir son évolution est de le simuler.
- Complexité à partir de la simplicité : l'émergence est une description simple d'un comportement complexe.
- Interprétation : l'émergence est l'interprétation d'un phénomène résultant d'une interaction entre les agents et l'environnement. L'observation et l'interprétation peuvent être faites par une personne extérieure au système (émergence faible) ou par les entités locales elles-mêmes (émergence forte).
- Émergence et auto-organisation : bien qu'elles se rencontrent souvent ensemble, ces notions ne font pas consensus. Pour [DeWolf 05a], elles sont indépendantes mais toutes deux positives, et c'est leur combinaison qui fait leur force. Pour [Shalizi 01], au contraire, l'auto-organisation génère la complexité du système, ce qui incite à rechercher des régularités émergentes pour la réduire.
- Le tout est supérieur à la somme des parties : cette phrase est classique de la caractérisation de l'émergence, et traduit le gain apporté au système par l'interaction de ses composants locaux.
- Point de vue pragmatique : un phénomène émergent est surprenant pour un utilisateur extérieur, même s'il connaît les règles de comportement locales.

Plusieurs de ces critères expriment l'idée que la connaissance du niveau local ne suffit pas pour savoir quel phénomène émergent va apparaître, ou pour expliquer ce phénomène. Réciproquement, obtenir un phénomène émergent en définissant ou en influant sur le niveau local d'un SMA est un problème difficile.

A.2.2 Enjeux de l'émergence au sein des SMA

Le chapitre [Drogoul 04] donne une définition de l'émergence axée sur ce qui est nécessaire dans le système pour obtenir un phénomène émergent, et sur les questions de recherche actuelle qui y sont liées. Nous le résumons ici.

Dans un système d'entités en interaction, l'émergence est la production d'un phénomène (processus, état stable ou invariant) global au regard des entités, et observé soit par un observateur extérieur, soit par les entités elles-mêmes.

Si l'on considère un système multi-agents réactif, l'*observateur* est généralement extérieur au système car les agents eux-mêmes n'ont pas la capacité d'identifier des comportements globaux. Dans un SMA cognitif, les agents identifient les phénomènes globaux et modifient explicitement leur dynamique pour en tenir compte.

La construction d'un SMA vise à la *réalisation d'objectifs globaux* en définissant son niveau local. Son évolution dépend ensuite de données fournies par l'environnement, d'une initialisation particulière, ou de la modification d'agents due à l'ouverture du système. Il y a émergence si le SMA atteint un état non définissable a priori pour chaque évolution.

Un phénomène émergent est *imprévisible* pour diverses raisons. Cela peut provenir d'une évolution stochastique du SMA, c'est-à-dire de l'intervention du hasard dans les comportements individuels qui gouvernent la dynamique du système. Ce caractère stochastique peut être inhérent aux agents, ou palliatif de l'ignorance d'un système modélisé par un SMA dédié à la simulation. Le non-déterminisme du système peut aussi provenir de son caractère ouvert ou de l'influence de l'utilisateur. Enfin, même un système clos et déterministe peut être analytiquement imprévisible, à cause de la complexité liée à l'émergence.

Un *utilisateur* du SMA peut avoir le double rôle d'observer le système pour identifier et classer les formes émergentes, et d'agir sur le SMA pour orienter l'émergence. En effet, le phénomène global identifié par un observateur possède un sens pour lui, et peut correspondre à des états intéressants, à des solutions recherchées. En tant qu'utilisateur, il cherche à assurer un résultat émergent. Pour y parvenir, une analyse de type Monte-Carlo est nécessaire : explorer les états en faisant varier la source d'imprévisibilité, et classifier les résultats en observant leurs régularités.

L'une des questions principales liées à l'émergence et aux SMA est de déterminer comment *concevoir le niveau local* du système (agents, interactions et organisation) pour garantir l'obtention d'un comportement émergent particulier au niveau global. Généralement, la démarche consiste en une élaboration préalable d'un SMA grâce à des considérations heuristiques, suivie d'une phase empirique composée d'une succession de tests et d'adaptations permettant de parvenir à une configuration voulue. Il n'y a pas de vérification formelle possible des propriétés du SMA. Des protocoles expérimentaux permettent toutefois de couvrir largement l'espace des réponses, mais se heurtent à la difficulté de spécifier

formellement les comportements émergents.

La phase d'élaboration préalable du SMA repose sur des principes d'*ingénierie* qui cherchent à reproduire des phénomènes naturels connus, en particulier le biomimétisme et le sociomimétisme.

A.2.3 Difficultés pour diriger le comportement d'un SMA

Certains auteurs cherchent à diriger les phénomènes émergents d'un SMA, et parlent alors de comportements émergents. Parmi eux, [DeWolf 05c] explique qu'il est difficile de prouver que le comportement est bien maîtrisé, par manque de démonstrations formelles.

Dans cet article, l'auteur explique qu'il est difficile de garantir un certain comportement global émergent dans un SMA à cause de sa nature non déterministe. Il présente un système composé de véhicules automatiques destinés à transporter un chargement entre des points d'entrée et des points de sortie dans un réseau. Les propriétés désirées sont soit une distribution homogène des agents, soit un flux de transport élevé. Il explique qu'avant de concevoir un système qui présente ces propriétés, il faut être en mesure de garantir qu'elles sont atteintes et maintenues pour un système existant. Pour cela, il propose une approche « equation-free » qui permet d'analyser le comportement global pour assurer que les spécifications sont atteintes. Il insiste aussi sur la difficulté de simplement identifier le comportement global présenté par un SMA.

Cette même difficulté est citée dans [Contet 08], qui propose d'utiliser des outils de physique statistique, basés sur la notion d'énergie, pour vérifier la stabilité du comportement global d'un système de flocking.

Nous résumons ainsi les trois problèmes qui se heurtent à la difficulté de lier le niveau local où est défini le SMA à ses comportements globaux observés :

- Mesurer le comportement global, c'est-à-dire déterminer celui qui est présenté par un SMA à un instant donné. Ce problème dépend essentiellement du SMA étudié, et nous y reviendrons de manière plus approfondie lorsque nous parlerons d'application de notre proposition.
- Maîtriser un SMA, c'est-à-dire faire en sorte qu'il présente effectivement un comportement global désiré, diriger le système.
- Certifier qu'une solution de maîtrise d'un SMA donne bien les résultats attendus, ou encore évaluer cette solution pour savoir si elle est utile.

A.3 Bilan : de l'émergence aux comportements du SMA

En résumé, un phénomène émergent dans un SMA réactif est un phénomène global, difficilement prévisible à partir de la connaissance du niveau local du système, interprété par un observateur extérieur. Il est global en espace, car il fait intervenir collectivement plusieurs agents, et en temps, puisque sa dynamique plus lente le fait apparaître stable.

Nous définissons le comportement d'un SMA à un instant donné comme une description selon le point de vue de l'observateur de l'ensemble des phénomènes émergents qu'il observe. Ce comportement a un sens pour l'observateur, et peut être désiré ou non. Plusieurs

comportements différents peuvent être atteints par un même SMA, en fonction de conditions initiales, de perturbations, ou du caractère aléatoire de l'évolution du système.

La question que nous posons et précisons au fil de cette première partie est de savoir comment l'utilisateur d'un SMA, par exemple son concepteur, peut influer dessus afin d'assurer qu'il présente un comportement désiré. De la même manière que [Drogoul 04], nous considérons donc l'observateur comme un utilisateur : il cherche à agir sur le SMA pour atteindre, ou au moins favoriser, un comportement souhaité. Le comportement global est facilement observé, mais plus difficile à expliquer et à maîtriser, à cause de sa nature émergente. En outre, il est difficile de prouver que ce comportement est bien maîtrisé, par manque de démonstrations formelles, comme indiqué dans [DeWolf 05c].

Ces problèmes se posent que l'on cherche à diriger le comportement d'un SMA pré-existant, ou même que l'on construise un SMA pour qu'il présente un comportement désiré. L'utilisation d'approches par mimétisme, qui copient des systèmes connus en espérant que les phénomènes qui s'y trouvent soient transférés dans le SMA construit, montre à quel point il est difficile de maîtriser l'émergence en définissant les comportements individuels.

La problématique de maîtriser le comportement global d'un SMA fait l'objet de l'état de l'art qui suit.

B. État de l'art

Dans ce chapitre sont présentées différentes approches qui proposent de maîtriser le comportement global d'un SMA. L'objectif est d'assurer le comportement pour lequel le SMA est créé. Nous distinguons deux grandes catégories : les approches par construction, et celles qui cherchent à contrôler le système. Pour chacune, nous mettons en relief les apports techniques, les difficultés rencontrées par les auteurs et leurs recommandations pour y faire face.

B.1 Approches par construction

Nous regroupons sous cette appellation les travaux qui visent à concevoir et construire des SMA qui présentent un comportement désiré. [Edmonds 04a] explique qu'il existe deux manières d'obtenir un SMA *utile*, c'est-à-dire qui se comporte conformément aux spécifications : l'ingénierie et l'adaptation. La première se situe entièrement en amont de la construction du système, tandis que la seconde considère un système existant et l'améliore expérimentalement par essais et révisions jusqu'à ce qu'il soit satisfaisant. Il est montré dans [Edmonds 04b] que l'ingénierie seule n'est généralement pas suffisante pour parvenir aux spécifications attendues et qu'une approche expérimentale est nécessaire.

La partie ingénierie a déjà été évoquée avec la notion de mimétisme de systèmes connus. Nous nous concentrons sur la phase empirique d'adaptations successives dont on attend qu'elles permettent d'atteindre un comportement souhaité. Deux types d'approches qui s'occupent de cette phase se trouvent dans la littérature :

- Des méthodes de conception, qui gèrent à la fois la phase d'ingénierie et la phase de révision. Dans la seconde, elles aident à remettre en question des parts importantes du SMA, le plus souvent les fondements du comportement des agents, en suivant des modèles sociaux ([Boissier 04]).
- Des approches par ajustement des paramètres du SMA, qui recherchent les valeurs de paramètres optimales d'un système déjà conçu mais pas encore calibré pour assurer un comportement souhaité.

Ces types d'approches peuvent être utilisés successivement.

B.1.1 Méthodes de conception

Une analyse de ces méthodes est donnée dans [Arlabosse 03], et [Campagne 05] fournit une liste de 40 d'entre elles en en détaillant 9. Leur but est de définir entièrement le comportement des agents, sans cadre formel qui délimite leurs capacités : le concepteur du système est libre de créer ses agents comme il l'entend. On comprend qu'elles s'appliquent plus naturellement à des SMA cognitifs que réactifs.

Une présentation générale de ces méthodes serait trop longue. À la place, nous présentons un problème typique qui y fait appel, avant de donner deux exemples notoires de méthodes de conception. Nous discutons pour terminer de leurs avantages et de leurs inconvénients.

B.1.1.1 Utilité de ces méthodes

L'article [Gage 92] pose le problème du maintien d'une formation d'un groupe de robots mobiles en définissant entièrement les comportements individuels pour y parvenir[1]. Par exemple, l'objectif peut être de garantir un espacement constant des robots qui se déplacent sur un terrain irrégulier (voir figure 1), en définissant leur comportement de toutes pièces.

Cet exemple est typique des méthodes de conception, puisqu'il y a une très grande liberté pour définir le SMA au niveau local, même si certaines contraintes dues ici à la nature des robots subsistent.

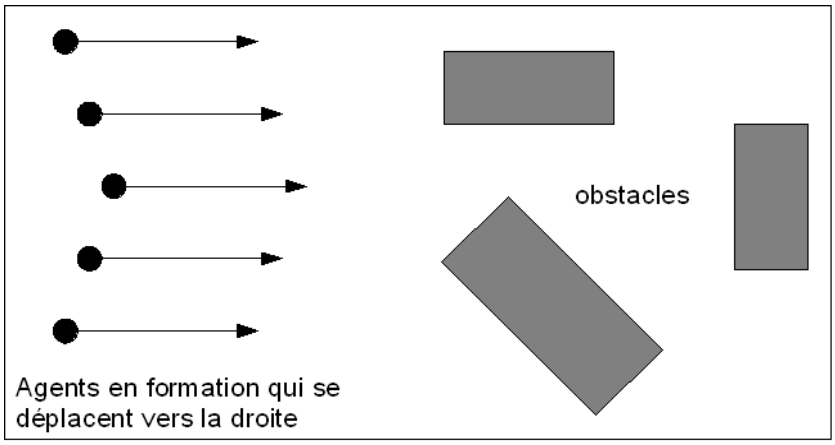

Figure 1: Problèmes du maintien d'une formation de robots mobiles

Dans cet article, l'auteur insiste sur deux recommandations importantes : d'une part observer les comportements globaux possibles pour comprendre les capacités du SMA en terme de comportements globaux, et d'autre part trouver des mesures de réussite pour vérifier les performances du contrôle. Les deux mêmes recommandations se retrouvent sous la forme d'étapes dans [DeWolf 05b]. La difficulté de la vérification du comportement global est due au manque de caractérisation formelle de ce comportement, dont nous avons déjà parlé.

B.1.1.2 Exemples illustratifs de méthodes de conception

La méthode ADELFE ([Bernon 06], [Bernon 05]), qui se place dans la théorie AMAS, est fondée sur le modèle social de la coopération des agents. Elle consiste à identifier les situations non coopératives et à modifier les comportements de certains agents pour revenir à une situation plus favorable. Par exemple, [Bernon 06] présente un écosystème composé de poissons, de crevettes et d'algues. Les crevettes consomment les algues et se font manger par les poissons. L'objectif, en considérant la population de crevettes, peut être de survivre aussi longtemps que possible, ou de consommer un maximum d'algues en un temps limité. Le comportement des crevettes est défini dans son ensemble, de manière simple. Appliquer l'approche revient à améliorer ce comportement, en lui ajoutant des règles qui augmentent la collaboration. Par exemple, ne manger une algue que si aucune autre crevette n'en est plus

1 Il appelle ce problème le *contrôle* du SMA, mais nous réservons ce terme pour une autre notion que nous verrons plus loin.

proche, ou encore ne pas la manger en cas de conflit.

La méthode Voyelles de [Demazeau 01], évoquée en tout début de chapitre, ne se contente pas de définir un SMA selon les quatre dimensions Agent, Environnement, Interaction et Organisation, mais propose de guider le choix de modélisation pour chaque dimension lors de la phase de conception. La thèse [Ricordel 01] qui s'appuie sur cette méthode propose l'exemple très parlant de son application à la Robocup[1]. L'environnement est fourni par les règles officielles, et peut donc être codé initialement et définitivement. Ces mêmes règles interdisent toute communication directe entre robots, donc la dimension Interactions est inutile. Finalement, seuls les agents et l'organisation du SMA vont faire l'objet de révisions pour faire en sorte que l'équipe joue le mieux possible.

B.1.1.3 Avantages et limites de ces méthodes

Les méthodes de conception laissent une grande place aux heuristiques : pour un objectif global donné, un intervenant humain essaie de déterminer et de corriger les comportements des agents grâce à son expérience personnelle. Le lien entre le niveau local de définition du SMA et son comportement global est ainsi appréhendé par un cerveau humain, ce qui supprime toute question de complexité algorithmique.

Mais ce qui fait la force de ces méthodes est aussi leur faiblesse : un humain n'a pas la capacité d'effectuer de nombreux tests et calculs comme une machine, et ne peut pas assurer de penser à toutes les améliorations possibles. Par exemple, dans la méthode ADELFE, on ne peut pas garantir d'identifier toutes les situations non coopératives par rapport à un objectif donné. L'implication de l'humain dans ces approches les rend très riches, mais fait qu'elles dépendent des capacités de cet humain à faire les bons choix.

Une autre limite est liée au principe même de conception du SMA. Il présuppose qu'il existe un moyen de toujours assurer un comportement désiré. Ces méthodes sont donc valables dans un cadre où les source d'imprévisibilité de l'évolution du SMA sont limitées. En particulier, elles ne considèrent pas le cas où le système subit des perturbations extérieures. Nous reviendrons sur ce problème un peu plus loin.

B.1.2 Approches par ajustement des paramètres

Un champ entier de la recherche sur les SMA est lié à la *calibration* d'un système existant ([Culioli 94]), c'est-à-dire l'obtention de valeurs de paramètres du système qui permettent d'optimiser l'une de ses caractéristiques. Cette optimisation se fait expérimentalement, en explorant l'espace des paramètre du SMA à calibrer, et en retenant la meilleure combinaison de valeurs. Typiquement, il s'agit de trouver des valeurs qui permettent d'obtenir un comportement global quelles que soient les conditions initiales.

Nous présentons un échantillon des approches qui répondent à ce problème, en insistant sur les difficultés rencontrées et sur les recommandations des auteurs pour y répondre. Nous développons aussi les problèmes traités dans chaque approche, pour montrer le type de comportements globaux souhaités et les influences envisagées au niveau local.

1 Compétition de football entre agents : http://robocup.org/

B.1.2.1 Utiliser la connaissance du système

L'article [Fehler 04] propose des techniques qui exploitent la connaissance qu'on a d'un système pour calibrer les paramètres d'un SMA qui le simule. Trois problèmes propres à la calibration d'un SMA sont donnés pour justifier le recours à ces techniques :

- Le nombre de paramètres à calibrer est particulièrement élevé dans un SMA.
- Le comportement global dépend des paramètres de façon complexe, à cause des interactions entre agents.
- L'évolution d'une simulation multi-agents présente un important coût de calcul.

Dans cette proposition, l'espace des paramètres à calibrer est décomposé de manière à résoudre des sous-objectifs, choisis en fonction de la connaissance du système. Le problème de calibration est successivement résolu pour des sous-espaces de plus faible dimension ce qui limite l'explosion combinatoire de l'espace des paramètres.

Cette approche est appliquée à un SMA qui modélise une colonie d'abeilles, à la fois à l'intérieur de la ruche, pour l'apport de nourriture aux larves, et à l'extérieur, pour la recherche de pollen. Deux objectifs sont donnés en exemple : optimiser la récolte du pollen, et obtenir des structures sociales spécifiques, observables au niveau global. Pour y parvenir, on peut agir sur plusieurs paramètres : un seuil pour nourrir les larves, des paramètres pour chercher les larves qui ont besoin de nourriture, et d'autres qui guident la recherche du pollen, comme des seuils pour chercher une ressource, l'indiquer ou en changer.

B.1.2.2 Explorer les paramètres en parallèle

Une solution de calibration proposée dans [Calvez 07] se base sur un principe similaire à la proposition précédente, à savoir scinder l'espace des paramètres. Il s'agit à la fois d'explorer en parallèle cet espace par les différents agents du SMA[1], et de déterminer les zones à explorer en traitant indépendamment chaque paramètre. Cela permet d'optimiser un paramètre en fonction uniquement des meilleures valeurs trouvées pour les autres. Cette approche est justifiée par le nombre élevé de paramètres à calibrer dans le SMA.

L'algorithme proposé est appliqué à un SMA qui représente le fourragement de fourmis, c'est-à-dire la recherche de nourriture couplée à une émission et un suivi de phéromones. L'objectif global est de maximiser la quantité de nourriture rapportée au bout de 500 pas de simulation. Il s'agit donc plus de diriger et d'optimiser une propriété du système que de l'amener à présenter un comportement global. Pour cela, on peut jouer sur les taux de diffusion et d'évaporation des phéromones, sur la vitesse des agents, leur perception (2 paramètres) et la quantité de phéromones déposée.

B.1.2.3 Suivre un plan d'expériences dynamique

La plupart des travaux de calibration d'un SMA cherchent à tirer partie d'une idée algorithmique résumée dans [Amblard 03] sous le nom de plan d'expériences dynamique. Il s'agit de choisir les simulations à effectuer pour connaître le comportement du système sous certaines valeurs de paramètres, en fonction des résultats déjà obtenus.

---

1 Cela n'est donc possible que pour l'optimisation de paramètres propres aux agents.

Pour l'auteur, la compréhension d'un modèle individus-centré[1] échappe à son modélisateur car les interactions entre agents produisent un comportement global qui n'est pas spécifié explicitement dans le modèle.

L'objectif de [Amblard 03] est de comprendre le fonctionnement d'un phénomène individus-centré, en proposant une collection de SMA qui le modélisent, du plus simple au plus réaliste. L'observation de chacun de ces modèles se fait à différents niveaux, et en particulier, il est intéressant de caractériser et d'identifier les phénomènes collectifs, qui correspondent à nos comportements globaux, grâce à un ensemble d'indicateurs minimal. Ceux-ci doivent *« rendre compte de la situation observée au cours d'une expérience »* et également *« être pertinents et le plus possible porteurs de sens pour chacune des expériences »*. Le principe de créer une collection de SMA pour comprendre un comportement global est appliqué pour identifier l'évolution de dynamiques d'opinions.

Dans ce problème, les opinions sont réparties dans un espace muni d'un ordre total, et les comportements globaux sont représentés par le nombre de points de convergence dans cet espace et la valeur de ces points. Ces comportements sont étudiés en fonction d'un paramètre d'incertitude, qui représente le rayon d'influence d'un individu sur un autre, et un paramètres d'influence, qui définit la manière dont les opinions sont modifiées. Ces deux paramètres sont globaux, c'est-à-dire communs à tous les agents.

L'exploration de l'espace des paramètres se fait en suivant un plan d'expériences, c'est-à-dire une planification des simulations à réaliser pour les tester. Mais l'auteur évoque le principe de plans d'expériences dynamiques, où la planification n'est plus préalable aux simulations, mais est orientée en fonction de leurs résultats. De cette manière, seules les expériences les plus prometteuses sont menées. Des zones de l'espace à explorer sont évitées, ce qui diminue la complexité algorithmique, au détriment de l'assurance d'atteindre un optimum global.

a) Algorithmes génétiques et autres métaheuristiques

On retrouve l'idée de plan d'expériences dynamique implicitement dans l'utilisation de métaheuristiques[2] ([Dreo 06]), en particulier des algorithmes génétiques, pour explorer l'espace des paramètres de façon rapide. L'exploration est incomplète mais privilégie les zones susceptibles de fournir les meilleurs résultats. Ces méthodes n'assurent donc pas d'obtenir un optimum global, mais proposent une façon d'obtenir un optimum local que l'on peut supposer proche de l'optimum global. Le résultat mesuré, celui qui fait l'objet d'une optimisation, est appelé ici *fitness*. Il s'agit en général d'une simple propriété globale directement mesurable (voir figure 2).

Les métaheuristiques sont appliquées à la calibration de SMA dans de nombreux travaux comme [Calvez 05], [Narzisi 06], [Sauter 01] ou [Sierra 02]. Leur utilisation sous-entend que l'espace à explorer est trop vaste et chaque simulation trop longue pour effectuer une simulation en chaque point de cet espace. Le recours à ces méthodes indique aussi qu'il n'y a pas de connaissance a priori sur le lien entre les paramètres et la valeur de fitness, qui permettrait par exemple d'interpoler ou d'extrapoler le comportement à partir des simulations déjà effectuées.

1 Équivalent pour nous à un SMA de la famille d'application dédiée à la simulation d'autres systèmes.

2 Sous ce terme sont regroupées les techniques d'optimisation approchée, telles que la descente de gradient ou le recuit simulé, qui explorent partiellement mais efficacement un espace de paramètres.

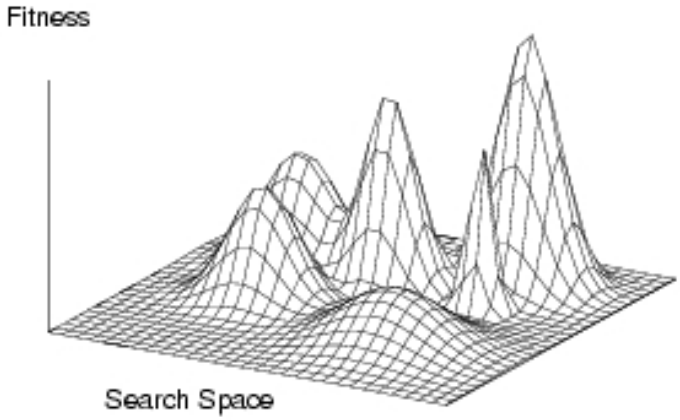

Figure 2: Recherche d'optimum dans un espace à deux dimensions. La fitness est une valeur directement mesurable. Une méthode approchée risque de donner comme résultat un optimum local.

b) Estimation de la valeur de fitness

Les méthodes précédentes présupposent qu'il est possible de mesurer la fitness lors d'une expérience. Or dans certains cas, de nombreuses répétitions de l'expérience, appelées *réplications*, sont nécessaire pour estimer l'espérance de cette mesure. La notion de plan d'expériences dynamique ne consiste donc plus seulement à privilégier l'exploitation de zones prometteuses de l'espace des paramètres, mais également à l'affinement de l'estimation de la fitness en des points prometteurs en multipliant les réplications en ces points. [Brueckner 03] propose une solution décentralisée à ce problème, en décidant d'effectuer ou non une expérience en un point de l'espace des paramètres en fonction de la fitness estimée en ce point et de la meilleure fitness trouvée par ailleurs dans l'ensemble de cet espace.

Cette approche est appliquée à un SMA chargé de résoudre un problème de coloration de graphe distribuée. Le principe est d'associer un agent à chaque sommet du graphe. Ces agents mettent à jour la couleur associée à leur sommet en fonction de leurs voisins. L'objectif est de trouver un équilibre entre une coloration jugée bonne et un temps de résolution restreint. On modifie pour cela les paramètres qui définissent les comportements des agents, tous identiques, comme la probabilité de changer de couleur ou le mécanisme de sélection de cette couleur.

c) Combiner une exploration rationnelle et une évaluation de la fitness

L'approche proposée par nous-mêmes dans [Klein 06], comme celle, déjà présentée, de [Calvez 07], tente de combiner une exploration rationnelle de l'espace des paramètres et une limitation du nombre de réplications pour estimer la fitness. Un plan d'expériences dynamique est utilisé pour explorer l'espace des paramètres en discrétisant cet espace par des mailles plus ou moins fines en fonction des variations du comportement global. Ces

variations sont estimées par des expériences aux sommets de ces mailles, dont le nombre varie en fonction des résultats trouvés.

Le système étudié est une dynamique de population, dont le comportement global est mesuré par l'évolution de la population totale, selon qu'elle est stable, périodique, chaotique, ou qu'elle s'éteint. L'un de ces comportements est choisi comme objectif, et l'influence de quatre paramètres est étudiée : le taux de régénération des ressources, la quantité maximale de ressources consommée par un agent en une fois, la consommation d'énergie des agents, et leur seuil d'énergie pour se reproduire.

B.1.2.4 Synthèse de la calibration

Le principe de la calibration de système est de tester successivement des combinaisons de valeurs de paramètres pour trouver celle qui convient le mieux. Trois limites s'opposent à une exploration complète et naïve de l'espace des paramètres, liées à la complexité algorithmique de cette exploration :

- La première vient de la taille de l'espace à explorer : l'influence d'un paramètres peut varier en fonction des autres paramètres, donc chaque combinaison de paramètres doit être testée. Cela est bien sûr impossible dès qu'un paramètre a des valeurs continues ou n'est pas borné. Même dans le cas où l'espace est fini, il devient rapidement trop grand pour être exploré quand le nombre de paramètres augmente, à cause de l'explosion combinatoire qui en résulte.
- La seconde limite concerne l'évaluation d'une combinaison de valeurs de paramètres : le résultat d'une simulation avec des valeurs de paramètres données peut dépendre de conditions initiales aléatoires de la simulation. Il est alors nécessaire d'effectuer plusieurs fois la même expérience avec des conditions initiales différentes pour évaluer correctement les valeurs de paramètres.
- La troisième limite est le temps nécessaire pour effectuer chaque expérience en un point de l'espace des paramètres, c'est-à-dire chaque simulation. La durée d'une simulation multi-agents est particulièrement élevée par rapport à d'autres systèmes, d'après [Fehler 04].

Le point commun à toutes les méthodes de calibration présentées est de limiter cette complexité de l'exploration. Nous retenons les principes qui permettent d'y parvenir : utiliser la connaissance du système, scinder l'espace à explorer, et utiliser des méthodes d'optimisation expérimentales approchées.

Un autre problème est régulièrement soulevé : estimer l'intérêt d'un point de l'espace des paramètres. [Brueckner 03] met en relief la nécessité d'évaluer ce point de manière approchée en effectuant plusieurs expériences, et [Amblard 03] explique la difficulté d'identifier un comportement global.

La calibration présente l'avantage de permettre l'optimisation de n'importe quel paramètre du système, qu'il entre dans la définition du comportement des agents ou de l'environnement du SMA par exemple. Un autre avantage est sa relative simplicité : ses méthodes sont faciles à formaliser et peuvent s'appliquer à des SMA divers.

Son principal défaut est lié à cette simplicité : la calibration est une optimisation des paramètres en moyenne, pour toutes les exécutions possibles, sans considération de cas

particuliers. Les valeurs de paramètres optimales sont sensées assurer un comportement souhaité quelles que soient les conditions d'une simulation. Or il existe des SMA pour lesquels les meilleurs paramètres dépendent de l'instance de simulation.

C'est le cas par exemple du système présenté dans [Bourjot 01]. Un SMA inspiré du comportement des araignées sociales est dédié à la détection de régions dans des images. Les valeurs optimales de deux de ses paramètres dépendent de l'image, donc de l'environnement du SMA ou encore de l'instance de problème à résoudre, selon le contraste de la région à détecter. Il n'existe donc pas de valeurs de paramètres optimales dans l'absolu.

Le principe même de la calibration est ainsi remis en cause : au lieu de fixer des valeurs de paramètres optimales en concevant le SMA, c'est lors de son exécution, en fonction du cas de figure rencontré – par exemple en fonction des conditions initiales – que les meilleurs paramètres doivent être appliqués.

B.1.3 Discussion : limite des approches par construction

Les approches par construction ont pour objectif de créer un SMA tout en assurant qu'il présente un comportement global désiré. Or un SMA présente de nombreuses sources d'imprévisibilité, comme nous l'avons vu :

- son évolution peut être stochastique,
- son évolution est complexe, analytiquement imprévisible, à partir de conditions initiales non maîtrisées ou d'une instance de problème à résoudre,
- le système peut être soumis à des perturbations, venant d'un extérieur non maîtrisé ou de l'utilisateur lui-même.

Il est donc difficile, sinon impossible, de garantir analytiquement qu'un SMA présente un comportement désiré, quelle que soit la méthode de construction utilisée (comme le signale [DeWolf 05c]), dès que l'une de ces sources d'imprévisibilité entre en jeu.

De ce fait, même si la construction d'un SMA permet d'atteindre les spécifications dans la plupart des cas, elle n'indique pas quoi faire lorsqu'une évolution particulière l'amène dans un comportement non désiré. Comme ce comportement est par définition stable, la méthode de construction aura échoué à remplir son rôle de maîtrise du comportement.

Dans le chapitre suivant, nous présentons des approches qui prennent en compte de façon dynamique l'évolution d'un SMA en cours d'exécution, pour l'amener d'un comportement non désiré à un comportement voulu.

B.2 Approches par contrôle

Le contrôle d'un système consiste à agir au cours de son évolution afin de diriger son comportement. C'est une notion formelle issue de l'étude des systèmes dynamiques. Ces systèmes sont traditionnellement étudiés, en théorie du contrôle et en automatique, comme des objets préexistant sur lesquels on exerce une influence.

Dans ce chapitre, nous commençons par définir ce qu'est un système dynamique et à mettre en parallèle cette notion avec celle de système multi-agents.

Nous présentons ensuite ce qu'est le contrôle dans le cadre des systèmes dynamiques, et nous expliquons pourquoi les approches analytiques classiques ne s'appliquent pas à un SMA. Il existe des approches empiriques qui se réclament du contrôle pour diriger des SMA, appliquées soit au niveau local, soit au niveau global du système. Nous les présentons dans la suite ce chapitre. La seconde catégorie est très peu développée : à notre connaissance, il n'y a qu'un seul travail de recherche qui traite ce problème à ce jour.

B.2.1 Systèmes dynamiques et SMA

D'après [Drogoul 04], l'étude de l'évolution d'un SMA réactif peut être exprimée dans le paradigme des systèmes dynamiques. Nous déclinons un SMA comme l'un de ces systèmes, ce qui permet d'éclairer sous un nouvel angle ses particularités.

B.2.1.1 Définition d'un système dynamique

La définition qui suit est paradigmatique, on la retrouvera à peu de variations près dans tout traité sur les systèmes dynamiques, comme [Manneville 00].

Un système dynamique est un système qui évolue au cours du temps de façon déterministe. Cette évolution peut être continue ou discrète. Dans tous les cas, elle est donnée par en ensemble de règles $\mathcal{F}$, le plus souvent un système d'équations, différentielles dans le cas continu. L'*état* du système à tout instant est donné par les valeurs de ses variables d'état X. Les règles d'évolution s'appliquent à ces variables d'état :

$X_{t+1} = \mathcal{F}(X_t)$ dans le cas discret, et $\frac{dX}{dt} = \mathcal{F}(X)$ dans le cas continu.

L'état du système peut être représenté dans son *espace des phases* $\mathcal{X}$, dont chaque dimension correspond à l'une des variables d'état. L'évolution du système est alors équivalente à une trajectoire dans cet espace.

Le système est appelé *linéaire* si la fonction $\mathcal{F}$ est linéaire, et il est appelé *non linéaire* dans le cas contraire. Un système non linéaire peut présenter un comportement dit chaotique, caractérisé par une forte sensibilité aux conditions initiales : deux états proches à un instant donné peuvent conduire à deux évolutions du système radicalement différentes. Ce comportement chaotique amène une imprévisibilité de l'évolution du système en dépit de son déterminisme.

Nous pouvons présenter, de manière simplifiée, la notion d'*attracteur* comme une portion $P \subset \mathcal{X}$ de l'espace des phases telle que toute évolution du système à partir d'un état $X \in P$ reste dans P : $\forall X \in P, \mathcal{F}(X) \in P$.

Un même système peut comporter plusieurs attracteurs simultanément[1].

Enfin, il peut exister un ensemble *A* de *paramètres de contrôle*, tels que

$X_{t+1} = \mathcal{F}(X_t, A_t)$ ou $\frac{dX}{dt} = \mathcal{F}(X, A)$

1 Un traité sur les systèmes dynamiques en apportera une définition plus rigoureuse.

On dit alors que le système est paramétré. Lorsque *A* varie, le nombre et la nature des attracteurs peuvent être modifiés. On parle alors de *bifurcation* : le comportement du système varie qualitativement lorsque ces paramètres franchissent certaines valeurs.

B.2.1.2 Parallèle avec les SMA

Nous avons vu dans [Ferber 96] qu'un SMA pouvait être défini par un état dynamique et une fonction d'évolution. L'état dynamique décrit précisément le SMA à un instant donné et à son niveau local, et correspond à l'état $X \in \mathcal{X}$ d'un système dynamique. La fonction d'évolution est la composition des comportements individuels des agents, de la dynamique propre de l'environnement, et des règles de mise à jour du SMA. Elle est assimilable aux règles d'évolution $\mathcal{F}$ d'un système dynamique. Les concepts de SMA et de système dynamique sont donc assez proches.

Ils divergent néanmoins sur deux points. Tout d'abord, la fonction d'évolution d'un SMA peut être stochastique, tandis que l'évolution d'un système dynamique est toujours déterministe. Ensuite, un SMA est susceptible de subir des perturbations à cause de son caractère ouvert, qui peuvent avoir une influence sur son état dynamique – par exemple sur les objets de l'environnement – comme sur sa fonction d'évolution – par exemple sur les comportements individuels. Comme indiqué dans [Boissier 04], l'utilisateur lui-même peut être vu comme une source de perturbation, ce qui différencie son influence de celle qu'il exerce sur un système dynamique au travers des paramètres de contrôle.

Non-linéarité d'un SMA

D'après [Goldstein 99], le caractère émergent d'un phénomène global est indissociable de la nature non-linéaire du système dans lequel il apparaît. Il doit notamment exister des boucles de rétroaction positives (ou interactions rétroactives dans [Hassas 03]) pour que surviennent ces phénomènes. Cela signifie que le phénomène émergent influe sur les entités locales pour se renforcer, ce qui assure leur stabilité. Ce type d'influence est nécessairement non-linéaire. Il en résulte qu'un SMA dont on étudie le comportement global a une évolution potentiellement chaotique et fortement sensible aux conditions initiales et aux perturbations.

L'idée qu'un SMA est un système non-linéaire se retrouve dans le parallèle qui existe entre son comportement global et un attracteur. Il s'agit dans les deux cas d'une régularité stable et observable. Il existe souvent plusieurs attracteurs dans un système dynamique, de même qu'il existe plusieurs comportements accessibles à un SMA en fonction des conditions initiales et de ses autres sources d'imprévisibilité. Pour modifier le comportement global courant d'un SMA, il faut agir sur le système de manière équivalente au franchissement d'une bifurcation dans un système dynamique.

B.2.2 Contrôle d'un système dynamique

Le problème du contrôle d'un système dynamique est traité par deux disciplines scientifiques complémentaires : la théorie du contrôle et l'automatique. Nous les présentons rapidement ici. L'idée est que si les systèmes dynamiques et les SMA sont similaires, les outils qui permettent de diriger les premiers pourraient peut-être être appliqués aux seconds avec succès. Nous allons voir pourquoi ce n'est pas tout à fait le cas, mais nous nous inspirerons de ces outils lorsque nous définirons le contrôle d'un SMA.

La théorie du contrôle est l'étude formalisée du comportement des systèmes dynamiques paramétrés en fonction de l'évolution des valeurs de leurs paramètres (voir [Manneville 00]). En particulier, la question du contrôle consiste à se demander quelle suite de valeurs des paramètres de contrôle A (cas discret) ou quelle fonction A (cas continu) doit être utilisée pour faire converger l'état du système vers un état particulier appelé *cible*. L'exemple typique est la position d'équilibre instable d'un pendule inversé. Des outils mathématiques sont utilisés pour étudier la façon dont se fait cette convergence.

D'un abord plus facile, l'automatique pose une question similaire, d'un point de vue systémique et d'ingénierie [Prouvost 04]. Son objectif est de réguler le comportement d'un système dynamique paramétré, pour que son état se rapproche autant que possible d'un état voulu appelé *consigne* et résiste à d'éventuelles perturbations. Des exemples typiques de consignes sont le maintien de la vitesse d'un véhicule ou de la température d'un four. La consigne peut évoluer dans le temps. Les paramètres de contrôle sont appelés les commandes du système, et c'est sur elles que peut agir le système de contrôle, en fonction de l'observation de l'état courant du système.

Nous détaillons les différentes étapes et questions qui se posent lors du contrôle d'un système dynamique.

B.2.2.1 Données du problème

Un problème de contrôle est défini par trois données :

- La cible, ou consigne, à atteindre et maintenir,
- Des moyens d'action sur le système : ses paramètres ou commandes,
- Des moyens d'observation du système.

La cible peut être un point de l'espace des phases $\mathcal{X}$. Il s'agit en général d'un point fixe, c'est-à-dire un point de l'espace des phases $\mathcal{X}$ qui soit *stable* pour la fonction $\mathcal{F}$:

$$\textit{point cible stable} : \bar{X} \in \mathcal{X} \textit{ tel que } \mathcal{F}(\bar{X}) = \bar{X}$$

L'état courant X_t du système à l'instant t n'est pas nécessairement connu : il est possible qu'une observation du système soit imparfaite et n'apporte que des informations partielles sur cet état. On note Y_t l'observation du système au même instant t. Seule cette information est disponible pour décider comment contrôler le système.

B.2.2.2 Objectif du contrôle

Le problème du contrôle recouvre plusieurs objectifs différents.

Le premier est l'étude de la *contrôlabilité*, c'est-à-dire vérifier s'il est possible d'atteindre la cible $\bar{X}$ avec les moyens d'action à disposition, à partir de tout état $X_0 \in \mathcal{X}$. Formellement, et en se limitant au cas discret, la contrôlabilité s'écrit :

$$\forall X_0, \ \exists (A_t)_{t \in \mathbb{N}} \textit{ tel que } \exists t \in \mathbb{N} \textit{ tel que } X_t = \bar{X}$$

et il faut donc trouver cette suite de valeurs de paramètres $(A_t)_{t \in \mathbb{N}}$. Plus précisément, et en prenant en compte l'observation limitée du système, le choix de la valeur A_t à tout instant t dépend des observations $(Y_i)_{i \in [0,t]}$ effectuées jusqu'à t, et le problème s'écrit :

$$\forall t, \ \forall (Y_i)_{i \in [0,t]}, \ \textit{trouver} (A_j)_{j \in [t,\infty]} \textit{ tel que } \exists t' \in \mathbb{N} \textit{ tel que } t' \geq t \textit{ et } X_{t'} = \bar{X}$$

Le second objectif est l'*optimisation* du contrôle. Si l'on se donne une mesure de performances, qui représente par exemple le temps nécessaire pour atteindre la cible, il s'agit de trouver une suite de valeurs $(A_t)_{t\in\mathbb{N}}$ similaire, qui optimise cette mesure.

Enfin, dans le cadre de l'automatique, l'objectif de *régulation* correspond non seulement à l'obtention de la cible, mais également à sa stabilisation face à des perturbations.

B.2.2.3 Comparaison de l'observation et de la cible

Une étape clef, en général passée sous silence car peu difficile, est de comparer la connaissance issue de l'observation de l'état courant du système à la cible ou à la consigne. Cela sert d'une part à évaluer le contrôle, en sachant quand la cible est atteinte, et d'autre part à guider les actions de contrôle en déterminant A_t à tout instant t, en fonction de l'écart entre l'état et la cible. La figure 3 présente un exemple typique de régulation de vitesse qui utilise cette différence.

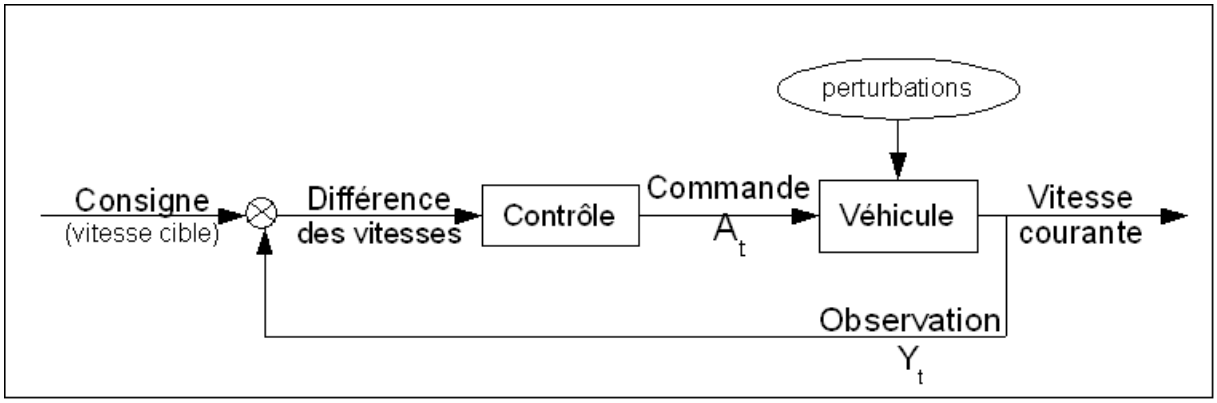

Figure 3: Contrôle de la vitesse d'un véhicule, en mesurant la différence entre la vitesse courante et la consigne.

B.2.2.4 Résolution du contrôle

Pour trouver la suite de valeurs $(A_t)_{t\in\mathbb{N}}$ qui répond à l'objectif de contrôle fixé, trois questions se posent, bien que seule la dernière pose réellement problème en général :

- trouver l'information utile pour le contrôle,
- sélectionner les types de commandes ou les paramètres qui permettent de contrôler,
- choisir la commande A_t en fonction de l'information retenue.

Le but est de trouver une politique de contrôle, c'est-à-dire une fonction qui, à tout instant t, associe une commande A_t aux observations du système $(Y_i)_{i\in[0,t]}$ effectuées jusqu'à t. Cette politique doit permettre d'atteindre l'objectif fixé.

Toutes les observations ne sont pas nécessairement utiles, et le premier point consiste à trouver les antécédents qui rendent cette fonction plus simple tout en conservant sa capacité à atteindre l'objectif. Dans l'exemple de la figure 3, seule la différence instantanée des vitesses est prise en compte. Mais en fonction du système étudié, d'autres indications sur l'état courant peuvent être utiles, de même qu'une certaine mémoire sur les états passés.

De la même manière, tous les paramètres ne sont pas utiles pour atteindre la cible, et la politique peut être simplifiée si l'on supprime ceux qui ne sont pas nécessaires. Toujours dans le même exemple, on peut supposer que les commandes sont l'accélération et le freinage du véhicule. Même s'il est possible de commander sa direction, cela n'a a priori pas d'intérêt pour obtenir la vitesse cible.

Enfin, il faut calculer la réponse à apporter, au niveau des commandes, à une observation donnée. Pour cela, on travaille souvent de façon analytique, à partir d'un modèle du système étudié, typiquement un système d'équations différentielles linéaires. On modifie ensuite mathématiquement ce modèle, par exemple en inversant la matrice représentant le système d'équations (en théorie du contrôle) ou en calculant la fonction de transfert du système c'est-à-dire sa transformée de Laplace (en automatique). Ces outils mathématiques permettent finalement de déterminer formellement les valeurs de paramètres donc la commande A_t.

B.2.2.5 Insuffisance de cette approche pour le contrôle d'un SMA

Les méthodes de contrôle analytiques, théorie du contrôle et automatique, traitent peu des systèmes non-linéaires. Les outils qu'elles proposent s'appliquent souvent à des systèmes soumis à de fortes hypothèses, et si leur utilisation en dehors de ce cadre est le sujet de nombreuses études ([Faubourg 01], [Rondepierre 06]), « dans le cas non linéaire, il faut en général fortement limiter ses ambitions » [Manneville 03].

L'approche des systèmes non linéaires par la théorie du contrôle passe habituellement par une linéarisation locale des équations du système $\mathcal{F}$ autour de la cible [Khalil 02]. Cette approche peut être améliorée par l'utilisation de réseaux de neurones formels [Lehalle 05]. Quand à l'automatique, elle traite le plus souvent de systèmes linéaires simples, qui permettent de calculer puis d'utiliser la fonction de transfert du système. Dans les cas non linéaires, des outils plus complexes sont mis en oeuvre, qui rejoignent la théorie du contrôle.

Telles qu'elles sont utilisées et étudiées traditionnellement, ces approches présupposent qu'il est possible de déterminer sans difficulté l'effet d'une modification du système sur son comportement, alors qu'il s'agit justement du point crucial de l'étude des SMA. La difficulté du contrôle d'un SMA est donc liée à son caractère non-linéaire et imprévisible. Wegner démontre dans [Wegner 97] qu'un système basé sur les interactions ne peut pas être parfaitement modélisé analytiquement, ce qui achève d'écarter les approches analytiques que nous venons de présenter.

Cependant, elles offrent un cadre de réflexion dont nous pourrons nous inspirer pour répondre à la problématique d'assurer un comportement global dans un SMA.

B.2.3 Contrôle des comportements individuels dans un SMA

Nous discutons ici d'approches comme [Scherrer 04] qui traitent du *contrôle optimal d'agents*. Elles cherchent à diriger chaque agent en définissant son comportement individuel. Elles se situent dans un cadre formel, qui restreint les comportements possibles des agents à une classe fixe de comportements, ce qui permet de calculer un comportement optimal dans cette classe par rapport à un critère donné.

Leur objectif est de fournir aux agents une *politique*, c'est-à-dire une fonction qui correspond à leur comportements individuels. Cette fonction associe une action à chaque état

représentant la situation dans laquelle se trouve l'agent à un instant donné, telle qu'il peut la percevoir. Cette politique est optimisée par rapport à un objectif à atteindre, estimé par ailleurs.

La nuance avec les méthodes de construction d'un SMA est subtile, puisque dans les deux cas, on cherche à définir le comportement des agents au niveau local pour obtenir des propriétés attendues au niveau global. Là où les méthodes de construction définissent le comportement des agents par des valeurs de paramètres, ces même comportements sont ici définis en extension, c'est-à-dire par une description précise des réactions à chaque situation.

La quasi-totalité de ces méthodes est liée à l'utilisation de processus de décision markoviens (MDP) pour modéliser le système, même si d'autres solutions sont avancées, comme dans [Lee 04]. C'est cette majorité que nous présentons.

B.2.3.1 Présentation et utilisation des MDP

Les notions suivantes sont classiques et traditionnellement admises. Pour plus de détails, le lecteur pourra se référer par exemple au livre [Sutton & Barto 98].

Un MDP est un problème qui englobe à la fois le modèle de la dynamique d'un système – généralement un agent – et ce qu'on en attend. Formellement, il s'agit d'un quadruplet <S, A, T, R> où S est un ensemble d'états et A un ensemble d'actions. T est une fonction de transition qui donne la probabilité T(s, a, s') pour le système d'arriver dans l'état s'∈S quand l'action a∈A est effectuée alors que le système se trouve dans l'état s∈S. Le nombre de transitions augmente rapidement quand le MDP devient grand : il est proportionnel au nombre d'actions et au carré du nombre d'états. La figure 4 illustre les huit transitions possibles pour un MDP à deux états et deux actions. Enfin R est une fonction de récompense qui associe un bonus ou un malus R(s, a, s') à la transition qui fait passer de l'état s à l'état s' en effectuant l'action a.

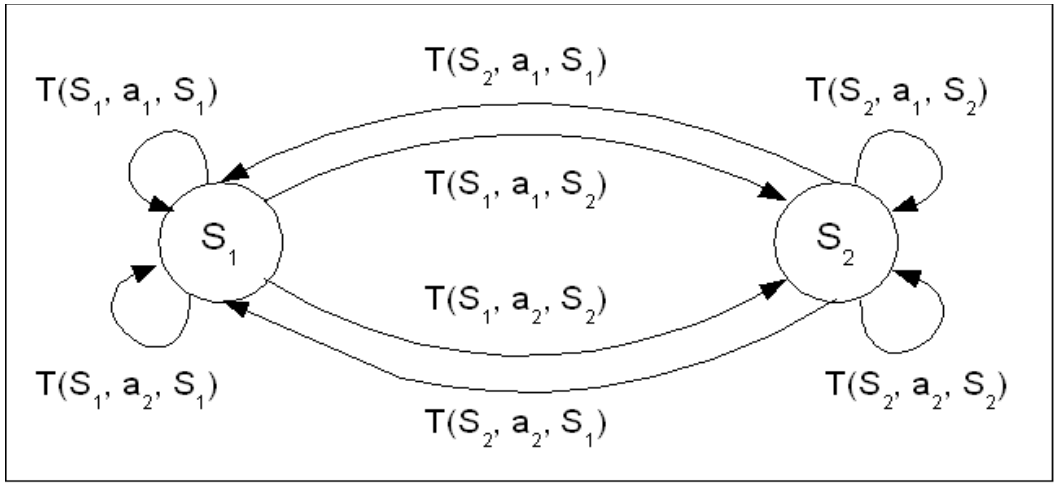

Figure 4: Transitions dans un MDP avec seulement deux états (S1 et S2) et deux actions (a1 et a2)

La particularité de ce modèle est qu'il respecte la propriété de Markov, qui stipule que le résultat d'une action ne dépend que de l'état courant du système, ce que l'on peut vérifier

dans la définition de T. C'est une hypothèse a priori forte, mais qui se révèle souvent une bonne approximation. Le problème d'un MDP est de trouver une politique, c'est-à-dire une fonction qui associe une action à chaque état, telle que la récompense soit maximisée lorsqu'elle est appliquée. Formellement, on définira une politique déterministe par une fonction de S → A, et une politique stochastique par une fonction de SxA → [0,1] qui associe une probabilité d'effectuer une action dans chaque état.

La résolution d'un MDP est une question très étudiée, dont la solution passe souvent par des outils d'apprentissage par renforcement [Sutton & Barto 98]. Lorsque les fonctions T et R du système étudié ne sont pas connues, certains de ces outils, comme les algorithmes Q-learning et Sarsa, permettent de calculer la politique optimale expérimentalement. Ils permettent également de réduire le nombre d'expériences nécessaires en privilégiant dans chaque état les actions les plus prometteuses[1] et en se reposant sur des principes de programmation dynamique.

B.2.3.2 Application des MDP aux SMA

Dans l'étude des SMA, on peut représenter chaque agent par un MDP, on parle alors de MDP décentralisé ou DEC-MDP pour représenter l'ensemble du système. Dans cette optique l'objectif est de définir une politique optimale pour chaque agent, qui dépend de la politique de tous les autres agents. On choisit directement l'action à associer à un état possible pour un agent. Le critère d'optimalité, qui définit la fonction R pour chaque agent, est global et correspond au comportement du SMA que l'on souhaite atteindre. On cherche à déterminer les comportements locaux des agents pour atteindre ce critère global du système.

Par exemple, [Chades 02] propose cette approche pour résoudre un problème de poursuite, dans lequel des prédateurs doivent encercler une proie au comportement aléatoire. Pour atteindre cette cible, les comportements des prédateurs sont définis par une politique qui associe une direction de déplacement en fonction de la perception de l'agent.

Mais la détermination d'une politique optimale est un problème complexe lorsque le nombre d'agents augmente. Les DEC-MDP sont des problèmes NEXP-complet [Goldman 2004], généralement étudiés avec peu d'agents, et ne peuvent techniquement pas être appliqués à des SMA à grande échelle. Divers travaux, comme [Szer 06] et [Aras 07], traitent de cette complexité et de techniques d'approximation pour résoudre ce problème.

La difficulté du lien entre les niveaux local et global se traduit aussi par le problème de la répartition de la récompense (globale) entre les agents (locaux), appelé *credit assignment* (voir [Agogino 04]).

Le lien entre le changement des comportements individuels au niveau local et le comportement global du SMA est appris au travers des nombreuses expériences réalisées pour trouver un ensemble de politiques d'agents proche de l'optimal. Comme pour la calibration, la principale difficulté reste la complexité algorithmique. La combinaison des espaces d'états de chaque agent, de même que la combinaison de leurs espaces d'actions, doivent être explorés. Leur taille est importante car les approches discutées ici se situent au niveau local du système.

---

1 Ce qui est comparable à l'utilisation de plan d'expériences dynamique sur le nombre de réplications (voir §B.1.2.3).

B.2.4 Approche de contrôle au niveau global

On trouve une proposition de contrôle au niveau global d'un SMA dans [Campagne 04a et b, Campagne 05]. Il s'agit de l'unique travail de recherche à traiter explicitement de ce problème que nous avons trouvé dans la littérature, c'est pourquoi nous le développons tout particulièrement.

B.2.4.1 Intuition du contrôle

L'auteur indique que « les recherches [...] qui s'intéressent à la construction [des SMA] se concentrent sur les méthodes de type génie logiciel et les méthodes d'analyse, mais le problème du contrôle n'est pas vraiment abordé. » Il ajoute que le contrôle est utile lorsqu'on constate qu'une méthode uniquement auto-adaptative n'est pas envisageable pour arriver au même résultat, c'est-à-dire si le comportement du SMA est susceptible de subir des dérives qui ne pourront pas être réparées par le seul comportement des agents. Il propose donc de réaliser un système dont le comportement est approximativement correct, puis de moduler ce comportement par un contrôle extérieur.

B.2.4.2 Proposition

Ce travail propose d'atteindre et de maintenir un comportement global. L'organisation globale des agents est représentée à l'aide d'une analyse morphologique, afin d'obtenir une *forme*, et cette forme est supposée corrélée au comportement du SMA. Elle résulte de l'agrégation d'une ou quelques valeurs propres aux agent. Pour obtenir un comportement cible, l'auteur cherche à atteindre une forme particulière, en agissant directement sur la forme courante, donc sur les agents.

Le coeur de la proposition est de mesurer la forme courante, et de la rectifier lorsqu'elle n'est pas adéquate pour se rapprocher d'une forme correspondant à un meilleur comportement. Elle associe à la forme courante présentée par le SMA une éventuelle autre forme plus adéquate, et modifie directement le comportement du système.

Le choix de la forme se fait ainsi : si la forme actuelle est considérée comme mauvaise, on sélectionne la forme considérée comme bonne la plus proche. Les informations qui associent une valuation aux formes et définissent une distance entre elles sont issues d'un modèle de la dynamique du SMA. Ce modèle est généré à partir du système grâce à son observation et à des expériences empiriques (voir figure 5). Cette construction est fondée sur l'observation, avec une intervention humaine importante, puisque l'utilisateur intervient dans le classement des formes considérées comme bonnes et mauvaises. Cela ne garantit pas l'exactitude du modèle, mais permet à la méthode d'être opérationnelle, contrairement à l'utilisation de méthodes formelles d'après l'auteur.

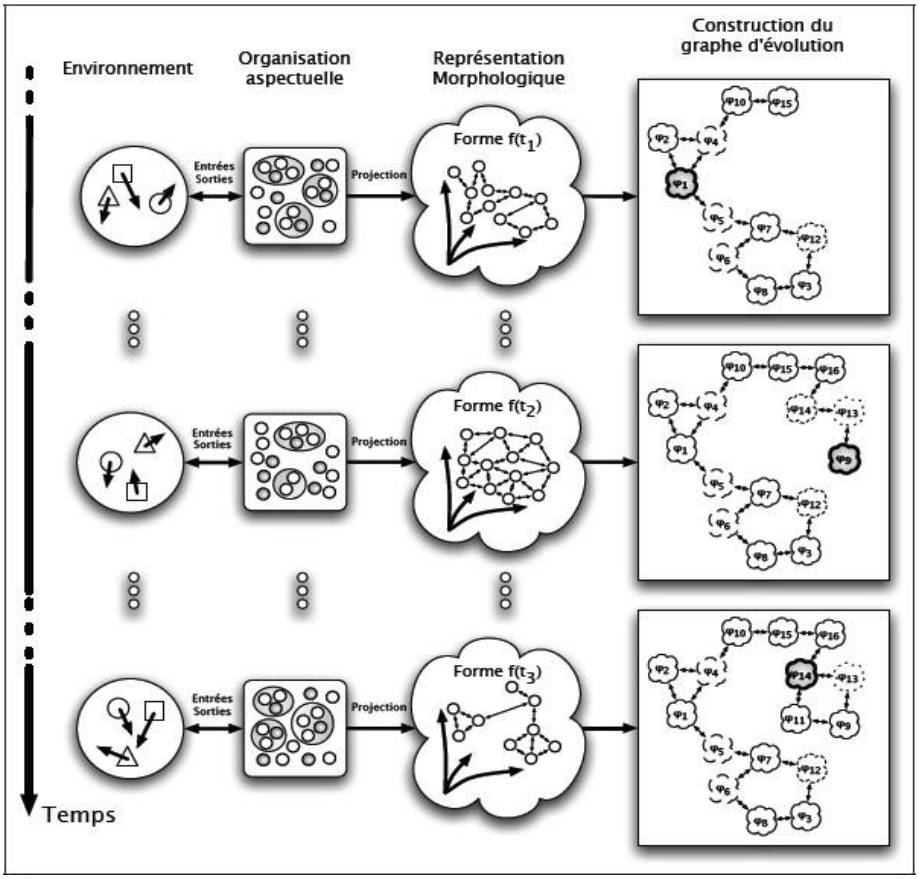

Figure 5: Construction d'un modèle de la dynamique du SMA, sous forme d'un graphe représentant les transitions entre les formes. Issu de [Campagne 05], page 88.

B.2.4.3 Application

Le système étudié dans ces travaux est composé d'une population au sein de laquelle des rapports de force gouvernent le transfert d'énergie entre individus. Lorsqu'un agent refuse de transférer une partie de son énergie à un autre, il y a éventuellement un combat qui fait diminuer l'énergie des deux individus. Un individu sans énergie meurt, et l'objectif est de conserver une population stable, en nombre et en distribution d'énergie entre les individus.

Pour cela, un contrôleur peut transférer artificiellement de l'énergie d'un agent à un autre, en fonction de l'état du SMA supposé connu à tout instant. Dans ce système, la forme est un histogramme représentant la répartition de l'énergie entre les agents. Les actions du contrôleur permettent donc de passer d'une forme à une autre.

B.2.4.4 Discussion

L'hypothèse qu'il est possible d'agir directement sur les formes, donc sur le comportement global, semble un peu forte. En effet, dans un cas général, une action sur un SMA possède une influence au niveau local. Par exemple, il s'agit de définir un comportement individuel, ou de fixer des valeurs de paramètres. Or l'influence d'une telle action sur le comportement global est difficilement prévisible, à cause du caractère émergent du comportement global.

Une autre hypothèse de cette approche est restrictive, celle selon laquelle la forme est définie comme une agrégation de valeurs propres aux agents, comme l'histogramme des énergies des agents. Dans un SMA quelconque, le comportement global peut être plus difficile à caractériser, comme l'indique [DeWolf 05c].

Finalement, cette approche pose les bases d'une modélisation de la dynamique globale d'un SMA pour le contrôler. Elle fonctionne sous des hypothèses fortes, et nous souhaitons la généraliser.

C. Bilan

La problématique à laquelle nous nous intéressons est la maîtrise du comportement global d'un SMA. L'objectif est d'assurer le comportement pour lequel le système est construit. Nous cherchons l'influence que doit exercer un utilisateur du SMA pour que le comportement global qu'il observe soit satisfaisant. Ce problème est difficile car le SMA est spécifié au niveau local, et c'est à ce niveau que s'exerce l'influence de l'utilisateur. Le comportement global du système n'est pas directement maîtrisé, et difficilement expliqué, à cause de son caractère émergent.

Cette problématique est liée à deux problèmes annexes : identifier le comportement global d'un SMA, et vérifier que la maîtrise est bien effectuée quelles que soient les conditions d'utilisation du SMA. La difficulté est à chaque fois de lier le niveau global du comportement au niveau local des influences de l'utilisateur, des observations, et des conditions d'utilisation.

On trouve dans la littérature des approches variées pour répondre à la problématique, preuve de sa difficulté. Elles se répartissent en deux principes différents mais non exclusifs :

- Assurer que le SMA présente un comportement spécifié lors de sa construction, ce que nous avons appelé les approches par construction.
- Assurer ce comportement lors de l'évolution du SMA, à l'aide d'une approche par contrôle. Dans ce cas, l'utilisateur influe dynamiquement sur un SMA existant pour diriger son comportement global vers un comportement désiré.

Lorsqu'un SMA peut être construit de toutes pièces, les méthodologies de conception indiquent des pistes à suivre pour faciliter la création du système, et s'approcher progressivement des spécifications. Ces méthodologies s'arrêtent une fois qu'il est possible de certifier que, dans des conditions d'utilisation données, le SMA présente toujours un comportement désiré.

Quand la structure d'un SMA est construite, mais que les valeurs de ses paramètres doivent encore être déterminées de manière à obtenir un comportement voulu, la calibration des paramètres permet de trouver ces valeurs. Les paramètres sont optimisés en moyenne par rapport aux conditions d'utilisation, à l'aide d'outils efficaces, qui trouvent une solution approchée si la complexité du problème est trop élevée.

Le contrôle est originellement un principe analytique qui s'applique à des systèmes dynamiques simples. Or les outils analytiques classiques ne conviennent pas aux SMA à cause de la complexité de ces derniers, surtout s'ils présentent des comportements émergents.

Les outils d'apprentissage par renforcement sont à la base de la majorité des approches de contrôle du SMA à son niveau local. Ces approches cherchent à définir de manière fine les comportements des agents qui permettent d'obtenir le comportement global voulu.

Le contrôle d'un SMA à son niveau global, quant à lui, a été très peu étudié, et uniquement sous des hypothèses restrictives. Son principe est d'agir sur le SMA en fonction de l'observation de son évolution globale par un utilisateur.

Les approches par construction présentent l'avantage de la simplicité. Elles permettent d'optimiser le comportement du SMA en moyenne par rapport aux conditions d'utilisation. Mais elles se révèlent insuffisantes si le comportement varie en fonction de ces conditions.

Les approches de contrôle au niveau local bénéficient de techniques formelles éprouvées, mais posent un problème de complexité. En effet, les espaces à explorer expérimentalement sont souvent très vastes, car les états et les actions définis au niveau local sont nombreux.

Le positionnement global du contrôle permet de réduire la complexité de l'approche, même dans le cas de SMA composés de nombreux agents. La difficulté est alors de déterminer les informations globales à considérer pour décider comment agir sur le système. En outre, la seule étude qui traite de cette approche se place sous des hypothèses restrictives comme la possibilité de modifier directement et à volonté le comportement global du SMA.

Nous souhaitons étudier ce que le contrôle au niveau global peut apporter à la maîtrise d'un SMA, en utilisant des techniques expérimentales éprouvées de contrôle optimal. Ainsi, nous nous positionnons entre l'idée du contrôle au niveau global d'un SMA avancée par Campagne, et l'utilisation de techniques utilisée habituellement dans un cadre formel pour le contrôle au niveau local d'un SMA. Nous cherchons également à automatiser le calcul d'un contrôle optimal, ce en quoi nous nous démarquons des approches par méthodologie de conception.

CHAPITRE III - CONTRÔLE PAR MODÉLISATION DE LA DYNAMIQUE GLOBALE DU SMA

Dans cette partie nous présentons les fondements de notre proposition pour le contrôle d'un SMA. Celle-ci consiste à modéliser la dynamique globale du SMA et à la représenter sous forme d'un graphe d'états dont les transitions correspondent aux actions de contrôle.

Plusieurs étapes sont nécessaires : décider des sommets du graphe en fonction des comportements observés et de la cible, apprendre les transitions, et enfin exploiter cette modélisation pour contrôler le système.

A. Problématique de contrôle d'un SMA

La définition du contrôle d'un SMA est développée ici, en s'appuyant sur celle du contrôle d'un système dynamique.

A.1 Cadre et données du contrôle

Nous faisons l'hypothèse que le SMA étudié admet plusieurs comportements globaux identifiables. Celui qu'il présente à un instant donné peut être souhaité ou non souhaité. L'ensemble des comportements désirés est appelé la cible du contrôle. Nous supposons que la régularité des comportements globaux permet de prévoir, au moins en partie, l'évolution du SMA. Il est ainsi possible de choisir de façon rationnelle des moyens d'action à utiliser en fonction de l'évolution courante et observée du système.

L'objectif du contrôle est d'agir sur le SMA pour sortir d'un comportement non souhaité et revenir à la cible. Cela suppose l'existence de moyens d'action sur le système, dont l'utilisateur peut se servir pour le ramener vers un comportement favorable. Cela suppose également que l'utilisateur doit avoir accès à des moyens d'observation qui lui fournissent des indications sur l'état du SMA, pour pouvoir décider de l'action à effectuer.

Le problème du contrôle d'un SMA est finalement énoncé par la donnée :

- D'une *cible*. Les comportements globaux du système décrivent des phénomènes émergents stables. Cela est cohérent avec l'idée d'un point fixe cible de l'espace des phases dans le contrôle d'un système dynamique. L'ensemble des comportements cibles est considéré invariable dans le temps.
- De *moyens d'observation*. Ils apportent la connaissance nécessaire pour décider comment agir sur le SMA. Ils peuvent être limités à cause de la décentralisation du système.
- D'un ensemble $\mathcal{A}$ de *moyens d'action* sur le SMA, qui permettent d'influer sur son comportement. Ces moyens d'action possèdent un sens au niveau global du contrôleur : on peut leur associer une durée ou un coût. Mais leur influence se fait au niveau local du SMA, où est définie la dynamique du système. Elles dépendent éventuellement de son état courant.

A.2 Objectif du contrôle

En parallèle avec les systèmes dynamiques, nous appelons état d'un système multi-agents et nous notons X la description du SMA à un instant donné. Nous appelons $\mathcal{X}$ l'ensemble des états possibles du SMA. Nous notons $\mathcal{F}$ la fonction qui recouvre l'ensemble des règles d'évolution du SMA. Elle se situe au niveau local des agents ou de l'environnement.

Comme pour un système dynamique, on appelle Y_t l'observation de l'état X_t à un instant t. L'objectif du contrôle est de déterminer les actions $(A_t)_{t\in\mathbb{N}}$ à effectuer sur le SMA pour passer d'un comportement non cible à un comportement cible, en fonction des observations Y_t successives. La figure 6 montre la place du contrôle dans la dynamique du système. Une action a une influence au niveau local du SMA, soit sur son état X, soit sur son évolution $\mathcal{F}$.

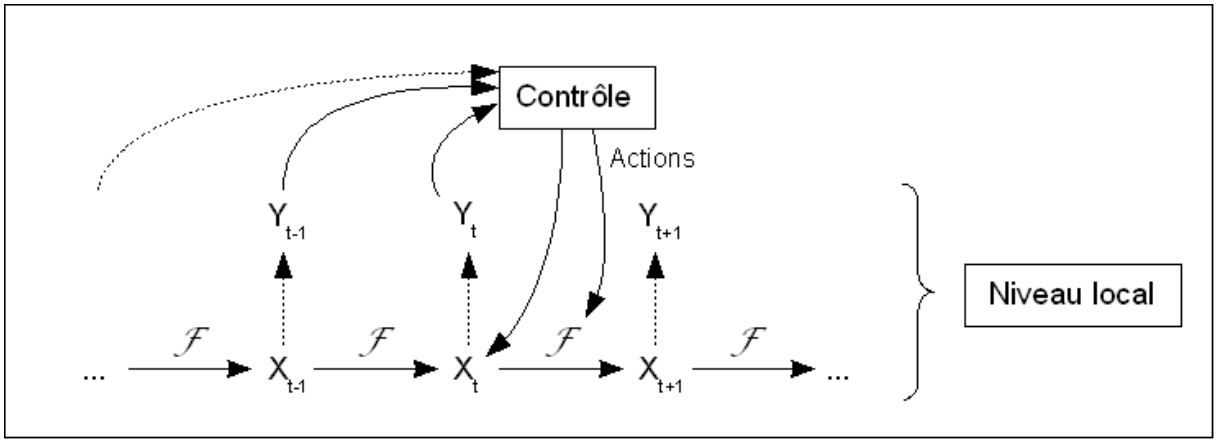

Figure 6: Contrôle du SMA à un instant donné, en ne considérant que son niveau local

Nous appelons les actions $A_t \in \mathcal{A}$ sur le SMA des *actions de contrôle*. Appliquer une action de contrôle revient à utiliser l'un des moyens d'action à disposition. Quand le contrôleur sélectionne une action de contrôle à effectuer, il s'agit d'une modification définie au même niveau que le comportement global qu'il observe. La figure 7 fait apparaître ces actions de contrôle et les autres données du problème, leur relation au mécanisme de contrôle, et leur caractère global ou local. La modification du SMA par une action de contrôle dépend éventuellement de son état courant.

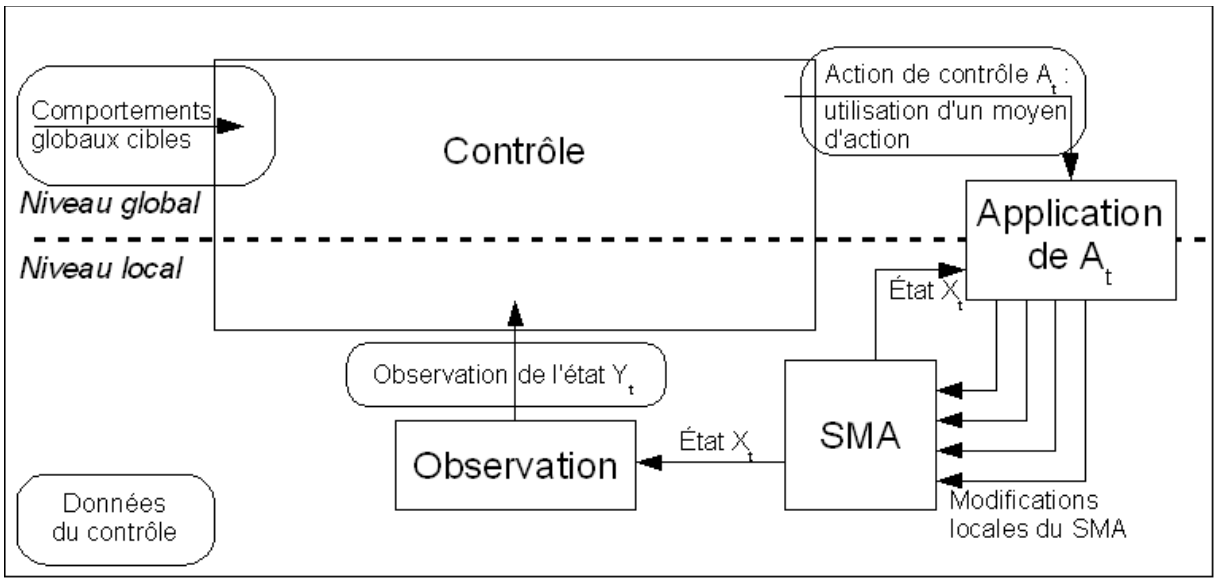

Figure 7: Articulation des données du contrôle d'un SMA

On appelle *simulation* l'évolution du SMA à partir d'un état initial X_0, en appliquant des actions de contrôle $(A_t)_{t \in \mathbb{N}}$, jusqu'à ce qu'un comportement cible soit atteint.

Trois objectifs d'étude ont été proposés pour le contrôle d'un système dynamique : la contrôlabilité, l'optimisation et la régulation. Nous en proposons une définition précise du point de vue du contrôle global d'un SMA.

La *contrôlabilité* consiste à faire présenter et maintenir un comportement cible par le SMA à partir de n'importe quel état initial, ou lorsqu'il se trouve dans un comportement non cible suite à une perturbation. Étant donné que les comportements globaux du SMA apparaissent stables, nous supposons qu'une fois un comportement cible atteint, il n'y a plus besoin d'agir sur le SMA... Du moins jusqu'à ce qu'une perturbation le fasse à nouveau sortir de ce comportement. Par conséquent, l'objectif de contrôlabilité se limite à atteindre une fois la cible pour toute simulation.

Par sa nature dynamique, le contrôle assume la charge de *régulation* du SMA : il permet de revenir à un comportement cible dès que le système s'en écarte. Néanmoins, si la dérive du comportement est due à des perturbations, il est nécessaire que le contrôle intervienne à un rythme plus élevé que celui des perturbations.

Pour *optimiser* le contrôle, il faut considérer une mesure de performance d'une simulation. On cherche alors la suite d'actions de contrôle qui optimise cette mesure en fonction de la suite d'observations $(Y_t)_{t\in\mathbb{N}}$, tout en permettant d'atteindre la cible. La mesure représente par exemple le temps moyen sur les simulations nécessaire pour atteindre la cible, ou tout autre type de coût, ce qui peut également inclure le coût des actions elles-mêmes.

L'objectif du contrôle dépend du SMA et du problème de contrôle considérés. Toutefois, comme nous essayons d'atteindre un comportement cible là où une approche par construction n'assure pas ce comportement, l'objectif prioritaire est la contrôlabilité.

A.3 Questions-clefs pour la résolution du contrôle d'un SMA

Le choix d'une action de contrôle A_t en fonction de la suite des observations $(Y_i)_{i\in[0,t]}$ de l'état du SMA lors de sa simulation se heurte à deux difficultés.

La première est l'existence de deux niveaux dans le SMA. Le comportement et les moyens d'action sont définis au niveau global, tandis que l'influence des actions et l'observation se situent au niveau local. Or nous avons vu au chapitre II que le lien entre les deux est difficile à établir. En particulier, il n'y a aucune certitude sur l'effet d'une action sur le comportement.

La seconde difficulté est la multiplicité des états du SMA au niveau local, qui donne lieu à de nombreux moyens d'action A_t et de nombreuses observations Y_t, donc à une complexité algorithmique importante pour l'exploration de ces espaces.

Pour répondre à ces difficultés, l'état de l'art incite à utiliser la connaissance heuristique que l'on a du SMA, ainsi que des outils expérimentaux qui estiment statistiquement l'effet des différentes actions.

Ces difficultés se retrouvent dans différents aspects du contrôle, exposés dans la figure 8. Les blocs intitulés *mesure du comportement* et *sélection, mémoire et temporalité des informations* sont chargés du passage de l'observation locale à des concepts globaux. Le bloc *choix d'action* correspond au contrôle proprement dit : il s'agit de déterminer une action de contrôle à effectuer parmi les moyens d'action en fonction des informations disponibles sur le SMA si le comportement courant n'appartient pas à la cible.

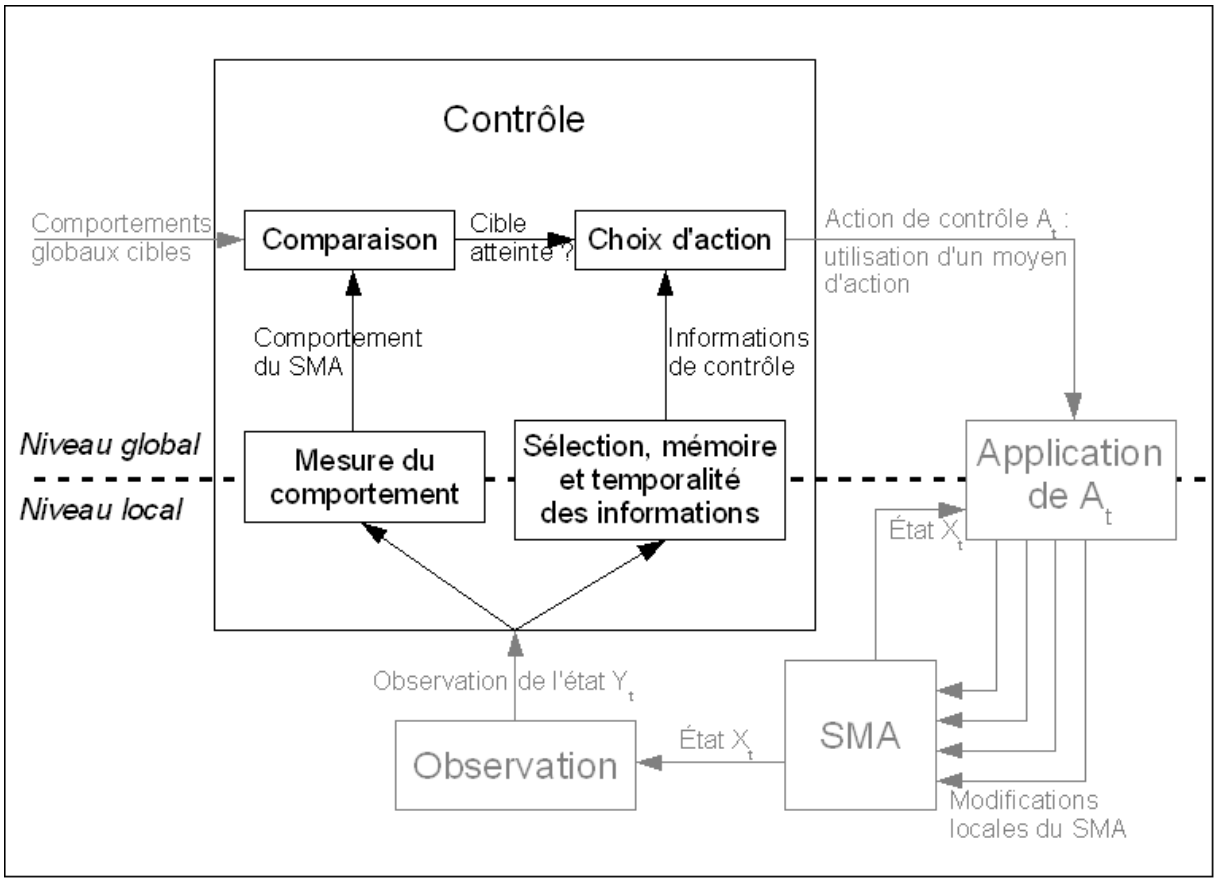

Figure 8: Les questions-clefs du problème de contrôle d'un SMA, et leur place dans la boucle de contrôle (en grisé)

L'étape de comparaison entre l'état courant et la cible correspond à la question similaire pour le contrôle d'un système dynamique. Associée à l'identification du comportement global, elle permet de savoir quand arrêter une simulation, de déterminer si une action de contrôle a eu ou non une bonne influence sur le comportement du SMA, et d'évaluer la contrôlabilité du SMA comme l'optimisation de son contrôle.

Le choix d'une action de contrôle à effectuer parmi tous les moyens d'action disponibles se fait en fonction d'informations utiles et discriminantes sur l'état du SMA. Nous les appelons les informations de contrôle. Ces informations de contrôle sont globales à la fois en espace, puisqu'elles sont sélectionnées sur l'ensemble du SMA, et en temps, puisqu'elles peuvent être gardées en mémoire et influer sur le choix des actions avec un certain retard. En outre, elles possèdent une temporalité qui indique le moment où une action doit être effectuée, autorisant un contrôle plus lent que l'évolution du SMA.

Pour choisir une action de contrôle de façon rationnelle, il faut connaître son influence probable sur le comportement du SMA en fonction des informations de contrôle disponibles. Dans une démarche expérimentale, cela signifie qu'il faut explorer simultanément l'espace des actions et celui des informations de contrôle possibles. Comme l'effet d'une action à partir de ces informations n'est pas prévisible avec précision, il faut en plus répéter chaque expérience pour connaître l'influence statistique de chaque action. Le contrôle nécessite donc une méthode d'exploration efficace pour limiter sa durée.

B. Proposition

Afin de contrôler un SMA, nous proposons de modéliser sa dynamique globale, c'est-à-dire l'effet de chaque moyen d'action à partir de chaque comportement global du système. Ce modèle est ensuite utilisé pour choisir des actions de contrôle en fonction du comportement courant.

La modélisation est réalisée par apprentissage par renforcement, en considérant le SMA comme un MDP. La durée de ces techniques d'apprentissage par renforcement dépend du nombre d'états et d'actions dans le MDP. C'est pour réduire cette durée que nous choisissons de modéliser la dynamique du SMA au niveau global : les états du MDP, qui correspondent aux comportements globaux du SMA, sont peu nombreux.

Ce choix est conforté par l'observation des régularités au niveau global, favorables à la prévision partielle de l'évolution du SMA, en dépit de son caractère complexe et non linéaire.

Le coeur de la proposition est présenté dans un premier sous-chapitre. Le suivant est consacré aux hypothèses qui lui sont nécessaires, donc au cadre de la proposition. L'application de la proposition se fait en plusieurs étapes, qui correspondent à des problèmes-clefs du contrôle. Chaque étape est ensuite développée, avant d'expliquer comment les choix qui y sont faits sont évalués et éventuellement rectifiés.

B.1 Présentation générale

Pour contrôler un système, une connaissance de sa dynamique est nécessaire. Dans le cas d'un SMA, cette connaissance est collectée par une modélisation expérimentale, car il est difficile, voire impossible, d'en fournir un modèle analytique.

Nous proposons de réaliser un contrôle en deux phases :

- La première phase, que nous nommons apprentissage, sert à construire expérimentalement un modèle de la dynamique globale du SMA.
- La seconde phase, le contrôle proprement dit, consiste à exploiter le modèle ainsi créé pour choisir les actions de contrôle de façon rationnelle.

Pour la modélisation, nous proposons de représenter la dynamique du SMA sous la forme d'un graphe d'états. Chaque état du graphe correspond à l'un des comportement globaux du système, notés C, C' et C'' sur la figure 9. Nous modélisons ainsi sa dynamique globale. Les arêtes de ce graphe sont liées aux actions de contrôle, qui permettent de modifier le comportement du SMA. Pour conserver un vocabulaire homogène, nous appelons états de contrôle du SMA les états du graphe. Ces états de contrôle doivent être différenciés des états locaux X_t du SMA considéré comme un système dynamique.

Les états du graphe sont définis avant l'apprentissage. Le but de l'apprentissage est de calculer les transitions du modèle, c'est-à-dire d'estimer l'état obtenu quand une action est effectuée dans un état de départ. Ceci est fait de manière expérimentale, en partant d'un état, en appliquant une action, et en identifiant l'état dans lequel on aboutit à l'aide d'une mesure adéquate. L'influence de chaque action dans chaque état doit être évaluée.

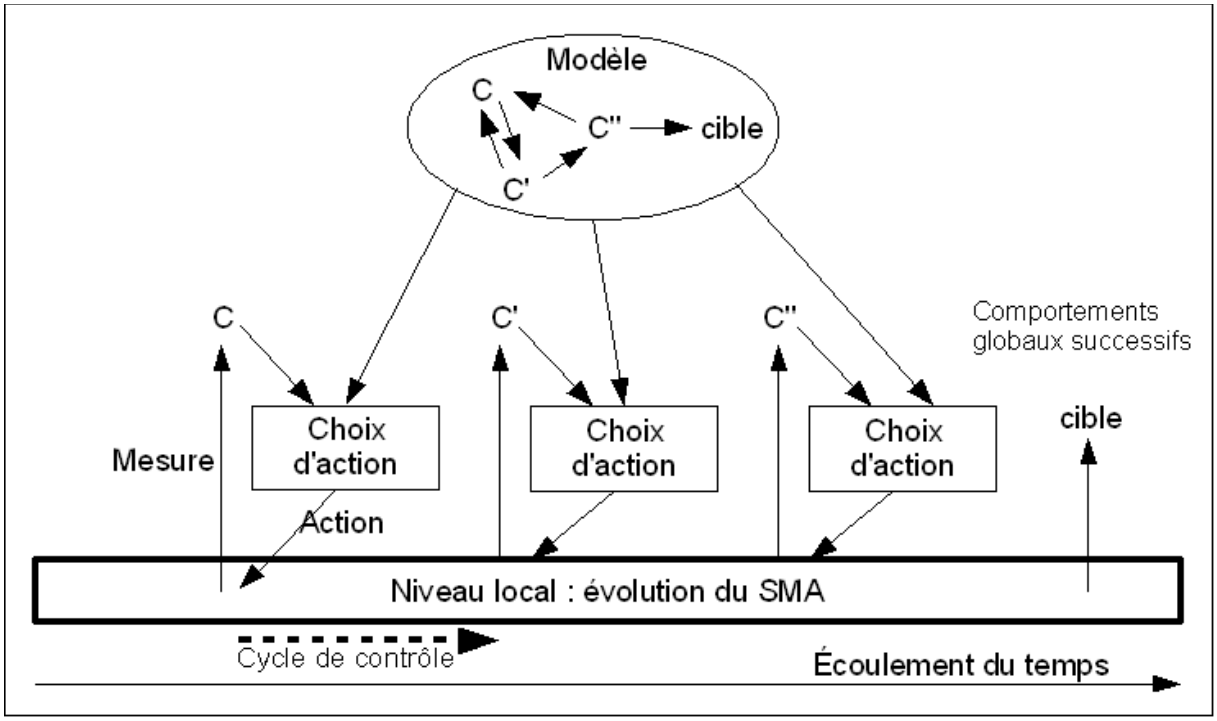

Figure 9: Proposition. Une action de contrôle est choisie en fonction du comportement courant mesuré et d'un modèle expérimental de la dynamique du SMA

Dans la phase d'exploitation, cette connaissance accumulée est utilisée pour contrôler le SMA. Cette phase est donc composée de deux étapes qui se répètent en boucle : identifier l'état courant du système, et choisir une action en conséquence grâce au modèle.

On appelle *cycle de contrôle* la partie de la simulation du SMA qui sépare deux actions de contrôle effectuées. Un cycle est donc composé des pas de simulation nécessaires pour observer la stabilité du comportement, c'est-à-dire un nouvel état courant du modèle.

La figure 10 montre le fonctionnement de notre proposition, en détaillant les cycles de contrôle lors des simulations d'apprentissage et d'exploitation. Elle peut être comparée à la figure 8 du chapitre précédent qui présentait le problème du contrôle d'un SMA de manière générale. La mesure du comportement global à partir de l'observation locale Y du SMA sert à déterminer à la fois si la cible est atteinte et l'état de contrôle courant du système. Au cours d'une simulation, une action est choisie en fonction de l'état courant, en suivant une politique. Lors de l'apprentissage, chaque action est évaluée en fonction de l'état auquel elle aboutit.

B.2 Cadre de la proposition

Nous nous intéressons à des SMA dont les observations successives Y_t suffisent à déterminer le comportement courant. Comme nous avons décidé de considérer les comportements comme les états du modèle, l'identification de l'état courant est nécessaire pour choisir une action de contrôle.

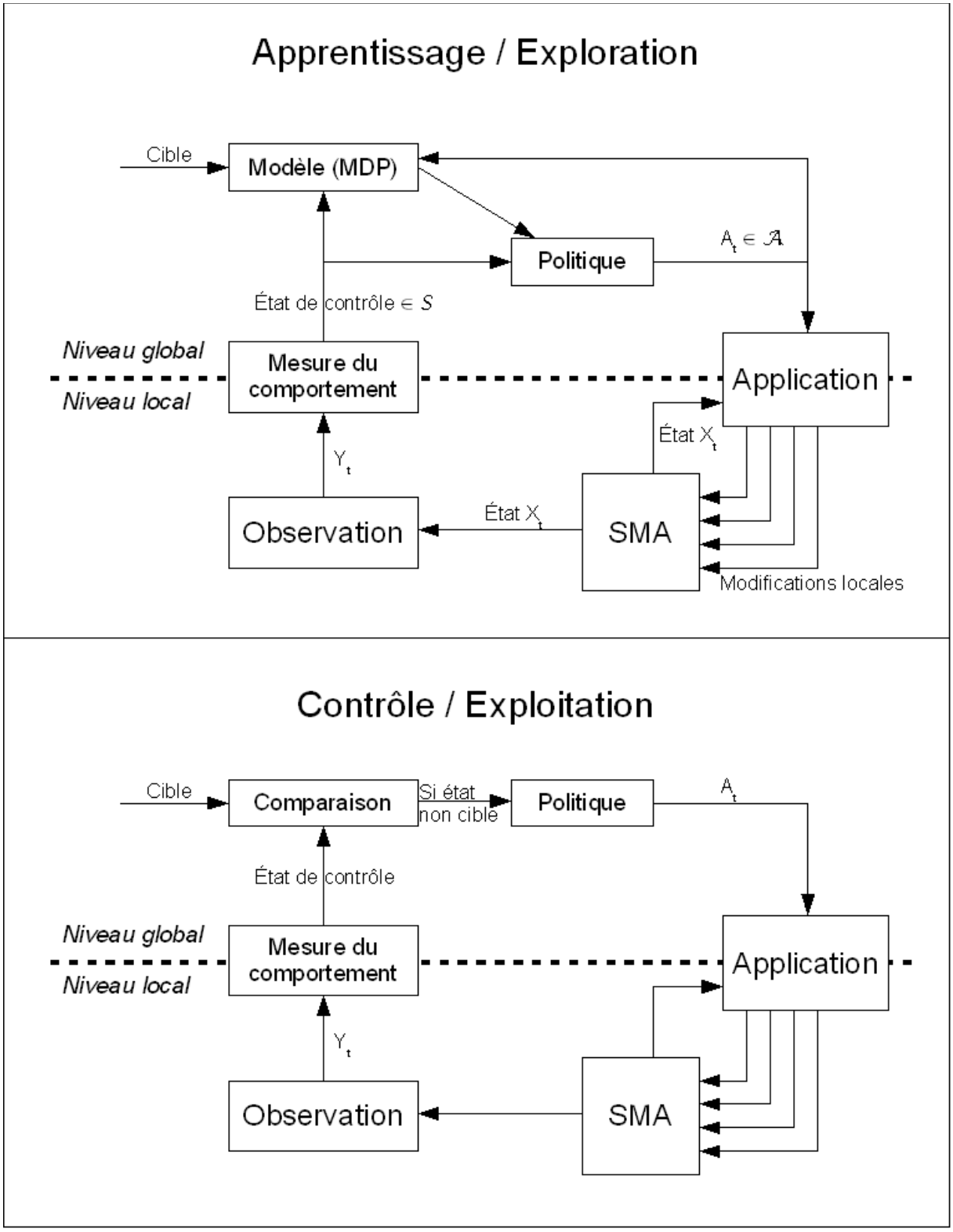

Figure 10: Détails de la proposition : cycles de contrôle lors de l'apprentissage et de l'exploitation. Un modèle markovien est utilisé pour associer une action de contrôle à un état, c'est-à-dire l'information utile pour le contrôle, qui est ici le comportement global du SMA.

Une seconde hypothèse est qu'à part le temps d'apprentissage lié à la complexité algorithmique, il n'y a pas de contrainte qui limite le nombre d'expériences qui peuvent être effectuées sur le SMA. Les ensembles d'états et d'actions peuvent être explorés à volonté.

Enfin, nous considérons des SMA qui possèdent peu de comportements globaux différents et peu de moyens d'action. Typiquement, la proposition s'applique plus facilement à un système réactif qu'à un système cognitif. Nous nous focalisons sur les aspects collectifs plutôt que sur les intelligences individuelles.

B.3 Étapes de la proposition

La mise en oeuvre de la proposition passe par trois étapes qui impliquent des choix pour un utilisateur humain :

- la caractérisation et la mesure du comportement,
- la détermination des informations utiles pour le contrôle,
- la méthode d'apprentissage.

Pour chacune d'elles, nous indiquons les choix qui se présentent lorsqu'il s'agit des les appliquer à un SMA donné, et les difficultés qu'elles peuvent poser.

B.3.1 Mesure du comportement global

Il faut caractériser le comportement global du SMA observé par un utilisateur à partir d'informations Y_t locales. La mesure du comportement dépend bien entendu du SMA étudié, et ne peut pas être généralisée d'un système à un autre, mais nous fournissons ici des spécificités qu'elle doit respecter. Cette mesure est utilisée pour :

- comparer le comportement courant et la cible lors de l'apprentissage, afin d'encourager les actions qui permettent d'atteindre cette dernière,
- identifier en temps réel l'état de contrôle du système, à la fois pour l'apprentissage du modèle et pour son exploitation,
- évaluer une solution de contrôle proposée, en vérifiant sur de nombreuses simulations si la cible est atteinte.

Lorsque le SMA est en évolution, il présente soit un comportement qui apparaît stable, c'est-à-dire avec une durée non négligeable au niveau global, soit un régime transitoire pendant lequel aucun comportement n'est identifiable. Le rôle de la mesure est de fournir des informations sur le comportement lorsqu'il est stable, et de ne donner aucun résultat dans le cas contraire. Elle possède donc une certaine dimension temporelle et prend en compte plusieurs observations Y_t successives du SMA pour vérifier la stabilité.

Pour choisir la mesure, il faut trouver un équilibre entre

- la justesse de son estimation du comportement, c'est-à-dire à la fois le fait qu'elle trouve le même comportement que celui identifié par un humain, et sa capacité à reconnaître un régime transitoire,
- et le temps nécessaire pour estimer ce comportement, dont dépend la durée de l'apprentissage et du contrôle.

Si la mesure du comportement est erronée, l'état de contrôle n'est pas correctement identifié, et l'action de contrôle qui en résulte risque d'éloigner le SMA du comportement cible. Il est donc préférable d'avoir une mesure qui donne des résultats sûrs à une mesure plus rapide mais approximative. De plus, il vaut mieux considérer à tort que la cible n'est pas atteinte plutôt que de surestimer les capacités du contrôle.

B.3.2 Choix des états de contrôle

L'objectif de cette étape est de choisir un **ensemble d'états** S du graphe qui modélise le SMA, à la fois suffisamment vaste pour autoriser un contrôle efficace, et suffisamment restreint pour limiter l'exploration donc la durée de l'apprentissage. Nous avons choisi de considérer comme état le comportement global courant du SMA. Il s'agit de ce que nous avons appelé au chapitre précédent *les informations de contrôle*. Plusieurs descriptions de ces comportements sont envisageables.

Le choix des informations à prendre en compte comme antécédents d'une politique est un problème classique. Des travaux comme ceux de Sertan Girgin [Girgin 08] proposent de découvrir automatiquement les meilleures informations possibles, à l'aide d'un algorithme génétique dans ce cas. Mais une telle solution systématique augmente la durée de l'approche, car un apprentissage doit être réalisé pour chaque ensemble S. Dans le cas d'un SMA, la durée de chaque simulation est élevée, ce qui nous pousse à réduire le nombre de simulations pour effectuer l'apprentissage. Nous ne pouvons donc pas nous permettre d'effectuer un grand nombre d'apprentissages pour trouver le meilleur ensemble d'états.

Nous choisirons un premier ensemble S d'états en fonction d'observations de la dynamique lorsque des actions sont effectuées. Cet ensemble sera éventuellement amélioré par la suite.

Nous introduisons un état particulier qui sera toujours présent dans le modèle, et que nous notons S_0. Il représente l'absence de comportement identifiable. Il permet de ne pas laisser le SMA présenter un comportement incertain indéfiniment, en effectuant une action de contrôle associée à cet état S_0. Il assure donc la terminaison d'un cycle de contrôle, même lorsque le comportement n'est pas assez stable ou n'est pas reconnu.

B.3.3 Méthode d'apprentissage

Le modèle que nous proposons de construire peut être considéré comme un processus de décision markovien[1]. Les états de ce MDP correspondent à l'ensemble S, et ses actions aux moyens d'action $\mathcal{A}$. Ses transitions sont inconnues a priori. Elles représentent la probabilité de passer d'un état à un autre en effectuant une action de contrôle, et seront estimées expérimentalement. Comme la cible est un sous-ensemble des comportements du SMA, c'est aussi un sous-ensemble des états du MDP. La fonction de récompense du MDP est donc facile à définir : elle donne une récompense fixe, par exemple 1, lorsque la cible est atteinte, et 0 sinon. L'arrêt des simulations lorsque la cible est atteinte assure que la récompense ainsi obtenue est toujours bornée, sans recourir à un horizon des récompenses (*discounted return*).

La résolution du MDP fournit une politique qui associe une action de contrôle à un comportement courant du SMA. Il existe des outils, comme ceux de l'apprentissage par

1 Voir page 47.

renforcement, qui permettent de calculer automatiquement cette politique. Ces outils permettent également de réduire le temps d'exploration pour la résolution du problème de contrôle. Ils explorent l'espace des actions et celui des états en suivant des principes similaires aux plans d'expériences dynamiques.

Un autre avantage des outils d'apprentissage par renforcement est qu'ils estiment le bénéfice à long terme de chaque action $a \in \mathcal{A}$ dans chaque état $s \in S$, noté Q(s,a). En effet, une bonne politique doit prendre en compte non seulement l'effet direct d'une action donnée dans un état donné, mais également son influence à long terme. C'est le cas en particulier s'il existe des états à partir desquels aucune action ne permet d'atteindre directement la cible.

L'apprentissage par renforcement suit et apprend une politique, qui dépend de l'estimation courante des Q(s,a) et d'un type de politique, c'est-à-dire une manière de choisir l'action à effectuer en fonction des valeurs Q(s,a). Une politique déterministe choisit toujours l'action qui maximise les Q(s,a) dans un état s. Si au contraire le contrôleur fait intervenir le hasard dans ce choix, elle est stochastique, par exemple :

- ε-gloutonne si elle choisit une action au hasard avec une probabilité ε, et reste déterministe avec une probabilité 1- ε,
- softmax si elle choisit aléatoirement les actions avec une distribution de probabilités de Boltzmann calculée à partir des valeurs Q(s,a)[1]
- proportionnelle si elle choisit les actions avec des probabilités proportionnelles aux valeurs Q(s,a).

Nous préconisons l'utilisation d'une politique d'exploitation stochastique. En effet, cela évite de persister inutilement à faire une action considérée comme bonne, en moyenne parmi toutes les situations locales que regroupe un état donné. Il pourrait alors exister certaines situations particulières auxquelles cette action ne changerait rien, ce qui résulterait en un blocage du contrôle.

Nous choisissons l'algorithme d'apprentissage Sarsa [Sutton & Barto 98], qui est adapté au calcul de politiques stochastiques, quand les transitions et les récompenses sont inconnues a priori. C'est le cas pour notre problème, et ces valeurs doivent être estimées par les résultats d'expériences. Il existe des améliorations à cet algorithme, comme l'algorithme Sarsa(λ), qui permettent entre autres d'augmenter sa rapidité pour arriver au même résultat. Dans ce document, toutefois, nous ne considérons que l'algorithme Sarsa originel.

L'apprentissage se fait sur un nombre de simulations déterminé en fonction de la complexité du problème, en particulier du nombre d'états et d'actions de contrôle. Nous proposons de choisir un petit nombre de simulations et de vérifier a posteriori s'il est suffisamment élevé pour que l'apprentissage n'améliore plus la politique de contrôle. Dans le cas contraire, une nouvelle série de simulations est ajoutée pour parfaire l'apprentissage. Pour assurer la terminaison rapide de chaque simulation, et donc limiter la durée totale de

---

1 La probabilité de choisir l'action a dans l'état s est alors $P(a) = \frac{e^{(\frac{-Q(s,a)}{T})}}{\sum_{a'} e^{(\frac{-Q(s,a')}{T})}}$ où T est un paramètre qui tend vers 0 au cours de l'apprentissage (voir [Dutech 03] ou [Sigaud 08]).

l'apprentissage, nous proposons aussi d'arrêter les simulations à un nombre maximal k de cycles. La valeur de k dépend du SMA étudié et elle est choisie empiriquement, en fonction de la connaissance que l'on possède du système.

L'algorithme ci-dessous détaille le déroulement de l'apprentissage, sans préciser le fonctionnement des outils d'apprentissage par renforcement.

```
Algorithme d'apprentissage
∀ état, ∀ action
    Q(état, action) ← 0
Répéter (nbSimulations)
    SMA.initialise()
    s1 ← SMA.identifieEtatCourant()
    // essayer d'atteindre la cible en moins de k cycles
    nbActions ← 0
    tant que (nbActions<k et s1 ≠ cible)
        // choix d'une action
        action ← politique([Q(s1, a1), Q(s1, a2), ..., Q(s1, aN)])
        SMA.applique(action)
        nbActions++
        nbPas ← 0
        // laisser le comportement du SMA se stabiliser
        Répéter
            SMA.pasDeSimulation()
            nbPas++
        jusqu'à (SMA.étatCourantIdentifié()
                        OU critèreArrêt(nbPas)) // l'état courant est alors s0
        // mise à jour des Q-valeurs avec Sarsa (ou autre algorithme)
        s2 ← SMA.identifieEtatCourant()
        sarsa(Q(s1,action), s2)
        s1 ← s2
    fin tant que
fin
```

Algorithme 1: apprentissage d'une politique de contrôle

B.4 Évaluation et révision des choix

Une évaluation finale est nécessaire pour remettre en cause les choix effectués aux étapes précédentes, jusqu'à ce que le contrôle soit satisfaisant.

B.4.1 Critères d'évaluation

Des critères dévaluation sont nécessaires pour connaître les performances de la solution de contrôle, en terme de contrôlabilité et d'optimisation. Nous en retenons quatre, notés **π, ν, τ et γ**, que nous présentons ici. Le critère γ est estimé lors de la phase d'apprentissage. Les trois autres font l'objet d'un calcul, détaillé dans l'algorithme ci-dessous, qui comme l'apprentissage nécessite un nombre élevé de simulations.

```
Algorithme d'évaluation du contrôle
π ← 0, ν ← 0, τ ← 0
Répéter (nbSimulations)
    SMA.initialise()
    étatCourant ← SMA.identifieEtatCourant()
    // essayer d'atteindre la cible en moins de k cycles
    nbActions ← 0
    nbPasTotal ← 0
    tant que (nbActions<k et étatCourant ≠ cible)
        // choix d'une action
        action ← politique([Q(étatCourant, a1), Q(étatCourant, a2), ..., Q(étatCourant, aN)])
        SMA.applique(action)
        nbActions++
        nbPas ← 0
        // laisser le comportement du SMA se stabiliser
        Répéter
            SMA.pasDeSimulation()
            nbPas++
        jusqu'à (SMA.étatCourantIdentifié()
                        OU critèreArrêt(nbPas)) // l'état courant est alors s0
        nbPasTotal += nbPas
    fin tant que
    si (étatCourant = cible)
        π = π + 1
        ν = ν + nbActions
        τ = τ + nbPasTotal
    fin si
fin répéter
ν /= π
τ /= π
π /= nbSimulations
```

Algorithme 2: estimation de trois critères d'évaluation des performances d'une solution à la problématique

B.4.1.1 Critère π

Le premier critère indique l'espérance qu'une simulation atteigne la cible quand la solution de contrôle évaluée est utilisée. Il est lié à la contrôlabilité du système. Il est estimé par la proportion de simulations qui atteignent effectivement la cible.

Or s'il est possible de montrer qu'une simulation converge vers la cible en attendant suffisamment longtemps, on ne peut pas montrer qu'elle ne converge pas. Nous utilisons le même critère d'arrêt que pour l'apprentissage, en limitant chaque simulation à k cycles de contrôle. Nous appelons π la proportion des simulations qui convergent vers la cible en moins de k cycles. Le critère π sous-estime l'espérance d'atteindre la cible pour la solution de contrôle évaluée, en l'approchant d'autant mieux que k est élevé.

B.4.1.2 Critères ν et τ

Pour évaluer l'efficacité de la solution de contrôle testée, nous utilisons deux critères complémentaires : le nombre moyen de cycles, donc d'actions de contrôle, nécessaires avant d'atteindre la cible T et que nous notons ν, et le nombre moyen de pas de simulation du SMA avant qu'il atteigne T, noté τ. Le premier a un sens au niveau global, par exemple si les

actions possèdent un coût, tandis que le second a un aspect temporel que ν ne présente pas si les transitions entre deux comportements n'ont pas de durées homogènes.

Comme les simulations sont artificiellement arrêtées lorsqu'elles sont trop longues, seules celles qui atteignent la cible en moins de k cycles sont prises en compte pour calculer ces critères. Il s'agit donc là encore d'approximations d'autant plus fiables que k est grand, mais qui cette fois-ci surestiment l'optimalité.

B.4.1.3 Critère γ

Le dernier critère estime la durée de l'apprentissage de la solution de contrôle en évaluant sa durée. Il est mesuré lors de l'apprentissage. Nous appelons γ le nombre de cycles de contrôle qui ont été nécessaires lors de cet apprentissage. Ce critère est lui aussi imprécis, car il est généralement difficile de savoir exactement quand arrêter un apprentissage : à partir d'un certain temps, la politique est optimisée, et apprendre plus n'améliore plus les critères de performance précédents.

Notons que pour conserver une durée limitée, il est intéressant de fixer le critère d'arrêt des simulations d'apprentissage à un temps court, donc d'avoir une valeur de k peu élevée. Il y a cependant un risque de sous-évaluer les trois critères précédents.

B.4.2 Révision des choix

Les critères d'évaluation proposés précédemment permet d'estimer si le contrôle est satisfaisant par rapport à des attentes de performances ou par rapport à d'autres solutions de contrôle. Dans le cas contraire, il faut réviser les choix effectués à chaque étape de la proposition. La figure 11 montre que pour des critères d'évaluation donnés sur un problème de contrôle donné, les choix effectués aux trois étapes de la proposition peuvent être améliorés en les remettant en cause successivement, en fonction des résultats du contrôle. Il n'y a pas de moyen de certifier a priori que les choix sont corrects, avant d'avoir essayé le contrôle qui en résulte.

La mesure du comportement peut être évaluée à part, de façon directe, en comparant ses résultats aux comportements observés du SMA, donc elle n'entre pas nécessairement dans la présente étape de révision a posteriori.

L'ensemble d'états S doit être choisi avec attention, car plus il est petit, moins l'apprentissage est complexe, mais plus le critère d'évaluation π risque d'être faible car les situations que le SMA rencontre sont alors moins bien différenciées et les actions possibles moins nuancées. Nous cherchons par conséquent un équilibre entre temps d'apprentissage et performances, ce qui fait que cet ensemble se trouve à la limite de l'amélioration des performances.

Les choix de l'étape d'apprentissage sont nombreux. Nous considérons pour une révision éventuelle ceux qui déterminent le type de politique, le nombre de simulations d'apprentissage et le nombre de cycles k maximum par simulation. Les autres, comme l'utilisation d'un modèle markovien ou de l'algorithme Sarsa, sont considérés comme faisant partie de la proposition, et ne seront pas discutés dans ce document.

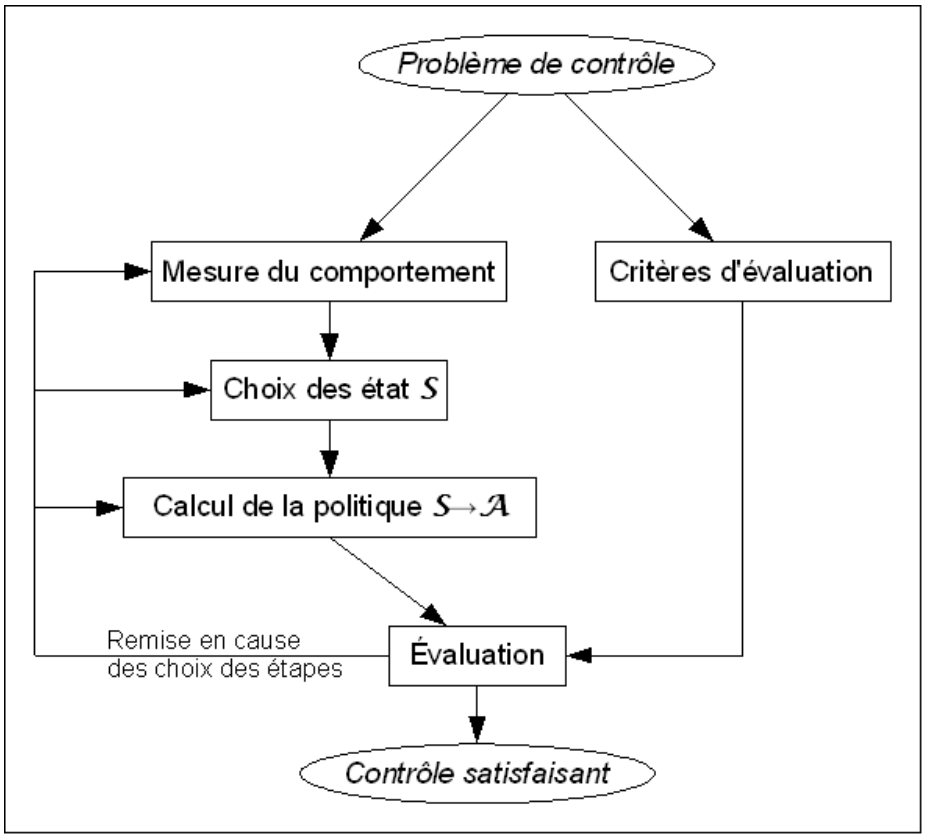

Figure 11: Révision, lors de l'évaluation, des choix effectués aux différentes étapes de la proposition

C. Conclusion

Nous proposons une solution de contrôle au niveau global d'un SMA, en modélisant sa dynamique par un MDP. Les états du MDP correspondent aux comportements globaux du système. Trois étapes préalables permettent de choisir la manière dont s'effectue la mesure du comportement, les états de contrôle, et la méthode d'apprentissage. Ces choix dépendent du SMA étudié et du problème de contrôle à résoudre. Un apprentissage de la dynamique est ensuite réalisé, afin d'optimiser automatiquement le contrôle en fonction de ces choix. La politique de contrôle ainsi obtenue est évaluée. Tant qu'elle n'est pas satisfaisante, les choix des étapes sont révisés par l'utilisateur humain, pour mieux répondre au problème de contrôle.

La proposition répond à la difficulté de lier le niveau local et le niveau global du SMA. Elle fournit des pistes pour diminuer la durée de l'apprentissage. Elle peut être présentée d'une façon différente, en reprenant les définitions du chapitre II.A.2 : il s'agit de donner à des phénomènes émergents observés de l'extérieur la possibilité d'influer le SMA au niveau local.

La pertinence de cette proposition dépend d'un certain nombre de choix, sur la mesure du comportement, les états de contrôle, et la méthode d'apprentissage. Il faut donc la confronter à un cadre applicatif pour estimer son potentiel. Le chapitre suivant présente le cadre applicatif retenu. Le chapitre V voit la concrétisation de notre proposition dans ce cadre applicatif.

Chapitre IV - Mise en oeuvre de la proposition sur un SMA d'étude

Dans ce chapitre, nous présentons en détails un SMA qui modélise des piétons dans un couloir. Nous exposons des exemples de problèmes de contrôle qui se posent sur ce système. Nous expliquons la mise en oeuvre de chaque étape de la proposition en fournissant de premiers choix pour chacun. Nous proposons une méthode originale pour mesurer les comportements sur le SMA étudié.

Ce chapitre laisse une certaine liberté sur les problèmes de contrôle et sur la façon d'appliquer la proposition, au lieu de fixer un problème de contrôle et une façon de le résoudre.

A. Système étudié

La proposition est appliquée sur un système qui représente grossièrement des piétons sur un trottoir ou dans un couloir. Le déplacement des piétons est une source d'inspiration classique du domaine des SMA [Gaud 08]. Notre système est avant tout un exemple jouet, intéressant pour ses propriétés d'émergence et de multiplicité des comportements globaux. En effet, ce SMA est choisi car des comportements globaux variés émergent en dépit de la simplicité du comportement des agents, et aussi parce que de nombreux paramètres entrent en jeu dans le comportement global et pourront être utilisés comme moyens d'action.

Les agents se déplacent dans un couloir circulaire[1], mus par un comportement réactif défini comme une réponse à une somme de forces. Ce type de comportement, inspiré par les boïds [Reynolds 87], est appelé modèle de forces sociales, et il est couramment utilisé pour modéliser des piétons [Helbing & Molnar 95, Helbing 00, Lacroix 06]. Dans un tel système, des structures stables de piétons émergent : ils se regroupent en lignes qui avancent dans une même direction, ou ils se bloquent mutuellement dans certains cas. On peut considérer comme comportement global tout agencement entre les structures émergentes.

A.1 Modèle des piétons

La spécification du système que nous donnons ici est inspirée de celle que l'on trouve dans [Helbing & Molnar 95], avec quelques simplifications et ajouts destinés à permettre l'obtention de comportements globaux particuliers, au détriment de la capacité du SMA à représenter des piétons de façon réaliste.

A.1.1 Spécifications des agents

L'environnement est un couloir cylindrique défini par sa longueur et sa largeur. Il possède une direction principale, celle de sa longueur, selon laquelle les agents vont essayer de se déplacer (voir figure 12).

Nous considérons que les agents sont tous identiques, de forme carrée, dont l'orientation est toujours la même (leurs bords sont parallèles aux bords de l'environnement), et de même taille. Ils peuvent prendre n'importe quelle position dans l'environnement à condition de ne pas se superposer et de ne pas chevaucher les bords.

Les agents ont en plus un objectif qui les attire et qu'ils cherchent à atteindre. Nous ne considérons que deux objectifs possibles, qui poussent les agents à se déplacer parallèlement aux longueurs du couloir, soit dans un sens soit dans l'autre. Cet objectif est propre à chaque agent et invariable. Nous le représentons par un vecteur $\vec{u}$.

Les agents possèdent également un champ de perception limité, que nous considérons formé par un demi-disque centré en leur centre, de même rayon pour tous, et orienté vers leur objectif.

Le temps du SMA est discret, et les agent se déplacent à chaque pas de temps. Ils

1 Plus précisément cylindrique : il peut être vu comme formé d'un rectangle, dont les longueurs sont infranchissables, et les largeurs sont confondues.

possèdent une vitesse instantanée, modifiée à chaque pas de temps par la résultante des forces qui s'appliquent sur l'agent, en fonction de sa masse inertielle. La vitesse est limitée en norme par une vitesse maximale. Nous supposons dans notre application que chaque agent a une masse unitaire, et que tous ont une même vitesse maximale.

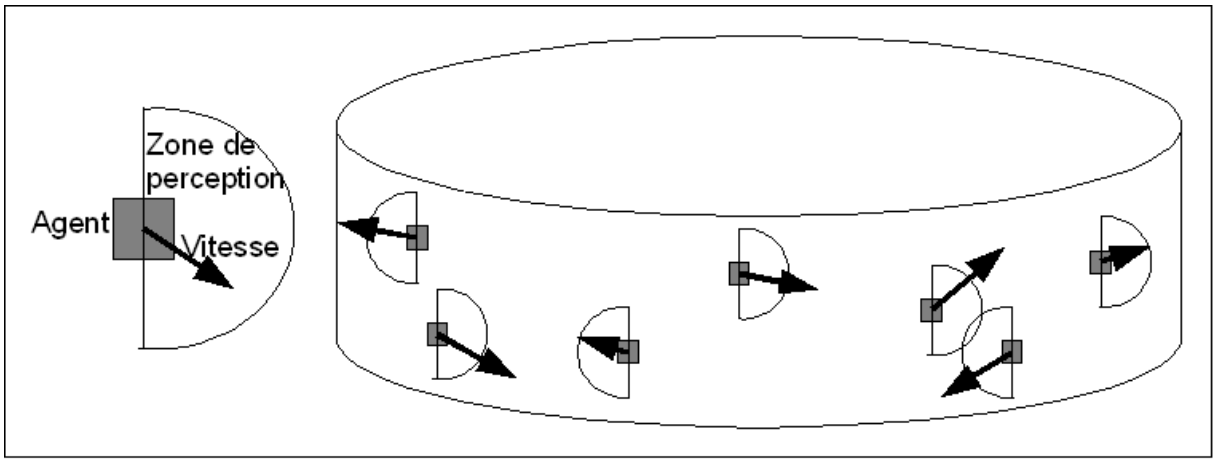

Figure 12: Évolution des agents dans un environnement cylindrique

A.1.2 Dynamique du modèle

Une simulation du système consiste à répéter un cycle de mise à jour de la position et de la vitesse de tous les agents. A chaque cycle, chaque agent est appelé une fois et une seule, toujours dans le même ordre, pour effectuer sa mise à jour. En fonction de ce qui se trouve dans sa zone de perception, l'agent A (pour les calculs, on appelle également A son centre) peut voir les bords de l'environnement et d'autres agents. Il calcule les forces qui s'exercent sur lui, et qui sont induites soit par son objectif, soit par sa perception.

A.1.2.1 Forces considérées

Nous retenons quatre forces qui s'exercent sur l'agent A. Les deux premières ne concernent que l'agent et son environnement, tandis que les deux suivantes sont calculées à partir de chaque autre agent B (de centre B) que perçoit A. Le calcul de chaque force tient compte d'un facteur multiplicatif qui détermine son influence.

La force de mouvement

Elle est induite par l'objectif $\vec{u}$ de A, et représente sa propension à avancer. La norme du vecteur $\vec{u}$ est constante et unitaire.

$\vec{f}_M = \mathrm{F_M} . \vec{u}$ où $\mathrm{F_M}$ est un coefficient appelé facteur de mouvement.

La force de bord

Elle représente la répulsion qu'exercent les bords du trottoir sur A. Elle est inversement proportionnelle à la distance qui sépare A du point du bord le plus proche de A qui appartient à sa zone de perception.

Soit T ce point du bord du trottoir le plus proche perceptible par A, c'est-à-dire le point de l'intersection entre la zone de perception de A et le bord qui minimise la distance AT :

$$T = \underset{X}{argmin}\{AX / X \in Bord \cap Zone\,de\,perception\}$$

On a alors $\vec{f}_B = \mathrm{F_B}\frac{\vec{TA}}{|\vec{TA}|^2}$ où $\mathrm{F_B}$ est un coefficient appelé facteur de bord.

La force de séparation

Elle représente le fait que A cherche à garder ses distances par rapport à B. Elle est inversement proportionnelle à la distance qui sépare A de B, et colinéaire à $\vec{AB}$, mais a le sens de $\vec{BA}$ puisqu'il s'agit d'une répulsion.

$\vec{f}_S = \mathrm{F_S}\frac{\vec{BA}}{|\vec{BA}|^2}$ où $\mathrm{F_S}$ est le facteur de séparation.

La force d'évitement

Elle représente le fait que A cherche à éviter une collision avec B. Elle est inversement proportionnelle à la distance qui sépare A de B, et proportionnelle à la projection de la vitesse relative de A et B ($\vec{v}_B - \vec{v}_A$) sur l'axe AB. Elle est colinéaire à $\vec{AB}$ (voir figure 13).

$\vec{f}_E = \mathrm{F_E}\frac{\vec{BA} \times (\vec{v}_B - \vec{v}_A)}{|BA|^3} * \vec{BA}$ où $\mathrm{F_E}$ est le facteur d'évitement.

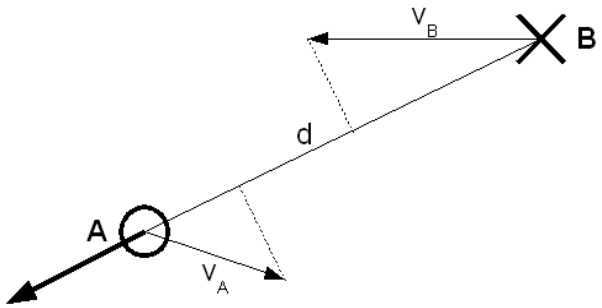

Figure 13: Représentation du calcul de la force d'évitement

A.1.2.2 Pas de simulation du modèle

On appelle f le vecteur qui résulte de la somme de ces forces $\vec{f} = \vec{f}_M + \vec{f}_B + \vec{f}_S + \vec{f}_E$. On peut écrire que

$\vec{f} = m * \vec{a}$ *et* $\vec{a} = \dot{\vec{v}}$ *ou, en temps discret,* $\vec{a}(t) = \vec{v}(t) - \vec{v}(t-1)$.

On en déduit que $\vec{v}(t)=\vec{v}(t-1)+\frac{\vec{f}}{m}$, et l'agent met ainsi à jour son vecteur vitesse.

Il limite alors sa vitesse : $si \ \ |\vec{v}(t)|>vitesseMax$, $\vec{v}(t)=\frac{\vec{v}(t)}{|\vec{v}(t)|}*vitesseMax$.

L'agent propose enfin à l'environnement sa nouvelle position $p(t)=p(t-1)+\vec{v}(t)$.

L'environnement vérifie s'il y a superposition entre la position proposée par A et un autre élément de l'environnement (autre agent, obstacle, ou extérieur du trottoir). Dans ce cas, il force A à reprendre sa position initiale : $p(t)=p(t-1) \ \ et \ \ \vec{v}(t)=\vec{0}$. Sinon, sa position et sa vitesse sont mises à jour.

La dynamique est résumée dans son ensemble sur la figure 14.

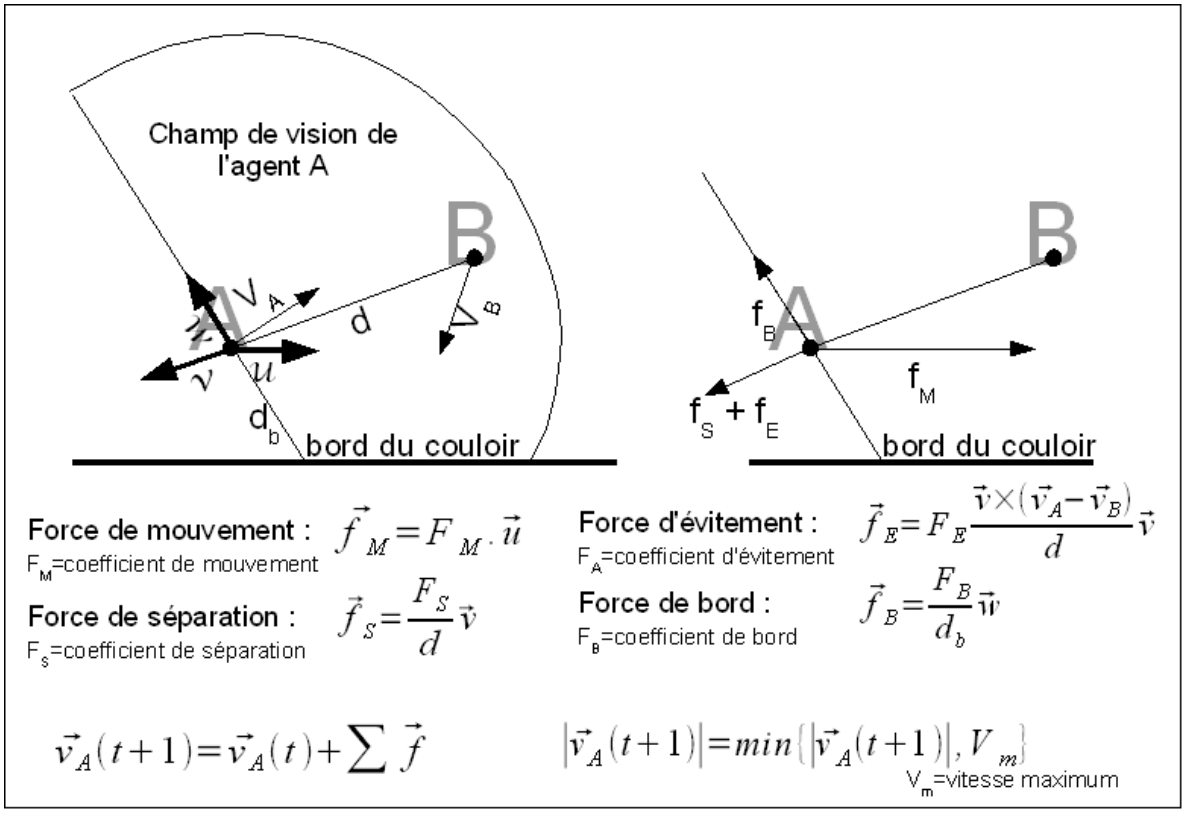

Figure 14: Calcul des forces qui s'exercent sur un piéton dans le modèle

A.1.3 Simulation du modèle

De manière pragmatique, si l'on souhaite effectuer une simulation de ce modèle, il reste à décider comment le système est initialisé, et quelles valeurs on donne aux différents paramètres.

L'environnement possède un « méridien » particulier, en partie fictif (il sert à représenter

l'environnement à plat), mais à partir duquel les agents sont introduits dans le système à intervalle de temps régulier. Initialement, il n'y a pas d'agent présent. Tous les cinq pas de simulation, un agent est ajouté sur ce méridien, à une position uniformément aléatoire sur la largeur du couloir, jusqu'à ce qu'il y en ait le nombre que l'on souhaite. L'ensemble des positions où les agents sont introduits forme les *conditions initiales* d'une simulation.

Le système possède plusieurs paramètres. Il s'agit de la longueur et de la largeur de l'environnement, du nombre d'agents – en réalité, 2 nombres sont nécessaires, puisque les agents ont deux objectifs possibles, donc sont répartis en deux populations – de leur taille, leur vitesse maximale, leur rayon de perception, et des quatre facteurs de force.

Avec des valeurs définies pour les 11 paramètres et une initialisation des agents donnée, le modèle peut être simulé en suivant les règles de calcul des forces et de mise à jour.

Dorénavant, si les valeurs de paramètres ne sont pas données explicitement pour une expérience, on se référera au tableau suivant qui fournit leurs valeurs standard.

Paramètre	***Valeur***
Largeur du couloir	200
Longueur du couloir	400
Nombre d'agents allant vers la gauche	15
Nombre d'agents allant vers la droite	15
Taille des agents (côté du carré)	12
Rayon du champs de perception	100
Vitesse maximale	3
Facteur de mouvement	4
Facteur de bord	20
Facteur de séparation	25
Facteur d'évitement	15

Tableau 1: Valeurs standard des paramètres du modèle

A.2 Spécificités et évolution du système à contrôler

Le modèle des piétons est implémenté sur machine, formant le SMA que nous étudions. Il est spécifié par des états dynamiques et des règles d'évolution, de façon similaire à un système dynamique, et par des comportements globaux. Nous développons ces aspects ci-dessous.

Dans le cadre du contrôle, nous avons posé l'hypothèse qu'il était toujours possible de mesurer le comportement du système. Pour y parvenir, nous supposons que toutes les informations sur l'état du SMA sont disponibles : à tout instant t, l'observation Y_t du SMA est égale à son état dynamique X_t.

A.2.1 États et règles d'évolution

L'état de l'environnement est entièrement défini par sa structure, c'est-à-dire sa longueur et sa largeur. L'état d'un agent est défini par sa position et son vecteur vitesse à tout instant. Finalement, l'état $X \in \mathcal{X}$ du SMA considéré comme un système dynamique est donné par la longueur et la largeur de l'environnement, et par la position et la vitesse de chaque agent.

Les règles d'évolution du SMA, qui correspondent aux règles d'évolution $\mathcal{F}$ d'un système dynamique, sont données par les quatre forces définies plus haut, et par les règles de déplacement des agents (mise à jour et limitation de la vitesse, pas de superposition et respect des limites de l'environnement). Il n'y a pas de dynamique propre à l'environnement : en l'absence d'agents, le SMA n'évolue pas (voir table 2). L'évolution du système est déterministe, la seule différence entre deux simulations avec les mêmes valeurs de paramètres vient des conditions initiales.

	Agents	**Environnement**
Règles d'évolution	Calcul des forces sur les agents Mise à jour/déplacement	$\emptyset$
État	Position Vitesse	Structure : longueur et largeur

Tableau 2: État et évolution du système des piétons

A.2.2 Comportement global

En fonction des valeurs de paramètres et de l'initialisation du système, différents phénomènes émergents peuvent survenir au niveau global. Dans [Helbing & Molnar 95], on voit sur un système similaire que des lignes d'agents peuvent apparaître, et dans [Helbing 00], une petite modification du même système – l'ajout de règles aléatoires – conduit à l'apparition de blocs d'agents.

Les deux mêmes types de structures émergent dans le SMA étudié. Les blocs d'agents sont plus fréquents que dans les travaux cités, en partie à cause de la forme carrée des agents. La figure 15 montre les deux types de structures. Elles apparaissent stables, c'est-à-dire qu'une fois qu'elles sont apparentes, il est *rare* qu'elles disparaissent spontanément au cours de l'évolution du SMA. Il arrive exceptionnellement qu'un bloc se crée suite à la rencontre de plusieurs agents aux directions opposées, ou qu'un bloc disparaisse sous l'influence des agents qui passent à proximité.

Un point de vue est choisi pour définir le comportement global du SMA : on s'intéresse au nombre de blocs et de lignes présents dans le système, mais ni à leur taille (nombre d'agents impliqués), ni à leurs positions relatives ou absolues. Un comportement est donc donné par un couple de valeurs :

- un nombre de blocs
- et un nombre de lignes.

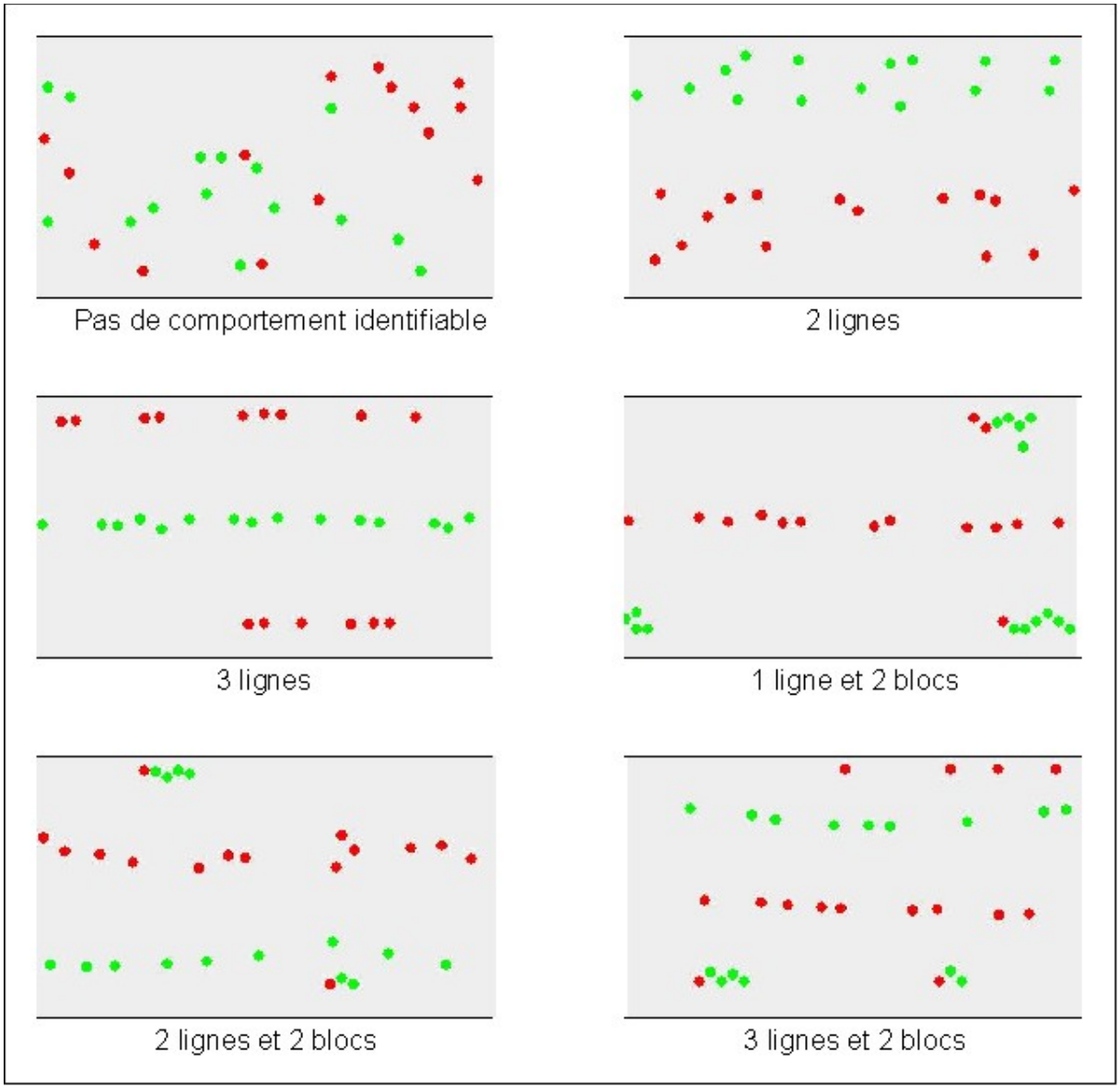

Figure 15: Différents comportements observables dans le système des piétons. Les agents verts (ou clairs en noir et blanc) vont vers la gauche et les rouges (foncés) vers la droite.

B. Problème de contrôle pour ce SMA

Comme nous l'avons vu dans le chapitre III, un problème de contrôle est défini par sa cible, les moyens d'action à disposition du contrôleur, et ses moyens d'observation. Nous déclinons ces trois données pour notre système d'étude.

La cible est composée d'un ou plusieurs comportements globaux, qui sont définis comme un nombre de blocs et un nombre de lignes dans le SMA. Nous choisirons généralement comme cible un unique comportement que doit présenter le SMA, mais il pourra également s'agir d'un ensemble de comportements globaux, comme par exemple l'absence de blocs dans le système.

Pour définir des problèmes de contrôle, nous sélectionnons des moyens d'action potentiels, regroupés en six ensembles. Chaque ensemble correspond à un paramètre du SMA. Les ensembles considérés sont :

1. Une modification de la largeur de l'environnement, en lui faisant prendre une valeur donnée. Ces valeurs sont réparties uniformément dans l'intervalle [30, 200], avec des intervalles de 5, on a donc 35 moyens d'action différents dans cet ensemble.

   Lorsque la largeur diminue, on autorise à chaque pas de temps le bord inférieur à remonter jusqu'à ce qu'il touche un agent. Lorsque la largeur augmente, ce bord descend en limitant la variation de la largeur à un maximum de 40 à chaque pas de temps.

2. Le second ensemble consiste à fixer la valeur du facteur de mouvement, en le choisissant dans l'ensemble {1, 3, 5, 7, 9}. La valeur est fixée instantanément, et de manière identique et simultanée pour tous les agents du système.

3. Le troisième ensemble regroupe des moyens d'action qui fixent le facteur de séparation de tous les agents. Il y a 5 moyens d'action dans cet ensemble : les valeurs possibles appartiennent à l'ensemble {4, 7, 10, 13, 16}.

4. De même, le quatrième ensemble fixe la vitesse maximale de tous les agents, à une valeur comprise dans {1, 2, 3, 4, 5}.

5. Le facteur d'évitement de tous les agents est modifié par les trois moyens d'action d'un sixième ensemble, qui fixent la valeur de ce paramètre à 10, 15 ou 20.

6. Le dernier ensemble de moyens d'action est similaire au second, le facteur de mouvement est modifié, mais cette fois ses valeurs sont comprises dans {2, 3, 4, 5}.

Pour un problème de contrôle donné, seuls certains de ces ensembles forment les moyens d'action : si plusieurs ensembles sont considérés, un moyen d'action est le choix combiné d'une valeur dans chaque ensemble. Par exemple, si les ensembles 2 et 3 sont retenus, effectuer une action de contrôle revient à modifier simultanément les facteurs de mouvement et de séparation. Il y a alors $5 \times 5 = 25$ moyens d'action différents dans $\mathcal{A}$.

Nous étudierons en plus un type de moyens d'action conceptuellement différent : l'ajout d'agents particuliers dans le SMA, que nous appellerons des leurres. Ces moyens d'action demandent des explications que nous fournirons lorsque nous réaliserons l'expérience correspondante.

Le choix de ces ensembles présente l'avantage d'être varié, en représentant des influences tant sur l'environnement que sur les interactions ou les comportements individuels, même si nous n'étudions pas les 11 paramètres du SMA. Nous avons considéré que les actions revenaient dans la plupart des cas à une modification des paramètres. Nous pouvons ainsi appliquer une méthode de calibration sur les mêmes problèmes de contrôle, afin de confronter une méthode statique à notre approche dynamique. Mais nous verrons que l'utilisation de leurres est un tout autre type d'actions, également utile pour contrôler un SMA.

Les moyens d'observation du SMA n'ont pas besoin d'être définis ou restreints, puisque nous supposons que l'état dynamique du SMA est entièrement observable. Il est donc possible de connaître l'état dynamique du système à chaque instant.

C. Application de la proposition

Afin d'illustrer comment appliquer la proposition, nous en développons chaque étape pour le système des piétons, en supposant qu'un problème de contrôle est fourni. Pour mémoire, les étapes de la proposition sont la mesure du comportement global, le choix des états de contrôle, et l'apprentissage de la politique de contrôle. Nous déclinons également la phase d'évaluation du contrôle et de révision des choix effectuées lors de ces étapes.

C.1 Caractérisation et mesure du comportement global

Nous présentons ici la mesure de comportement que nous avons utilisée pour déterminer automatiquement le nombre de lignes et de blocs d'agents présentés par le système des piétons. Elle répond à trois problèmes distincts :

- identifier des groupes d'agents
- déterminer s'il s'agit de lignes ou de blocs
- vérifier leur stabilité.

Seul le premier de ces trois problèmes pose une réelle difficulté, les deux autres sont plus faciles à résoudre. Nous portons donc une attention particulière à l'identification des groupes d'agents.

C.1.1 Le clustering classique et ses limites

L'identification de groupes d'agents, définis par leur position spatiale et leur vitesse, est un problème de clustering. Il existe de nombreux algorithmes de clustering, qui donnent une répartition d'entités en clusters, étant donné un nombre de clusters à trouver et une distance définie entre ces entités [Jain 99]. Dans le cas de notre système, cette distance pourrait être une combinaison entre, d'une part, la distance physique entre les positions de deux agents, et d'autre part la différence de vitesse entre ces deux agents. Mais dans notre cas, une difficulté supplémentaire existe : nous ne connaissons pas a priori le nombre de clusters.

Une solution possible à ce problème se trouve dans [Handl 04]. Il s'agit de chercher les clusters à l'aide d'une méthode de clustering classique, avec pour argument du nombre de clusters tous les nombres de 1 au nombre d'entités présentes. Un algorithme de clustering hiérarchique est particulièrement adapté à ce calcul. On obtient donc une collection de clusterings, parmi lesquels il faut choisir celui qui semble le meilleur et qui correspond donc au nombre de clusters optimal. Pour cela, on compare pour chaque clustering une mesure de dispersion à l'intérieur des clusters et entre les clusters. La dispersion est calculée à partir de la même distance que celle utilisée entre les agents pour créer les clusters.

Cette solution présente un inconvénient pour l'étude du SMA : si certains paramètres du système changent, il peut être nécessaire de redéfinir la distance entre les agents. Sans cela, la méthode de clustering risque de fournir des résultats erronés. Nous préférons nous affranchir de la définition d'une distance sur le SMA.

C.1.2 Solution utilisée

Nous proposons notre propre méthode de clustering, en partie distribuée, et que nous pourrions par la suite adapter à une éventuelle décentralisation du SMA. Le principe est de demander à chaque agent quels autres agents font partie du même cluster que le sien, d'après des observations simples qu'il peut effectuer. Ceci définit une relation $\mathcal{R}$ entre les agents : pour des agents a_1 et a_2, $a_1 \mathcal{R} a_2$ si et seulement si a_1 considère a_2 comme partageant le même cluster. Puis de façon centralisée, on peut construire un graphe $\mathcal{G}$ dont chaque sommet représente un agent, et les arêtes correspondent à la relation $\mathcal{R}$. En considérant que $\mathcal{G}$ est non-orienté (ou que $\mathcal{R}$ est symétrique), les clusters du système sont les composantes connexes de $\mathcal{G}$. Ainsi, les clusters sont construits de proche en proche, en passant d'un agent à un autre (voir figure 16).

Il reste à préciser ce qu'est la relation $\mathcal{R}$. La stabilité des clusters doit déjà apparaître à ce niveau, puisqu'un agent ne peut considérer qu'un autre appartient au même cluster que si ce dernier se trouve *souvent* dans son champ de perception. Chaque agent a_1 retient donc la fréquence f à laquelle il perçoit chaque autre agent a_2, et $a_1 \mathcal{R} a_2$ si et seulement si f>80%.

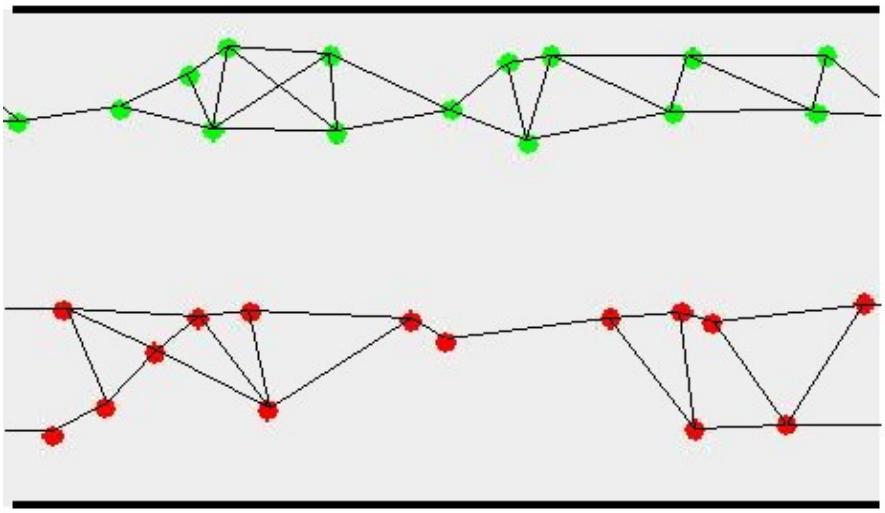

Figure 16: Exemple de graphe représentant des clusters d'agents

Une fois le clustering effectué, il reste à identifier le type de groupe pour chaque cluster. Un bloc est caractérisé par la présence d'agents essayant d'aller dans des directions opposées et par une vitesse moyenne nulle ou très faible des agents, tandis qu'une ligne est composée d'agents avançant dans le même sens. Ces deux critères sont testés, et une ligne est identifiée si et seulement si la vitesse moyenne des agents qui la composent est supérieure au quart de leur vitesse maximale et tous ces agents ont le même objectif.

Pour vérifier la stabilité du clustering, on compare le nombre de clusters trouvés pendant 100 pas de simulation, qui est de l'ordre de grandeur observé du nombre de pas nécessaires pour passer d'un comportement à un autre. Si le nombre de clusters ne varie pas une seule fois sur 100 estimations successives, on considère que le comportement est stable et que le clustering calculé correspond bien au comportement du SMA. Si ce critère de stabilité n'est pas atteint au bout de 5000 pas de simulation, on considère que le comportement est

intrinsèquement instable, donc non reconnaissable. La mesure indique alors que le SMA se trouve dans l'état S_0, défini dans le chapitre précédent comme un état particulier où aucun état attendu n'est reconnu.

Il y a deux limites à cette proposition :

- D'abord, le choix du seuil de la fréquence de perception à 80% est empirique et correspond à un champ de perception donné, mais la mesure devra être recalibrée si celui-ci est modifié. En effet, un agent perçoit les autres agents d'autant plus fréquemment que son champ de perception est vaste.
- Ensuite, la mémoire nécessaire pour retenir les fréquences de rencontre augmente avec le carré du nombre d'agents, ce qui peut poser problème dans des SMA de grande taille[1]. Dans le cadre du SMA étudié, cette limite ne cause pas de soucis.

C.2 Choix des états de contrôle

Nous avons défini le comportement du système des piétons comme le décompte du nombre de blocs et de lignes d'agents, mais la connaissance exacte des deux nombres n'est pas forcément utile. Choisir un ensemble $\mathcal{S}$ des états du MDP revient à décider comment regrouper ces différents comportements en fonction de l'influence qu'ils ont sur l'évolution du SMA et sa capacité à atteindre la cible.

Par exemple, si nous supposons que la cible est d'avoir un nombre de lignes compris entre 2 et 4 et un nombre de blocs quelconque, une liste non exhaustive des ensembles d'états peut être :

- le nombre exact de lignes du comportement courant
- l'indication que le nombre de lignes est supérieur à 4, inférieur à 2, ou compris dans la cible
- l'indication booléenne que ce nombre est compris dans la cible ou non
- toute autre indication, comme le fait que ce nombre est nul ou strictement positif
- aucune information sur les lignes (seulement sur les blocs par exemple)

L'ensemble S des états de contrôle doit être choisi en fonction de sa capacité à discriminer l'influence des différents moyens d'action.

C.3 Apprentissage

L'apprentissage se fait en effectuant des simulations sur le SMA. Chaque simulation est limitée à k=50 cycles de contrôle. Ce choix est empirique, mais nous pourrons montrer expérimentalement que les simulations qui atteignent la cible y parviennent la plupart du temps bien avant ces 50 cycles.

En raison de la durée importante des simulations, nous décidons d'en limiter le nombre à quelques milliers dans les expériences. Ce choix est lui aussi empirique, et nous vérifierons de manière expérimentale que ce nombre est suffisamment élevé pour permettre une amélioration réelle de la politique.

---

1 Dans ce cas, on peut tout de même supposer que tous les agents ne se rencontrent pas et utiliser des astuces algorithmiques en conséquence, comme un codage par matrice creuse. La proposition peut donc être réutilisée aussi dans des systèmes de ce type.

Par défaut, et sauf indication contraire, le type de politique utilisée est un simple choix d'actions aléatoire, avec des probabilités proportionnelles aux Q(s,a) pour chaque état s. Ce choix n'est sans doute pas optimal pour le contrôle, comme nous pourrons le vérifier expérimentalement, mais il présente l'avantage de ne pas utiliser de paramètres annexes, comme le ε d'une politique ε-gloutonne ou le paramètre T d'une politique softmax.

C.4 Évaluation et exploitation

Lors du contrôle proprement dit, une politique est utilisée pour déterminer les actions à effectuer en fonction des états rencontrés. Cette politique de contrôle est fondée sur les valeurs des Q(s,a) calculées dans l'étape précédente. Nous choisissons d'utiliser le même type de politique que celui de la phase d'apprentissage.

Pour évaluer le contrôle, comme pour l'apprentissage, un nombre élevé de simulations est nécessaire. Plus il est élevé, plus l'évaluation des critères est fiable. Dans les expériences que nous réaliserons, le critère le plus pertinent sera la proportion de convergence π, qui définit la contrôlabilité, et que nous voudrons connaître avec une précision de quelques pour cents. Par conséquent, nous choisissons de toujours estimer le contrôle après 500 simulations dans ces expériences, ce qui fournit la précision voulue[1]. Comme pour l'apprentissage, chaque simulation est limitée à k=50 cycles de contrôle.

1 Un intervalle de confiance à 95% pour la valeur de π possède un rayon donné par la formule $1{,}96\sqrt{\frac{\pi(1-\pi)}{500}}$ pour 500 simulations. Par exemple, si π est estimé à 90%, alors sa valeur réelle a 95% de chances d'appartenir à l'intervalle $[0{,}9-1{,}96\sqrt{\frac{0{,}9(1-0{,}9)}{500}};\ 0{,}9+1{,}96\sqrt{\frac{0{,}9(1-0{,}9)}{500}}]$ soit [87,4% ; 92,6%]. Le rayon de l'intervalle vaut 2,6%, donc 500 simulations sont satisfaisantes.

D. Bilan

Nous avons présenté un SMA réactif à la fois simple dans sa spécification, et riche dans les comportements globaux qu'il présente. Nous définissons ces comportements comme une description des blocs et des lignes d'agents qui émergent.

Nous avons également défini plusieurs moyens d'action variés sur les paramètres du système. Ils seront utilisés pour caractériser des problèmes de contrôle à résoudre dans le chapitre suivant.

L'ensemble forme un cadre applicatif auquel nous allons confronter notre proposition de contrôle. Nous avons présenté la manière dont elle est mise en oeuvre sur le système d'étude. Nous proposons une mesure du comportement dans ce SMA qui s'appuie sur une méthode de clustering originale. Nous avons aussi effectué des choix relatifs aux autres étapes de la proposition. Ces premiers choix seront évalués expérimentalement et discutés dans le chapitre suivant.

Il reste une grande liberté sur les problèmes de contrôle possibles et la manière d'appliquer la proposition, afin d'étudier les avantages et les limites de la proposition, et de valider le principe de contrôle au niveau global du SMA.

Chapitre V - Étude expérimentale de la proposition

Les expériences menées dans ce chapitre ont pour objectif d'évaluer les performances du contrôle d'un SMA au niveau global, en particulier avec les choix de la proposition, c'est-à-dire en considérant les comportements globaux du système comme les états d'un MDP. Pour cela, nous allons

- étudier la proposition en termes de contrôlabilité, d'optimalité temporelle de contrôle, et de temps d'apprentissage,
- montrer l'influence des choix à chaque étape de la proposition sur les performances du contrôle,
- valider l'approche de contrôle proposée en la comparant à d'autres approches.

Un second objectif est d'explorer les capacités de la proposition et d'en vérifier les fondements. Pour y parvenir, nous étudierons la robustesse d'une politique de contrôle lorsque les conditions de simulation sont différentes entre l'apprentissage et le contrôle.

Toutes ces expériences ont pour objet d'étude le système multi-agents représentant le déplacement de piétons, défini au chapitre précédent.

A. Explications préliminaires

Dans ce chapitre, nous validons à la fois l'approche de contrôle que nous avons proposée et le principe même du contrôle d'un SMA au niveau global. Nous effectuons cette validation en étudiant la proposition à la fois de l'intérieur – ses performances, sa mise en oeuvre – et de l'extérieur, en la comparant à d'autres approches. Nous étudions également les régularités du SMA au niveau global, sur lesquelles est fondée la proposition. L'approche est appliquée sur le système des piétons, avec les premiers choix évoqués au chapitre précédent.

Nous expliquons la démarche expérimentale suivie. Pour mémoire, nous rappelons aussi quels sont les critères d'évaluation, et ce qu'est un problème de contrôle, en présentant les principaux problèmes étudiés.

A.1 Démarche expérimentale

Ce chapitre regroupe un ensemble d'expériences réalisées pour répondre à nos divers objectifs. Une expérience est un canevas régulier, qui consiste à :

1. considérer un problème de contrôle sur le SMA,
2. appliquer la proposition ou une autre approche destinée à maîtriser le comportement global du SMA sur ce problème de contrôle,
3. estimer les critères d'évaluation des performances sur plusieurs simulations.

Entre deux simulations d'une même expérience, seules changent les conditions initiales, c'est-à-dire l'introduction des agents dans le SMA, et les modifications induites par le caractère éventuellement stochastique de l'approche utilisée.

Lorsque nous appliquons notre proposition dans une expérience, nous précisons l'ensemble S des états de contrôle choisis, le nombre de simulations d'apprentissage, et le type de politique apprise. Les autres choix effectués aux étapes de la proposition, qui concernent par exemple la mesure du comportement global et le nombre maximal de cycles de contrôle par simulation, sont définis par défaut au chapitre IV.

L'analyse des résultats d'une même expérience pourra être utilisée pour illustrer différents points de discussion. Les expériences apparaissent donc à l'écart du texte.

A.2 Critères d'évaluation des performances

Critère d'évaluation	π	ν	τ	γ
Description	taux de convergence vers la cible, sera exprimé en %	nombre moyen de cycles de contrôle avant d'atteindre la cible	nombre moyen de pas de simulation avant la cible	nombre de cycles de contrôle utilisés lors de l'apprentissage, exprimé en milliers

Tableau 3: Les quatre critères d'évaluation d'une approche

Pour rappel, quatre critères d'évaluation des performances ont été définis, ils sont résumés dans le tableau 3. Le taux de convergence π estime la contrôlabilité de l'approche utilisée, c'est-à-dire la capacité de cette approche à atteindre la cible. Les critères ν et τ évaluent l'optimisation temporelle du contrôle, en mesurant de deux façons différentes la rapidité d'atteindre la cible. Enfin, la durée de l'apprentissage est représentée par le critère γ. Les performances du contrôle sont évaluées sur 500 simulations.

A.3 Problèmes de contrôle sur un SMA

Pour le SMA étudié, nous avons défini un problème de contrôle comme la donnée d'une cible, d'un ensemble de moyens d'action $\mathcal{A}$, et de moyens d'observation sur le système. Nous avons vu au chapitre précédent que nous ne nous intéressions pas aux moyens d'observation, en considérant que l'ensemble du SMA est observable.

Les approches étudiées dans les expériences suivantes se résument à choisir les actions à appliquer afin d'atteindre la cible.

Problème	***Cible***	***Ensembles de moyens d'action***
p_1	0 bloc, 3 lignes	N°1: largeur de l'environnement
p_2	1 bloc, 2 lignes	N°2: facteur de mouvement ; N°3: facteur de séparation
p_3	0 bloc, 2 lignes	N°2 ; N°3 ; N°4: vitesse maximale
p_4	0 bloc, 3 lignes	Ajout de leurres[1]

Tableau 4: Résumé des principaux problèmes de contrôle étudiés

Quatre problèmes de contrôle différents seront fréquemment étudiés. Nous les notons p_1, p_2, p_3 et p_4. Ils sont synthétisés dans le tableau 4. Chacun propose une cible observée fréquemment sur le SMA des piétons, mais non triviale à atteindre avec les moyens d'action disponibles. De cette manière, les problèmes sont potentiellement suffisamment difficiles pour que des approches plus simples ne conviennent pas. Les moyens d'action sont plus ou moins nombreux en fonction des problèmes, et influent sur l'environnement ou sur les agents. Ainsi, les problèmes sont variés et permettent d'étudier le SMA sous plusieurs angles.

1 La définition de ce problème sera approfondie lorsque nous le traiterons.

B. Estimation de la contrôlabilité

Notre objectif ici est d'estimer la contrôlabilité du SMA sur plusieurs problèmes en utilisant notre proposition. Il s'agit également de vérifier qu'il est possible d'atteindre la cible grâce à notre approche, et de s'assurer que l'apprentissage peut être effectué en un temps raisonnable.

B.1 Expériences

Dans les expériences 1, 2 et 3 ci-dessous, nous appliquons la proposition sur les problèmes p_1, p_2 et p_3 décrits en introduction. Nous indiquons à chaque fois les choix effectués aux différentes étapes de la proposition. Ces choix seront discutés plus loin. Nous donnons pour chaque expérience un tableau des résultats obtenus.

Expérience 1: application de la proposition au problème p_1

Problème p_1

Mise en oeuvre

Application de la proposition avec les choix suivants :

États de contrôle $\mathcal{S}$: 6 états choisis empiriquement (nous les justifierons par la suite)

- S_0 (aucun état reconnu)
- état cible
- au moins un bloc et pas de ligne
- au moins un bloc et au moins une ligne
- pas de bloc et moins de 3 lignes
- pas de bloc et plus de trois lignes

Apprentissage de la politique en 4000 simulations, politique proportionnelle aux Q(s,a).

Résultats

π (en %)	ν	τ	γ (.10^3)
89	10	1500	57,5

Tableau 5: Résultats de contrôle pour l'expérience 1. La proportion de convergence π est exprimée en pourcentage, et le nombre de cycles d'apprentissage γ est donné en milliers.

Expérience 2: application de la proposition au problème p_2

Problème p_2

Mise en oeuvre

Application de la proposition avec les choix suivants :

États de contrôle $\mathcal{S}$: 18 états identifiés par un couple de valeurs qui indique

- le nombre de blocs. Valeurs possibles : 0, 1, et "2 ou plus".
- le nombre de lignes : 0, 1, 2, 3, 4, et "5 ou plus".

La cible est représentée par le couple (1,2), et l'état S_0 par (0,0)

Apprentissage de la politique en 3000 simulations, politique ε-gloutonne avec ε=40%.

Résultats

π (en %)	ν	τ	γ (.10^3)
94	8	1300	31,5

Tableau 6: Résultats de contrôle pour l'expérience 2.

Expérience 3: application de la proposition au problème p_3

Problème p_3

Mise en oeuvre

Application de la proposition avec des choix semblables à ceux de l'expérience précédente :

États de contrôle $\mathcal{S}$: 18 états identifiés par un couple de valeurs qui indique

- le nombre de blocs. Valeurs possibles : 0, 1, et 2 ou plus.
- le nombre de lignes : 0, 1, 2, 3, 4, et 5 ou plus.

Apprentissage de la politique en 3000 simulations, politique ε-gloutonne avec ε=10%.

Résultats

π (en %)	ν	τ	γ (.10^3)
67	11	1700	71,6

Tableau 7: Résultats de contrôle pour l'expérience 3.

B.2 Résultats

Les expériences montrent qu'il est possible d'effectuer l'apprentissage de la politique de contrôle en un temps limité. Dans les trois expériences, il aboutit en quelques heures de simulation dans des conditions de calcul peu optimisées[1].

Dans ces trois expériences, nous observons une valeur de contrôlabilité π élevée, mais différente de 100% : elle vaut respectivement 89%, 94% et 67%. Le contrôle est effectif, dans la mesure où la cible est régulièrement atteinte pour chaque problème de contrôle. Mais même avec un SMA et des problèmes aussi simples, l'obtention de la cible n'est pas assurée.

Critère	*Valeur pour p_1*	*Valeur pour p_2*	*Valeur pour p_3*
ν	10	8	11
τ	1500	1300	1700

Tableau 8: Comparaison des critères ν et τ dans les trois premières expériences

Les valeurs de ν et τ semblent corrélées (voir tableau 8), comme nous nous y attendions, puisqu'elles approximent la même notion. Cela indique que les cycles de contrôle ont des durées homogènes en terme de nombre de pas de simulation. Nous remarquons également que les valeurs de ces critères varient peu d'un problème à l'autre, comparativement au critère π.

B.3 Conclusions

L'apprentissage est assuré en temps limité dans les cas étudiés, mais avec une durée non négligeable. Dans nos conditions de calcul, une valeur de γ=50000 correspond à environ 3 heures d'apprentissage. Il reste à vérifier si cette durée d'apprentissage est nécessaire pour obtenir de bonnes performances de contrôle. L'approche proposée sera validée si elle permet d'obtenir de meilleurs résultats qu'avec des approches plus rapides.

La proposition permet d'obtenir des taux de convergence vers la cible satisfaisants, mais non d'assurer un comportement global en temps fini.

Les problèmes étudiés sont simples mais déjà intéressants pour le contrôle. Les valeurs de π trouvées indiquent que plusieurs comportements globaux sont possibles, ce qui correspond au cadre du contrôle défini au chapitre III. Les conditions initiales sont le seul aspect non maîtrisé de chaque simulation, qui implique ces différents comportements, et une simple modification de ces conditions initiales empêche de prévoir et de maîtriser le comportement, ce qui se traduit par une contrôlabilité limitée.

Le critère π est plus intéressant pour ces problèmes que les critère ν et τ. En effet, nous souhaitons assurer l'obtention de la cible avant d'essayer d'optimiser la manière dont on y

1 À titre indicatif, cet apprentissage a été réalisé en langage Java, sur un ordinateur portable sous Windows XP, et avec un SMA pas nécessairement optimisé pour la rapidité de ses simulations.

arrive. De plus, les valeurs de π varient davantage que celles de ν et τ, et semblent donc plus significatives pour la qualité du contrôle.

Nous pouvons finalement nous demander comment améliorer le taux de convergence π et les autres critères. Nous montrons ci-après qu'ils dépendent fortement des choix effectués à chacune des étapes de la proposition.

C. Influence des choix à chaque étape

Dans la proposition, l'optimisation des trois critères π, ν et τ est automatique, lors de l'apprentissage, mais cette optimisation dépend de choix préalables : les décisions effectuées lors des étapes de la proposition influent sur les performances du contrôle.

Notre objectif est de mettre en évidence l'influence des choix algorithmiques ou de modélisation sur un problème de contrôle. Plus précisément, nous étudions :

- le choix de l'ensemble d'états *S*,
- le choix du nombre de simulations d'apprentissage, afin de savoir s'il est suffisant ou s'il faut poursuivre cet apprentissage,
- l'influence du nombre de cycles maximal sur l'approximation du critère π,
- l'influence des familles de politiques, en les comparant pour trouver celle qui améliore la contrôlabilité.

Nous prenons comme fil conducteur le problème p_1, en essayant d'améliorer les choix effectués afin que la politique apprise offre de meilleures performances de contrôle.

C.1 Sélection de l'ensemble $\mathcal{S}$ des états

Le principe qui gouverne le choix des états de contrôle est de distinguer les situations à partir desquelles les actions ont des effets différents sur le comportement global, et en particulier sur la capacité du SMA à atteindre la cible.

La figure 17 montre une façon de comparer les états de l'expérience 1 en fonction des valeurs des Q(s,a) lorsque l'action a varie. Chaque diagramme représente les valeurs des Q(s,a) obtenues après l'apprentissage de l'expérience 1 en partant de l'un des six états étudiés. Le sixième état, qui représente la cible (pas de bloc et trois lignes d'agents), n'a pas de valeurs significatives car nous n'avons pas étudié l'influence des actions à partir de cet état.

Les deux premiers diagrammes sont presque identiques : nous remarquons que l'état S_0 et l'état où le SMA présente au moins un bloc et aucune ligne (appelons-le S_1) ont des influences similaires sur le comportement : $\forall a \in \mathcal{A},\ Q(S_0, a) \approx Q(S_1, a)$. Ces deux états peuvent être considérés comme identiques et confondus en un seul. Mais nous avons vu au chapitre III que la distinction de l'état S_0 est nécessaire pour que l'approche proposée fonctionne.

Lors de l'initialisation d'une simulation, la mesure du comportement ne peut pas être effectuée, par manque de temps pour observer sa stabilité. L'algorithme de contrôle considère donc systématiquement que le SMA est initialisé dans l'état S_0. Finalement, nous pouvons conclure de cette première observation que l'initialisation du SMA est très similaire, en termes de comportement global du système, à l'état où tous les agents sont bloqués.

Une telle similarité des diagrammes ne se présente pour aucune autre paire d'états. Cela signifie que les actions de contrôle ont des influences différentes en fonction des états choisis. La distinction de ces états est donc nécessaire pour choisir correctement une action de contrôle.

Sur le diagramme situé en bas à gauche, pour l'état où le SMA présente plus de trois lignes et aucun bloc, certaines valeurs des Q(s,a) sont très faibles. Il s'agit des valeurs initiales de l'algorithme, car les actions correspondantes n'ont pas été testées dans cet état, rencontré trop rarement. Ce diagramme est donc incomplet, mais nous observons néanmoins sa forme générale, avec un optimum qui se situe pour une largeur aux alentours de 170.

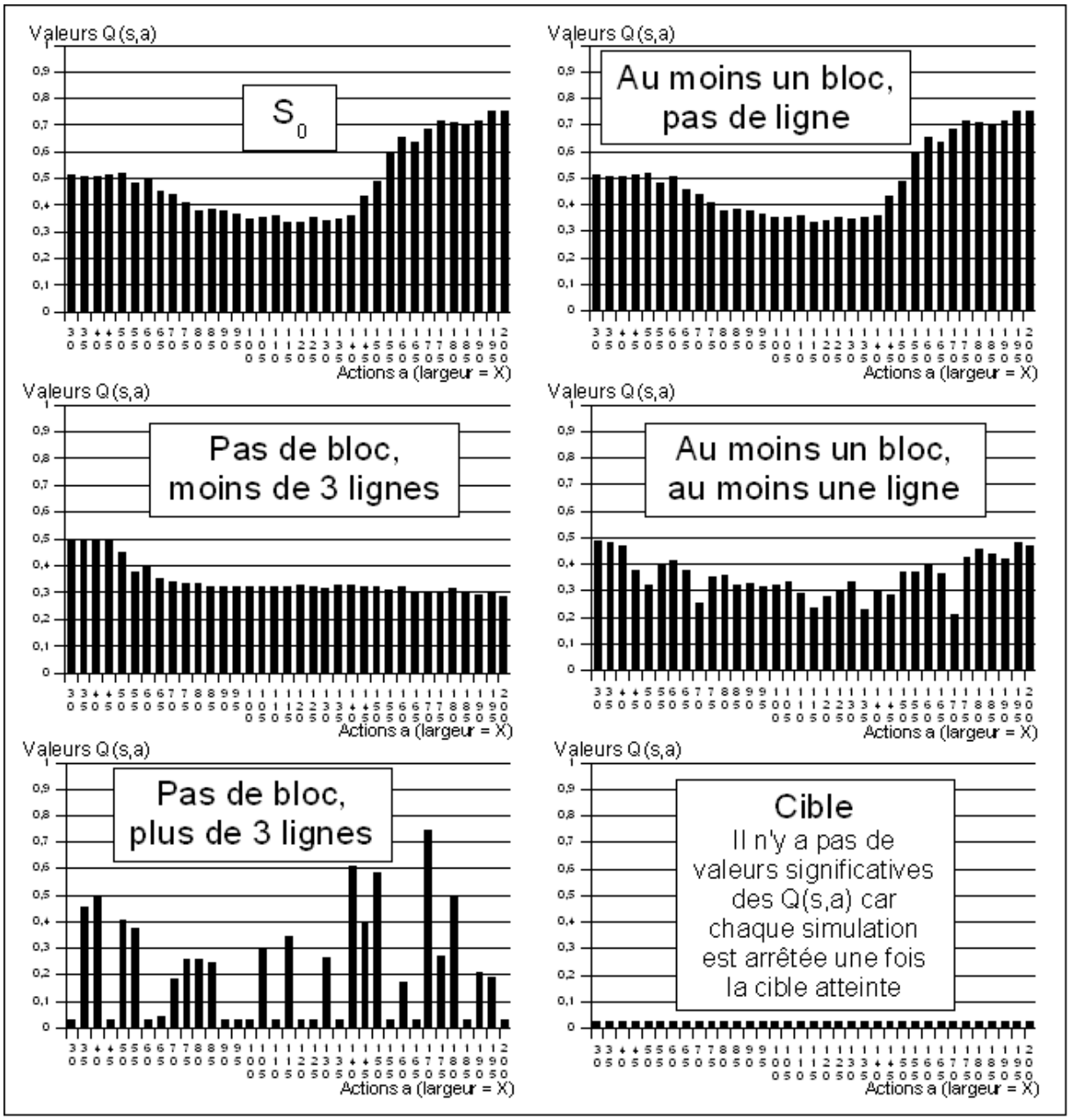

Figure 17: Comparaison des six états de l'expérience 1 en fonction des valeurs Q(s,a)

Le choix de ces états a été réalisé de façon heuristique, en observant les comportements globaux les plus fréquents et les transitions entre eux lorsque la largeur de l'environnement varie. Pour valider ce choix, les expériences 4 et 5 étudient respectivement l'influence d'un ensemble *S* plus grand et d'un ensemble *S* plus petit sur le contrôle.

Expérience 4: choix d'un ensemble d'états plus grand pour le problème p_1

Problème p_1

Mise en oeuvre

Comme pour l'expérience 1, la politique est proportionnelle aux Q(s,a).

L'ensemble $\mathcal{S}$ des états du MDP indique ici :

- le nombre de blocs, compris entre 0 et 4 (plus de 4 assimilé à 4)
- le nombre de lignes, compris entre 0 et 5, il y a donc 30 états

On effectue 6000 simulations d'apprentissage, au lieu de 4000 dans l'expérience 1, car plus d'états implique plus d'exploration nécessaire.

Résultats

Politique apprise avec	***π (en %)***	***ν***	***τ***	***γ (.10³)***
30 états	88	10	1500	89
6 états (expérience 1)	89	10	1500	57,5

Tableau 9: Résultats de contrôle pour un MDP avec 30 états, et comparaison avec ceux de l'expérience 1.

L'utilisation d'un ensemble d'états S plus grand, au sens de l'inclusion, n'améliore ni la contrôlabilité, ni l'optimisation du contrôle, et nécessite un apprentissage plus long. L'expérience 4 valide le choix initial des états lors de l'expérience 1 : la distinction d'états plus nombreux aboutit à des performances de contrôles quasi identiques.

L'utilisation d'un ensemble S plus petit dans l'expérience 5, en revanche, fait chuter la contrôlabilité pour le problème p_1. Le temps de convergence vers la cible est légèrement supérieur à celui de l'expérience 1. Le temps d'apprentissage est beaucoup plus élevé car les simulations sont plus longues en moyenne, en partie à cause de cette différence entre les valeurs de τ, mais surtout à cause du nombre plus élevé des simulations qui n'aboutissent pas à la cible et qui nécessitent 50 cycles de contrôle au lieu de υ=12.

Il manque ici de l'information sur l'évolution courante du SMA, puisque le seul renseignement disponible pour que le système de contrôle choisisse une action de contrôle est de savoir si le comportant est identifié ou non. Nous montrerons que cet ensemble trop petit d'états amène à des performances proches d'un contrôle aléatoire (cf. §D.1.2).

Expérience 5: choix d'un ensemble d'états plus petit

Problème p_1

Mise en oeuvre

L'ensemble S des états du MDP est limité à 3 états :

- S_0 (pas d'état reconnu)
- la cible
- tout autre état reconnu

et le nombre de simulations d'apprentissage repasse à 4000, comme dans l'expérience 1.

Résultats

Politique apprise avec	***π (en %)***	***ν***	***τ***	***γ (.10³)***
3 états	73	12	1600	89
6 états (expérience 1)	89	10	1500	57,5

Tableau 10: Résultats de contrôle pour un MDP avec 3 états, et comparaison avec ceux de l'expérience 1.

Le choix de l'ensemble S des états de contrôle influence les performances de la politique de contrôle apprise. Un ensemble S trop petit ou mal choisi ne permet pas de distinguer l'influence des moyens d'action et de sélectionner efficacement une action de contrôle. La politique de contrôle se rapproche alors d'une sélection aléatoire des actions. Un ensemble trop grand d'états n'augmente pas les performances, voire les diminue, tout en augmentant la durée d'apprentissage nécessaire puisque l'espace à explorer est plus vaste.

Il y a finalement un équilibre à établir lors du choix de l'ensemble S. Le choix de cet ensemble est dépendant du problème de contrôle traité. Pour la répartition des comportements globaux en états, sur le problème p_1, il semble exister un ensemble d'états minimum au sens de l'inclusion, équilibré entre l'obtention de bons résultats de contrôle, et un temps d'apprentissage limité.

C.2 Influence du nombre de simulations d'apprentissage

La durée de l'apprentissage dépend directement du nombre de simulations. Nous voulons arrêter cet apprentissage dès que les performances de contrôle n'augmentent plus (comme par exemple dans [Dutech 03]). C'est pourquoi nous étudions ici l'évolution du critère π au cours de l'apprentissage de l'expérience 1.

Pendant l'apprentissage, les performances sont estimées sur une fenêtre des 100 dernières

simulations. Cette estimation est effectuée régulièrement, toutes les 20 simulations. Ainsi, on peut suivre l'évolution de la contrôlabilité π qui est représentée sur le graphe de la figure 18.

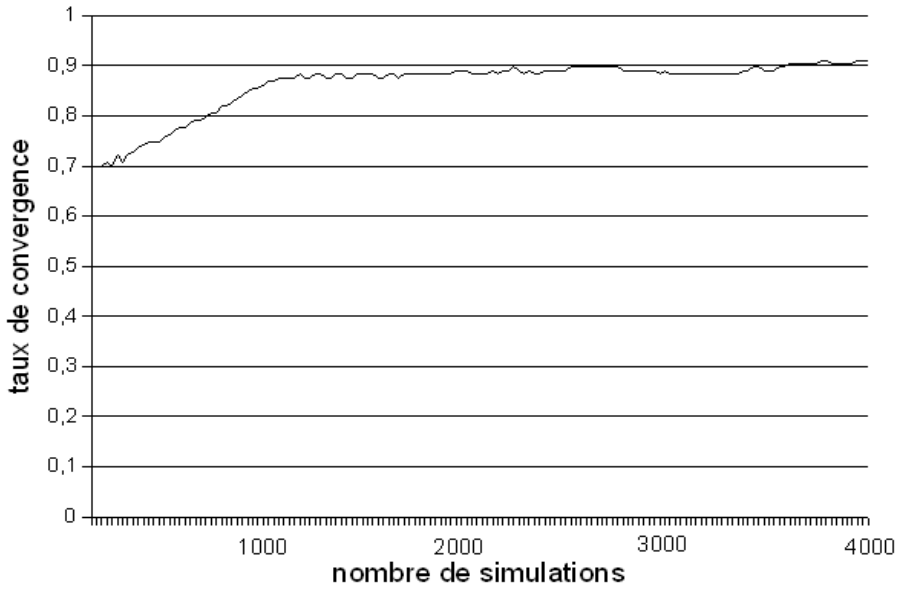

Figure 18: Évolution du taux de convergence π, évalué lors de l'apprentissage de l'expérience 1, entre 0 et 4000 simulations

On constate que l'apprentissage n'apporte pas d'amélioration flagrante pour ce critère entre environ 1500 et 4000 simulations. Le critère π semble converger, et il n'est pas nécessaire de poursuivre l'apprentissage. Un arrêt prématuré, avant 1000 simulations dans le cas présent, amoindrit les performances de contrôle.

Comme pour le choix des états, il y a un équilibre à trouver pour le nombre de simulations. Un nombre trop faible ne permet pas d'atteindre la convergence de la politique de contrôle et diminue ses performances. Un nombre trop élevé augmente la durée de l'apprentissage, sans améliorer les performances à partir d'un certain temps. Cet équilibre dépend du problème de contrôle étudié.

C.3 Choix de la limite du nombre de cycles par simulation

Pour assurer l'arrêt des simulations, le nombre de cycles de contrôle effectués a été limité à 50. Un nombre de cycles trop grand augmente la durée de l'apprentissage, mais un nombre trop petit ne permet pas une estimation suffisante des critères d'évaluation, ce qui leur fait perdre de leur pertinence.

Dans l'expérience 1, on remarque que le nombre moyen ν de cycles avant la convergence d'une simulation vers la cible est de 10, ce qui est largement inférieur à 50. Ceci indique que dans ce cas, la plupart des simulations qui convergent le font bien avant 50 cycles.

On peut le vérifier grâce à la figure 19, qui correspond à une évaluation de la politique apprise dans l'expérience 1 en limitant chaque simulation à 100 cycles. Cette figure représente pour tout nombre de cycles le nombre de simulations ayant convergé vers la cible

une fois ces cycles effectués. On voit que la plupart des simulations convergentes atteignent la cible dans les quelques premières dizaines de cycles. Par exemple, 84% des simulations convergentes atteignent la cible en 30 cycles ou moins[1]. Donc il n'est pas nécessaire de continuer les simulations au-delà du cinquantième cycle pour obtenir une estimation pertinente de la convergence.

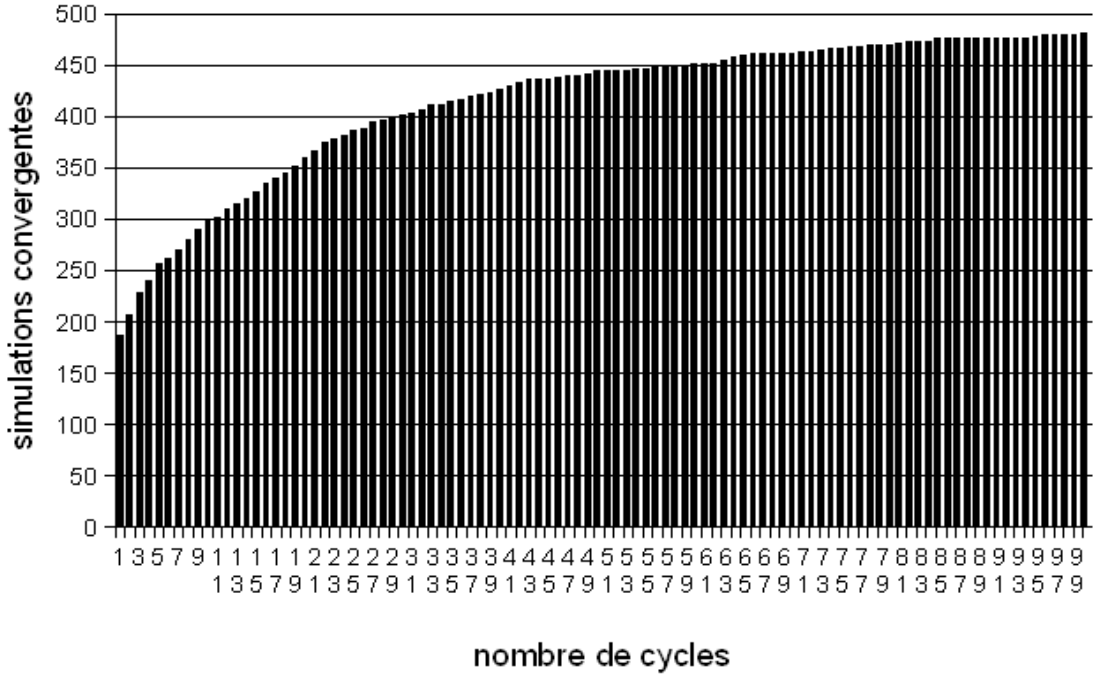

Figure 19: Convergence des simulations de l'expérience 1 vers la cible, en fonction du nombre de cycles

D'un point de vue numérique, π vaut 72% pour k=20, 89% pour k=50 et 96% pour k=100. La limitation du nombre de cycles à 50 semble donc présenter un bon équilibre entre une approximation correcte du taux de convergence (17% de différence avec k=20 contre 7% de différence avec k=100) et une limitation de la durée de l'apprentissage.

Le choix de k a une influence contraire sur les performances et sur la durée de l'apprentissage. Un équilibre entre les deux est susceptible de varier d'un problème de contrôle à un autre.

Nous remarquons que l'estimation du taux de convergence est sous-évaluée, ce qui signifie que notre approche est meilleure que ne l'indique le critère π pour assurer un comportement global cible.

C.4 Influence du type de politique

Le dernier choix est celui du type de politique, utilisée lors de l'apprentissage et surtout pendant l'étape d'exploitation, c'est-à-dire pendant le contrôle proprement dit. Nous avons

1 On a mesuré que le nombre de simulations ayant convergé vers la cible à 30 cycles était de 403 et de 482 pour 100 cycles. 403/482=0,84

utilisé jusqu'à maintenant un seul type de politique stochastique pour le problème p_1, la plus simple possible : le choix des actions se fait proportionnellement aux valeurs Q(s,a) du gain espéré à long terme en effectuant l'action a dans l'état s.

Dans l'expérience 6, on se place dans les mêmes conditions que pour l'expérience 1, et on réutilise les valeurs Q(s,a) calculées lors de cet apprentissage. On évalue le contrôle obtenu en appliquant des politiques différentes à partir de ces valeurs.

Expérience 6: comparaison de différents types de politique

Problème p_1

Mise en oeuvre

On évalue plusieurs politiques construites à partir des Q(s,a) calculés lors de l'expérience 1 :

- la politique initiale, proportionnelle aux valeurs Q(s,a),
- une politique softmax (voir §III.B.3.4),
- une politique ε-gloutonne, avec ε=10%, ce qui est une valeur classique,
- une politique déterministe.

Résultats

On peut comparer les résultats dans le tableau suivant (γ est sans signification ici, puisque toutes ces politiques découlent du même apprentissage) :

Politique	***π (en %)***	***ν***	***τ***
Proportionnelle aux Q(s,a)	89	10	1500
Softmax (T=1)	61	13	2100
ε-gloutonne (ε=10%)	99,8	1,5	500
Déterministe	99,8	1,3	500

Tableau 11: Comparaison de différentes politiques à partir d'un même apprentissage.

On remarque que les politiques qui favorisent le plus l'action qui maximise les Q(s,a) dans chaque état s, au détriment d'un choix aléatoire, amènent aux meilleurs résultats de contrôle. Les politiques déterministe et ε-gloutonne assurent quasiment le comportement cible en moins de 50 cycles. Notons qu'il n'y avait pas de moyen de prévoir a priori qu'une trop grande stochasticité réduirait autant les performances pour le problème p_1. Le choix d'un type de politique a donc une influence fondamentale sur le contrôle.

En nous intéressant à l'influence de la politique sur les problèmes p_2 et p_3, nous étudions l'influence de la valeur de ε pour la politique ε-gloutonne dans l'expérience 7. Nous

constatons qu'une politique plus déterministe fonctionne mieux dans le cas de p_3, tandis qu'une politique davantage stochastique donne de meilleurs résultats au problème p_2. C'est ainsi que nous avons choisi les valeurs de ε dans les expériences 2 et 3. Cette expérience met elle aussi en évidence l'influence du choix de la politique sur les résultats de contrôle.

Expérience 7: influence des valeurs de ε de la politique de contrôle pour les problèmes p_2 et p_3

Problèmes p_2 et p_3

Mise en oeuvre

On apprend et on évalue des politiques ε-gloutonnes sur les problèmes p_2 et p_3, avec les mêmes choix que ceux des expériences 2 et 3, en faisant varier la valeur de ε dans l'ensemble {10, 30, 40}.

Résultats

Valeur de ε	***Problème p_2***	***Problème p_3***
10	π = 90%, υ = 11	π = 67%, υ = 11
30	π = 93%, υ = 8	π = 61%, υ = 13
40	π = 94%, υ = 8	π = 59%, υ = 13

Tableau 12: Comparaison des performances en fonction de ε pour les problèmes p_2 et p_3.

C.5 Conclusion

Les choix étudiés semblent problème-dépendants : il n'y a pas de solution meilleure dans l'absolu, mais ces choix dépendent du problème de contrôle à résoudre. Il s'agit de trouver un équilibre, généralement entre les performances du contrôle et la durée de l'apprentissage.

Lors de l'application de notre proposition, il faut essayer successivement plusieurs solutions et les comparer, jusqu'à obtenir un contrôle satisfaisant. Nous avons présenté plusieurs pistes pour modifier les choix en fonction des résultats. C'est ainsi que l'observation des valeurs Q(s,a) indique les états à découper ou à regrouper, ou que l'observation de la répartition des nombres de cycles nécessaires (figure 19) permet de choisir le nombre k de cycles limite à chaque simulation. Un développement envisageable de notre proposition serait d'établir une méthodologie pour décider des essais successifs à effectuer.

Enfin, la capacité de notre proposition à assurer un comportement cible est meilleure que les 89% obtenus au chapitre B. Nous avons vu que le taux de convergence est sous-estimé de façon non négligeable et qu'il suffit de ne pas stopper les simulations pour augmenter la probabilité d'atteindre la cible. En outre, l'influence des choix est telle qu'il est possible d'améliorer ce taux et quasiment d'assurer la cible. En effet, le choix d'une politique adaptée pour le problème p_1 permet d'atteindre un taux de convergence vers la cible de π=99,8%.

D. Validation de l'approche de contrôle

Nous souhaitons maintenant valider l'approche proposée. Pour cela, nous la comparons à d'autres approches, selon les critères d'évaluation des performances que nous avons définis. Nous validons ainsi :

- le principe du contrôle, en nous comparant à une approche par construction,
- le recours à une optimisation automatique, c'est-à-dire à une phase d'apprentissage,
- la prise en compte de l'effet à long terme des actions, avec une modélisation par MDP.

Nous vérifions également que la proposition possède la propriété, propre au contrôle, de sortir d'un comportement stable non cible. En effet, le cadre d'application est tel que des comportements non souhaités émergent régulièrement dans le SMA, et nous avons vu que c'était le cas dans les problèmes étudiés. Cela justifie le recours à une approche de contrôle plutôt qu'à une approche par construction.

Enfin, nous vérifions que d'autres actions que celles sur les paramètres sont envisageables, telles que celles du problème de contrôle p_4.

D.1 Comparaison à des approches de référence

Nous retenons trois approches de référence auxquelles nous comparons la proposition. Nous appliquons une méthode de calibration pour déterminer les valeurs optimales des paramètres pour les problèmes p_1, p_2 et p_3. Nous utilisons ensuite une politique aléatoire sur ces mêmes problèmes. Enfin, nous évaluons une approche de contrôle naïve sur le problème p_1.

D.1.1 Comparaison à une méthode de calibration

Dans les problèmes étudiés précédemment, chaque moyen d'action consiste à fixer des valeurs de paramètres. Il est donc possible d'appliquer une méthode de calibration pour trouver les paramètres optimaux, soit l'action qui optimise le critère π. Dans ces trois problèmes, il y a suffisamment peu de moyens d'action pour déterminer cette action optimale en évaluant successivement tous les moyens d'action disponibles. Il n'est pas nécessaire de recourir à une solution approchée telle que celles décrites dans l'état de l'art.

La méthode de calibration utilisée consiste à évaluer chaque moyen d'action sur 500 simulations, en l'appliquant à partir de l'initialisation du SMA. La proportion de simulations convergeant vers la cible permet d'estimer le critère π pour comparaison. Avec cette méthode, chaque simulation ne dure qu'un seul cycle, jusqu'à stabilisation du comportement. Le critère γ correspond donc au nombre de simulations effectuées. Pour la même raison, les critères υ et τ ne sont pas pertinents, puisque les simulations ont des durées homogènes.

Cette méthode de calibration est appliquée aux problèmes p_1, p_2 et p_3 respectivement dans les expériences 8, 9 et 10. Nous indiquons à chaque fois la meilleure action trouvée, et nous rappelons les meilleurs résultats obtenus avec l'approche de contrôle proposée.

Expérience 8: utilisation d'une méthode de calibration pour le problème p_1

Problème p_1

Mise en oeuvre : Application de l'approche de calibration

Résultats

La meilleure convergence est obtenue pour une largeur de 185.

Approche	π *(en %)*	γ *(.10³)*
Calibration	93	17,5
Proposition	99,8	57,5

Tableau 13: Évaluation de la calibration sur le problème p_1 et comparaison avec l'évaluation de la proposition (expérience 6).

Rappelons que l'application rapide de la proposition sur le problème p_1, sans remise en cause des choix, amène à π=89% dans l'expérience 1. Nous conservons le résultat de l'expérience 6, qui assure presque l'obtention de la cible grâce à une politique bien choisie.

Expérience 9: utilisation d'une méthode de calibration pour le problème p_2

Problème p_2

Mise en oeuvre : Application de l'approche de calibration

Résultats

La meilleure convergence est obtenue pour les valeurs de paramètres :

- Facteur de mouvement = 1
- Facteur de séparation = 13

Approche	π *(en %)*	γ *(.10³)*
Calibration	15	12,5
Proposition	94	31,5

Tableau 14: Évaluation de la calibration sur le problème p_2 et comparaison avec l'évaluation de la proposition (expérience 2).

Expérience 10: utilisation d'une méthode de calibration pour le problème p_3

Problème p_3

Mise en oeuvre : Application de l'approche de calibration

Résultats

La meilleure convergence est obtenue pour les valeurs de paramètres :

- Facteur de mouvement = 1
- Facteur de séparation = 13
- Vitesse maximale = 1

Approche	π *(en %)*	γ *(.10³)*
Calibration	25	62,5
Proposition	67	71,6

Tableau 15: Évaluation de la calibration sur le problème p_3 et comparaison avec l'évaluation de la proposition (expérience 3).

Les résultats de convergence obtenus avec la proposition sont meilleurs que ceux obtenus avec la calibration pour ces trois problèmes. Le temps d'apprentissage est plus long pour la proposition, mais reste comparable entre les deux approches. Pour les problèmes de contrôle étudiés, l'approche de contrôle proposée est donc préférable pour tenter d'assurer un comportement global.

Remarque : quel que soit le problème, il est toujours possible de trouver une politique de contrôle qui fait aussi bien que la calibration. Il suffit de commencer par appliquer les valeurs de paramètres optimales de la calibration, puis de continuer le contrôle avec une politique quelconque si la cible n'est pas atteinte. Nous avons ainsi appliqué sur le problème p_1 les paramètres de la calibration (π=93%, υ=1) puis le contrôle avec une politique proportionnelle aux valeurs Q(s,a) (π=89%, υ=10), et nous avons obtenu comme performances π=99% et υ=2.

En poursuivant cet argument, il existe théoriquement toujours une politique combinée de calibration et de contrôle meilleure qu'une approche par calibration seule. Mais il faut se demander si le gain obtenu en performances de contrôle vaut la durée d'apprentissage supplémentaire nécessaire pour calculer la politique.

D.1.2 Comparaison à une politique aléatoire

Nous venons de voir qu'une approche par construction, statique, ne suffit pas pour résoudre les problèmes de contrôle étudiés. À l'inverse, la question se pose de savoir si seul le

caractère dynamique du contrôle, qui agit en cours d'évolution du SMA, est responsable des bons résultats de la proposition. En effet, les actions de contrôle successives perturbent le SMA, et peuvent l'amener par hasard à présenter le comportement cible, sans qu'il soit besoin d'optimiser le choix de ces actions.

Pour déterminer si une approche aussi simple suffit, nous évaluons une politique aléatoire : dans chaque état, chaque moyen d'action a une chance équiprobable d'être sélectionné comme action de contrôle et effectué. Une telle politique aléatoire n'a pas besoin de phase d'apprentissage, donc la valeur du critère γ est toujours nulle. Les performances de cette politique aléatoire sont évaluées sur 500 simulations, et comparées à celles obtenues avec la proposition, sur les problèmes p_1, p_2 et p_3, dans l'expérience 11[1].

Expérience 11: évaluation et comparaison d'une politique aléatoire sur les problèmes p_1, p_2 et p_3

Problèmes p_1, p_2 et p_3

Mise en oeuvre : évaluation de la politique aléatoire

Résultats

Politique	***π (en %)***	***ν***	***τ***
Aléatoire	71	12	1900
Calculée	99,8	1,3	500

Tableau 16: Comparaison de politiques aléatoire et calculée pour le problème p_1.

Politique	***π (en %)***	***ν***	***τ***
Aléatoire	69	15	2200
Calculée	94	8	1300

Tableau 17: Comparaison de politiques aléatoire et calculée pour le problème p_2.

Politique	***π (en %)***	***ν***	***τ***
Aléatoire	23	16	2200
Calculée	67	11	1700

Tableau 18: Comparaison de politiques aléatoire et calculée pour le problème p_3.

Lorsque la politique aléatoire est utilisée, le taux de convergence π chute par rapport à notre approche, et le nombre de cycles et de pas de simulation nécessaires pour atteindre la cible augmente. Cette différence de résultats montre que la phase d'apprentissage de notre

1 Comme nous l'avons annoncé au chapitre C.1, les résultats de l'expérience 5 sont semblables à ceux d'une politique aléatoire sur le problème p_1.

proposition a permis de prévoir en partie l'évolution du SMA, pour trouver de bonnes actions à effectuer dans chaque état.

Pour le problème p_2, nous avons vu qu'une grande part d'aléatoire dans la politique était nécessaire pour obtenir de meilleures performances de contrôle. Toutefois, nous observons maintenant qu'une politique trop aléatoire diminue ces performances. Il y a donc un équilibre à trouver entre une politique déterministe et une politique aléatoire pour optimiser le contrôle.

Problème	p_1	p_2	p_3
Calibration	93%	15%	25%
Politique aléatoire	71%	69%	23%

Tableau 19: Valeurs de π pour la calibration et la politique aléatoire

Le tableau 19 compare les résultats obtenus par la calibration à ceux du contrôle aléatoire pour les problèmes p_1, p_2 et p_3. On remarque qu'il n'existe pas de règle permettant de prévoir quelle approche est la meilleure. Les différents problèmes semblent posséder des propriétés intrinsèques, indépendantes de notre proposition. Par exemple, on remarque que plus il y a d'actions de contrôle disponibles, et moins la politique aléatoire permet d'atteindre la cible. Le problème lui-même semble alors plus difficile. Une perspective de travail serait de définir ces propriétés, afin de mieux adapter la proposition au problème à résoudre.

D.1.3 Comparaison à un contrôle naïf

Nous appelons contrôle naïf l'application d'une politique qui choisit l'action qui optimise la probabilité d'atteindre la cible à partir de l'état courant. Il s'agit donc d'un choix des moyens d'action selon leur influence à court terme. Le calcul de cette politique ne nécessite pas de nouvel apprentissage, puisqu'il est possible de la trouver à partir des valeurs des transitions entre états qui ont été calculées dans notre modèle.

Pour le problème p_1, les transitions liées à cette politique naïve sont représentées sur le graphe markovien de la figure 20. Sous chaque état est encadrée la valeur de l'action qui maximise la probabilité d'atteindre la cible, et les probabilités de transition en effectuant cette action sont indiqués sur les arcs. Par exemple, dans l'état S_0, l'action qui consiste à fixer la largeur à 185 donne la meilleure probabilité d'atteindre la cible, avec 93,4%.

En partant d'une probabilité de 100% de se trouver initialement dans un état donné, on peut, à l'aide du graphe, calculer les probabilités successives de se trouver dans chaque état en suivant cette politique. Si l'on note $p_t(s)$ la probabilité d'être dans l'état s au cycle t, T(s',s) la transition entre deux états s' et s en suivant la politique naïve, et $\Gamma^{-1}(s)$ l'ensemble des antécédents de l'état s par cette transition, on a

$$p_t(s) = \sum_{s' \in \Gamma^{-1}(s)} p_{t-1}(s').T(s',s)$$

En particulier, on peut calculer la probabilité d'atteindre la cible au bout de 50 cycles de simulation, donc estimer le taux de convergence π.

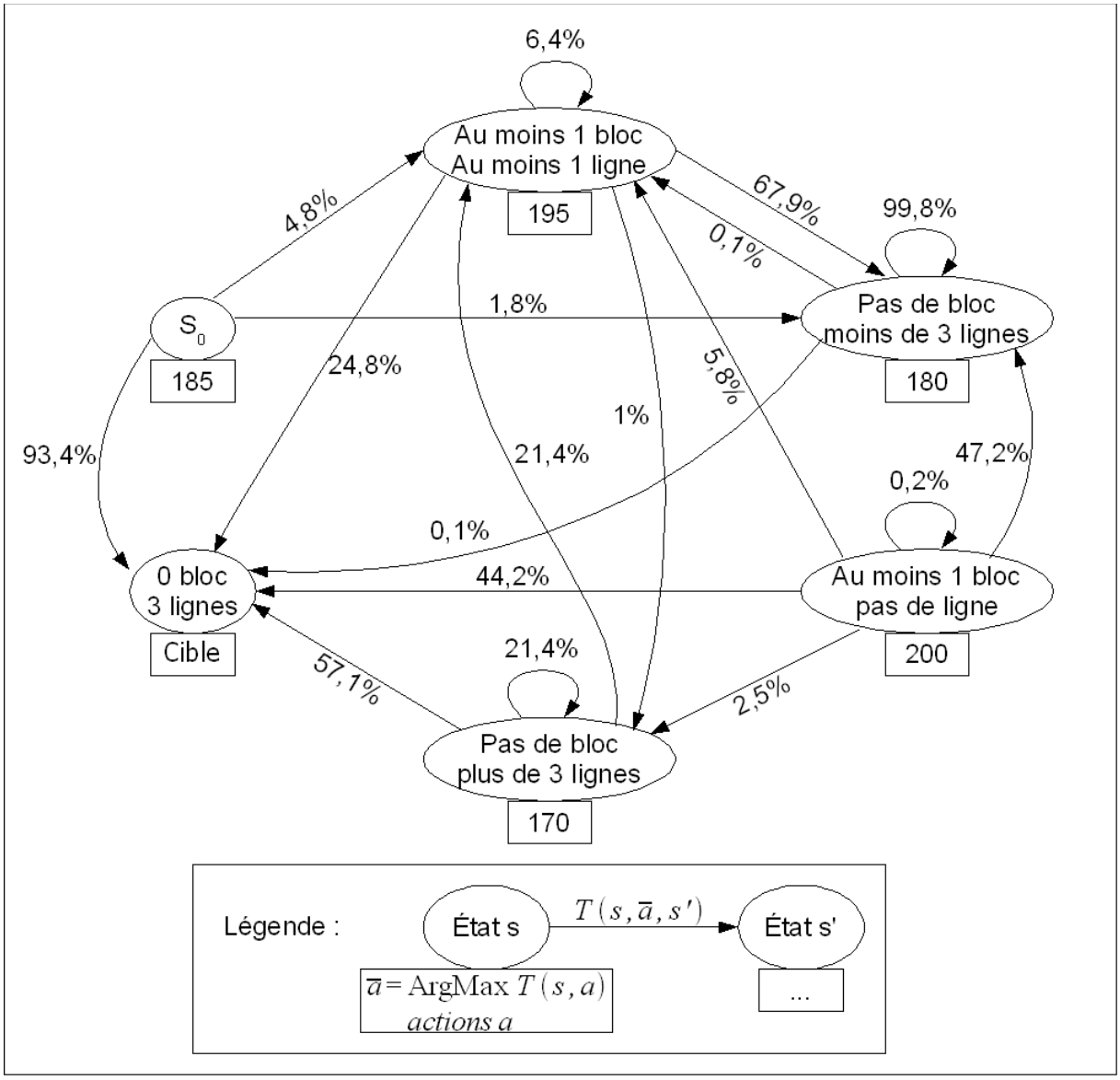

Figure 20: Transitions pour une politique naïve appliquée au problème p_1

Ce calcul à partir du graphe montre qu'en partant de l'état S_0, il y a 95% de probabilité d'aboutir dans l'état cible en moins de 50 cycles. Ce résultat est à comparer aux 99,8% de convergence obtenus avec une politique déterministe qui choisit les actions qui maximisent les valeurs Q(s,a) (évaluation des actions à long terme), là où la politique naïve choisit les actions qui maximisent les valeurs T(s,a) (évaluation à court terme). Le contrôle naïf ne permet pas d'assurer le comportement cible pour ce problème, même s'il est acceptable.

La différence est encore plus flagrante si les simulations commencent dans l'état [pas de bloc, moins de trois lignes], car dans ce cas, la politique naïve n'atteint plus que 6% de taux de convergence. L'initialisation dans un tel état fait l'objet du chapitre suivant.

D.2 Contrôle après initialisation dans un état stable non cible

L'un des problèmes auxquels doit répondre une approche de contrôle est la maîtrise du comportement global même quand le SMA présente un comportement stable différent de la cible. Ce cas peut se présenter pour deux raisons :

- L'approche utilisée n'assure pas d'atteindre la cible à 100% lors de la première action, comme le ferait une approche par construction, et a échoué dans la simulation considérée, en raison des conditions initiales ou de l'évolution aléatoire.
- Une perturbation du SMA a modifié son comportement pour l'amener dans celui observé à l'instant présent, différent de la cible.

Pour le problème p_1, nous supposons que le comportement présenté à un moment donné est constitué de 2 lignes et aucun bloc. Il s'agit d'un comportement très fréquent. Le problème est donc transformé : il s'agit de sortir de cet état pour atteindre la cible. Les simulations sont toujours limitées à 50 cycles de contrôle, mais sont maintenant initialisées dans cet état non cible. La largeur initiale de l'environnement est fixée à 185, c'est-à-dire la valeur optimale trouvée par la calibration dans l'expérience 8. L'état où le SMA présente deux lignes et aucun bloc est bien stable pour cette largeur.

Dans l'expérience 12, nous considérons ce nouveau problème dérivé de p_1, et nous appliquons la politique déterministe et la politique proportionnelle aux Q(s,a), obtenues respectivement dans les expériences 6 et 1, pour le résoudre. La politique déterministe reste meilleure que l'autre. Elle permet bien de sortir du comportement stable non cible initial.

Expérience 12: initialisation dans un comportement non cible pour le problème p_1

Problème p_1 modifié

Les simulations commencent lorsque le SMA présente le comportement stable de 0 bloc et 2 lignes, avec une largeur de 185, c'est-à-dire la valeur optimale pour la calibration.

Mise en oeuvre

La politique proportionnelle au Q(s,a) apprise dans l'expérience 1 et la politique déterministe de l'expérience 6 sont réutilisées dans ce cas, et évaluée sur 500 simulations.

Résultats

Politique utilisée	Π *(en %)*	ν
Déterministe	99,6	1,3
Proportionnelle	80	20

Tableau 20: Évaluation des méthodes lorsque la simulation est initialisée dans un comportement stable non désiré.

Le contrôle proposé semble finalement vérifier la propriété de sortir d'un comportement stable non cible et assure le contrôle vers la cible. La politique naïve présentée précédemment, par exemple, ne vérifie pas cette propriété, puisque l'évaluation dans ce cas montre que le critère π chute à 6% (voir chapitre précédent). Une approche de calibration, par nature statique, ne permet pas de sortir de l'état de par sa stabilité (nous l'avons vérifié expérimentalement, π=0%). Une politique aléatoire appliquée dans les mêmes conditions, quant à elle, présente seulement 58% de convergence vers la cible[1].

D.3 Utilisation de leurres

L'ensemble $\mathcal{A}$ d'actions défini pour le problème p_4 consiste à introduire des leurres dans le SMA, c'est-à-dire un petit nombre d'agents au comportement individuel différent de celui des autres agents. L'idée d'ajouter quelques leurres à un système réactif est proposée dans [Halloy 07] pour diriger des animaux réels à l'aide de robots. L'introduction de leurres peut être vue comme une perturbation du SMA par l'utilisateur.

Le principe est d'agir sur le SMA en lui ajoutant quelque chose de nouveau, sans modifier les agents ni l'environnement. Le comportement global du système découle à la fois des comportements individuels des agents "normaux" et de ceux des leurres. Il est donc envisageable de contrôler le SMA en ne dirigeant que ces leurres. Nous montrons de cette manière que les moyens d'action que peut traiter notre approche ne concernent pas seulement les paramètres, mais peuvent être plus élaborés.

Le comportement individuel des leurres est choisi de manière à désordonner le SMA pour sortir d'un comportement stable. Ils ont pour caractéristiques une direction objectif $\vec{u}$ aléatoire qui a 10% de chances d'être modifiée à chaque pas de temps, et les paramètres de comportement individuel suivants : vitesse maximale = 5, F_M=5, F_B=5, F_S=1, F_E=1.

Lorsque des leurres sont présents dans le système, il n'y a plus de comportements identifiables tel que nous les avons définis jusqu'ici : les agents ne restent pas en lignes ou en blocs stables. Il est donc nécessaire de retirer tous les leurres pour atteindre la cible.

Dans l'expérience 13, le contrôle agit sur le SMA en ajoutant des leurres, en attendant 500 pas de simulation, puis en retirant les leurres. Pour une raison de cohérence avec les autre expériences, une telle perturbation compte pour 2 actions de contrôle, donc deux cycles. Comme au §D.2, le SMA est initialisé dans l'état stable qui comporte 2 lignes et aucun bloc, avec une largeur de 185. Nous observons ainsi la capacité d'une perturbation à faire revenir le SMA dans le comportement cible. Cette expérience est effectuée pour un nombre de leurres, ajoutés puis retirés, compris entre 1 et 10.

Cette expérience revient à considérer 3 états de contrôle : la cible (pas d'action appliquée), un comportement différent de la cible en l'absence de leurres (y compris S_0, on ajoute alors des leurres), et un comportement non cible avec des leurres présents (on retire alors les leurres). Il n'y a pas d'apprentissage à effectuer avec ces règles simples. Chaque couple d'actions est évalué sur 500 simulations de k=50 cycles maximum.

1 Nous avons également obtenu ce résultat expérimentalement, mais nous ne développons pas cette expérience, similaire à celles du §D.1.2.

Expérience 13: Utilisation de leurres pour le problème p_4

Problème p_4

Initialisation = 0 blocs et 2 lignes, sans leurres, largeur de l'environnement = 185.

Mise en oeuvre

Nous voulons connaître le nombre optimal de leurres à ajouter puis retirer du SMA pour atteindre la cible. Nous étudions donc distinctement chaque nombre n compris entre 1 et 10, en ajoutant n agents puis en les retirant jusqu'à atteindre la cible.

Résultats

Nombre de leurres ajoutés et retirés à chaque action	1	2	3	4	5	6	7	8	9	10
π (en %)	0	2,2	3,4	6,8	9,2	8,6	8,8	9,4	10,4	11,2
ν	Ø	24	30	25	23	27	23	23	25	26
τ (.10³)	Ø	4,8	9	10,7	11,4	14,9	12,9	13,3	14,8	15,2

Tableau 21: Évaluation du contrôle en ajoutant et en retirant un nombre fixe de leurres successivement dans le SMA, jusqu'à atteindre la cible.

Lorsque le nombre de leurres est trop faible, la perturbation a moins de chances de modifier le comportement du système. Nous constatons qu'à partir de 5 leurres, la probabilité de passer de 2 à 3 lignes d'agents est à peu près constante. La valeur de ν est d'environ 25 dans toutes les expériences pour k=50 actions maximum par simulation. Cela indique que la probabilité d'obtenir la cible en un couple d'actions (ajout et retrait de leurres) est constante. Les expériences montrent que cette probabilité est faible, mais non négligeable[1]. Ainsi, un contrôle qui ne limite pas le nombre de cycles tend toujours à atteindre la cible, mais en un temps éventuellement très long.

Nous remarquons que la durée moyenne des simulations convergentes, en nombre de pas, augmente avec le nombre de leurres introduits, alors que le nombre moyen d'actions reste constant. Même si nous ne pouvons pas l'expliquer avec certitude, nous en déduisons qu'il est intéressant de limiter le nombre de leurres utilisés pour converger plus rapidement vers la cible. La limitation du nombre de leurres s'impose aussi si l'introduction d'un leurre a un coût important, comme la construction d'un robot-leurre dans un système de robots. Dans les conditions de l'expérience, l'utilisation de 5 leurres semble donc adéquate et équilibrée entre le coût d'une simulation et la capacité de contrôle du SMA vers la cible.

1 Pour π=10% en 25 couples d'actions, cette probabilité est de 0,42% : $1-0{,}9^{\frac{1}{25}} \approx 0{,}0042$

Toutefois, le taux de convergence vers la cible est décevant. En effectuant les mêmes expériences « à la main », c'est-à-dire en choisissant empiriquement le moment où les leurres sont retirés du SMA uniquement à partir de son observation directe, les performances de contrôle sont bien meilleures. Par exemple, nous avons réussi à passer de 2 à 3 lignes en 29 essais sur 40, en ajoutant puis en retirant 10 leurres une seule fois. En 25 couples d'actions successifs pour une même simulation (k=50), cela correspond à une convergence de $\pi \simeq 100\%$. Cela indique qu'un humain identifie des situations différentes lorsque des leurres sont présents dans le SMA. Distinguer ces situations par des états globaux mesurés automatiquement devrait permettre d'augmenter les performances du contrôle. Comme nous l'avons expliqué dans la proposition, l'étape du choix des états du modèle a été trop simplifiée, avec seulement trois états globaux retenus, et peut être améliorée.

Finalement, l'utilisation de leurres est une approche prometteuse pour un contrôle avec des états adaptés, mais qui demande d'effectuer intégralement une nouvelle étude, y compris pour la mesure des comportements globaux. Il semble possible de contrôler un SMA avec de telles actions, différentes de la modification des valeurs des paramètres du système.

D.4 Conclusion

Nous avons montré dans ce chapitre que :

- Le caractère dynamique du contrôle est essentiel : en l'opposant à une approche par construction (D.1.1), nous montrons que pour le problème étudié, il faut passer par une approche de contrôle et par la définition d'états de contrôle.
- Une optimisation est nécessaire (D.1.2), donc il n'est pas possible de se passer de l'étape d'apprentissage.
- Il faut prendre en compte l'influence à long terme des actions. Ce besoin est montré dans les sous-chapitres D.1.3 et D.2. Cela correspond pour nous à l'utilisation d'un MDP.

L'approche proposée fait mieux que d'autres approches pour les problèmes donnés selon les critères d'évaluation π, υ et τ, jusqu'à assurer la cible presque à 100% pour le problème p_1, moyennant une augmentation de la durée γ de l'apprentissage. Cela montre qu'il est possible de mieux diriger le comportement global grâce à la proposition qu'avec des approches de référence. Le recours à cette proposition est donc validé.

Nous avons vérifié par ailleurs que la proposition permettait bien de sortir d'un comportement stable non cible (D.2), conformément à ce qu'une approche de contrôle doit faire, et qu'elle permettait d'utiliser des actions différentes de la modification de paramètres (D.3). Nous validons ainsi en partie le cadre sur lequel nous avons présenté la proposition au chapitre III : le SMA peut admettre plusieurs comportements globaux différents, et l'ensemble des moyens d'action disponibles peut être varié. L'utilisation de leurres est un moyen d'influer sur les comportements individuels même s'il n'est pas possible d'agir sur les agents déjà présents dans le SMA.

E. Modification du contexte d'application

L'approche que nous avons proposée est fondée sur l'existence de régularités au niveau global du SMA. Nous voulons nous assurer que les mêmes régularités se retrouvent dans des contextes proches, pour des SMA similaires. Ainsi, la politique de contrôle peut être transposée d'un contexte à un autre. Par exemple, si une politique a été apprise sur un SMA donné, et que ce SMA évolue par la suite de manière inattendue (panne d'un agent par exemple), il est souhaitable que la politique de contrôle conserve de bonnes performances sur ce nouveau système, sans qu'il soit besoin de réaliser un nouvel apprentissage.

Notre objectif ici est donc de montrer que les régularités du SMA sont telles qu'il n'est pas nécessaire de le connaître parfaitement lors de l'apprentissage. Pour cela, nous vérifions qu'une politique apprise pour un problème est robuste lorsque les conditions de simulation sont différentes entre l'apprentissage et le contrôle. Les données du problème restent identiques, mais le SMA lui-même est modifié.

E.1 Expériences

Nous envisageons de modifier le SMA par rapport à :

- la manière dont chaque simulation est initialisée,
- les comportements individuels des agents.

Ce sont deux propriétés propres au SMA, et suffisamment différentes pour vérifier sur deux axes la robustesse d'une politique apprise. Nous les étudions sur le problème p_1.

L'expérience 12 nous a permis de voir que l'initialisation des simulations dans un état stable différent de la cible, plutôt qu'en les introduisant un par un (initialisation standard), ne modifiait pas les résultats de contrôle d'une politique déterministe pour le problème p_1 (plus de 99% de convergence dans les deux cas).

Ici, dans l'expérience 14, nous proposons d'initialiser les simulations d'une troisième façon, aléatoire : en répartissant tous les agents dans l'environnement, avec des positions aléatoires et des vitesses nulles. Des blocs d'agents apparaissent fréquemment avec cette initialisation, ce qui donne des performances de contrôle inférieures. Nous évaluons deux politiques proportionnelles aux valeurs des Q(s,a), apprises respectivement avec une initialisation standard et une initialisation aléatoire. Elles sont évaluées avec ces deux mêmes types d'initialisation.

Les performances de ces deux politiques sont similaires lorsqu'elles sont évaluées dans des conditions de simulation identiques, c'est-à-dire avec le même type d'initialisation, alors que les conditions d'apprentissage sont différentes. Cela confirme leur robustesse lorsque l'initialisation du SMA varie, ou encore que les régularités du SMA font qu'il conserve des réactions similaires.

Dans l'expérience 15, nous reprenons le problèmes p_1, mais nous changeons deux des paramètres qui définissent le comportement des agents :

- facteur de mouvement = 3 au lieu de 4
- facteur d'évitement = 25 au lieu de 15

Expérience 14: robustesse par rapport à l'initialisation pour le problème p_1

Problème p_1

Mise en oeuvre

La politique apprise dans l'expérience 1 (notée A) avec une initialisation standard est comparée à une politique apprise avec une initialisation aléatoire des agents (notée B).

Résultats

Politique évaluée / ***Contrôle avec initialisation***	A	B
Standard	89%	86%
Aléatoire	77%	78%

Tableau 22: Comparaison de π pour deux politiques apprises avec des initialisations différentes, et évaluées avec ces mêmes initialisations.

Expérience 15: robustesse par rapport aux comportements individuels pour le problème p_1

Problème p_1

Mise en oeuvre

La politique apprise dans l'expérience 1 (notée A) est comparée à une politique apprise avec des valeurs de paramètres individuels différents (notée B).

Résultats

Politique évaluée / ***Contrôle avec paramètres***	A	B
Standards	$\pi = 89\%, \nu = 10$	$\pi = 89\%, \nu = 10$
Modifiés	$\pi = 85\%, \nu = 12$	$\pi = 92\%, \nu = 11$

Tableau 23: Comparaison de π et ν pour deux politiques apprises avec des initialisations différentes, et évaluées avec ces mêmes initialisations.

La politique apprise avec des valeurs de paramètres modifiés (B) a des performances équivalentes à celles de la politique apprise avec des valeurs de paramètres standards (A) lorsqu'il s'agit de contrôler le SMA avec des paramètres standards. La politique B est légèrement meilleure que A lorsqu'il s'agit de contrôler le SMA avec des valeurs de paramètres modifiés, mais reste comparable. On en déduit que chaque politique est robuste par rapport à cette modification des comportements individuels, donc que le SMA présente des évolutions semblables de par ses régularités au niveau global.

Nous ajoutons une autre expérience, sur un problème original, afin de s'assurer que la robustesse des politiques n'est pas liée au problème p_1 uniquement. La modification du SMA concerne cette fois le nombre d'agents présents dans le système.

Dans l'expérience 16, on considère le problème de n'avoir que des lignes d'agents en agissant sur les facteurs d'évitement et de mouvement, et on apprend deux politiques selon que le SMA comporte 24 ou 48 agents. Elles sont ensuite évaluées chacune avec 24 et 48 agents présents dans le SMA.

Expérience 16: robustesse par rapport au nombre d'agents (présents ou contrôlés)

Problème

cible = aucun bloc

moyens d'action = ensembles 5 et 6 (facteur d'évitement et facteur de mouvement)

Mise en oeuvre

Apprentissage et comparaison de deux politiques lorsque le SMA comporte 24 et 48 agents.

Apprentissage sur 4000 simulations, avec 12 états (nombre de blocs ∈ {0, 1, 2 ou plus}, nombre de lignes ∈ {0, 1, 2, 3 ou plus}), et une politique ε-gloutonne avec ε=10%.

Résultats

Apprentissage avec / ***Évaluation du contrôle avec***	24 agents	48 agents
24 agents	85%	86%
48 agents	84%	84%

Tableau 24: Comparaison de π pour deux politiques apprises avec des nombre d'agents différents, et évaluées avec 24 à 48 agents.

Le contrôle réalisé avec ces deux politiques n'apparaît pas sensible au nombre d'agents, car elles ne présentent pas de différences importantes en ce qui concerne le taux de convergence vers la cible. Ces politiques sont donc elles aussi robustes, par rapport au nombre d'agents et sur un problème différent. Là encore, l'évolution du système est robuste.

E.2 Conclusion

Sur quelques expériences, nous constatons que les politiques de contrôle calculées avec notre approche sont robustes face à des petites modifications du SMA : elles permettent de conserver de bonnes performances de contrôle, comparables à celles qu'elles présentent sur le SMA non modifié. Un nouvel apprentissage sur le SMA modifié n'améliore pas significativement les résultats. Seule une évaluation de l'ancienne politique sur le SMA modifié est nécessaire, pour s'assurer qu'elle fonctionne toujours, ce qui est avantageux en termes de durée de mise en oeuvre du contrôle.

Nous en déduisons que les régularités, présentées au niveau global par le SMA étudié, varient peu lorsqu'il est modifié. Notre approche, qui s'appuie sur ces régularités, profite de cette faible variation. L'intérêt est que les politiques de contrôle apprises peuvent être transposées d'un contexte d'application à un autre : l'utilisateur n'est pas obligé de connaître parfaitement ce contexte pour déterminer une politique.

Une utilisation de cette propriété est par exemple de contrôler une famille de systèmes multi-agents en n'apprenant qu'une seule politique. Cela ouvre des perspectives d'application de notre proposition à des problèmes en-dehors du cadre d'application que nous avons défini au chapitre III.

F. Discussion

Dans ce chapitre, la proposition a été appliquée avec succès sur des problèmes de contrôle. Nous discutons ici des conséquences des résultats obtenus sur la possibilité de modéliser la dynamique globale d'un SMA, son implication sur la maîtrise du comportement, et sur la mise en oeuvre de la proposition. Nous évoquons aussi certaines questions laissées en suspens – sur les moyens d'action et d'observation, et sur la difficulté du problème de contrôle à résoudre – qui méritent d'être approfondies.

F.1 Modélisation de la dynamique globale

L'idée de considérer la dynamique globale du SMA pour le contrôler est pertinente et fondée. En effet, les comportements du système présentent une régularité qui permet de prévoir son évolution. La connaissance du comportement courant du SMA permet de différencier l'influence des actions de contrôle. Nous l'avons déterminé en montrant qu'une phase d'apprentissage est nécessaire par rapport à l'utilisation d'une politique aléatoire. La modélisation de la dynamique globale reste prédictive lorsque le SMA est modifié, puisque la politique apprise est robuste. Il peut néanmoins exister d'autres régularités, qui permettraient de contrôler le système d'une façon différente ou d'améliorer le contrôle en précisant les états du modèle. La principale difficulté de la proposition est d'identifier des états mesurables qui distinguent les influences des actions.

La modélisation est possible dans la mesure où le temps d'apprentissage est acceptable. Il est comparable à celui de la calibration, et reste raisonnable si les ensembles d'actions et d'états à explorer sont restreints. Cet apprentissage a été réussi en quelques heures sur les problèmes expérimentaux étudiés.

F.2 Maîtrise du comportement par un contrôle au niveau global

Un contrôle effectif du SMA est rendu possible par le modèle appris. Si le système est ouvert, il peut subir des perturbations qui l'amènent à présenter un comportement stable non souhaité. Le contrôle permet alors de sortir de ce comportement pour revenir à la cible. La cible est atteinte fréquemment et rapidement : en quelques milliers de pas de simulation et environ 10 actions de contrôle. Mais le comportement cible n'est pas toujours assuré. Un développement possible de ce travail est d'étudier la difficulté propre d'un problème de contrôle.

Les expériences de ce chapitre ont permis de valider les outils employés. Ainsi, le recours à un modèle markovien est justifié par le fait que les états fondés sur les comportements courants sont suffisants. L'apprentissage par renforcement permet de considérer les effets à long terme des actions de contrôle. Nous avons vu que cela est nécessaire pour obtenir des résultats satisfaisants. Toutefois, d'autres outils peuvent se révéler tout aussi valides. Une perspective de ce travail est d'étudier d'autres méthodes d'apprentissage, en particulier les réseaux de neurones.

Notre approche de contrôle améliore la maîtrise du comportement par rapport à d'autres approches de référence. La contrôlabilité est meilleure que celle obtenue par une approche de calibration. Le fait que plusieurs actions sont nécessaires avec le contrôle prouve d'ailleurs que la cible, dans les problèmes étudiés, est difficile à atteindre dès la première action. Par rapport à l'application d'une politique aléatoire, notre proposition permet d'atteindre la cible plus souvent et plus rapidement.

Mais le recours à une approche de contrôle n'assure pas une amélioration des résultats par rapport à une approche de calibration. En pratique, notre proposition peut même se révéler moins bonne que ces approches, si les choix des étapes sont mal effectués (expérience 1). En conséquence, un développement intéressant de la proposition consiste à développer une méthodologie pour effectuer les choix successifs qu'elle nécessite. Cette méthodologie pourrait aussi servir à décider rapidement si le gain de performances justifie la durée des apprentissages successifs. Enfin, une combinaison entre une approche par calibration et un contrôle fait au moins aussi bien que les deux approches prises séparément.

F.3 Mise en oeuvre de la proposition

La phase d'apprentissage de la proposition est automatisée, et ne nécessite donc pas de temps pour un humain. La seule limite de cette phase est le temps d'apprentissage nécessaire. Il dépend du nombre d'états et d'actions du modèle.

En revanche, un humain doit choisir le modèle à apprendre, en fonction de sa propre connaissance du SMA et des effets des actions. Il s'agit de ce que nous avons appelé les choix aux étapes. Ils sont difficiles car ils dépendent du problème de contrôle à résoudre, donc ils ne sont pas généralisables, et ils ont une influence sur les performances du contrôle. Plusieurs essais et plusieurs apprentissages peuvent s'avérer nécessaires avant de trouver un modèle satisfaisant, ce qui augmente le temps pour appliquer la proposition. C'est pourquoi nous avons fourni des pistes pour effectuer efficacement ces choix. De ce point de vue, un avantage de notre approche est de fournir des politiques de contrôle robustes, qui ne requièrent pas nécessairement un nouveau choix de modèle et un nouvel apprentissage lorsque le SMA est modifié.

F.4 Questions en suspens

Nous avons proposé un type d'action original sur le SMA : l'utilisation de leurres. L'idée est de profiter de l'ouverture du système pour ajouter des agents dédiés au contrôle, sans modifier les agents déjà présents. Nous avons vu que les résultats trouvés automatiquement n'étaient pas à la hauteur de ce que l'on attendait en les utilisant "à la main". Nous avons expliqué que cela provenait d'un mauvais choix d'états de contrôle, trop différents de ceux que nous avions utilisés jusqu'alors. Néanmoins, l'idée d'intervenir sur le SMA sans le modifier fondamentalement est intéressante, et l'étude de ces leurres mérite d'être approfondie. Il faut avant tout pour cela caractériser les comportements du SMA lorsque les leurres sont présents et définir les états de contrôle correspondants.

Nous avons étudié les variations du SMA à contrôler, mais pas celles des problèmes de contrôle. Or il pourrait être intéressant de simplifier ces problèmes avant de les résoudre. Par exemple, l'expérience 17 (en annexe A) montre qu'il est possible d'améliorer le contrôle en diminuant l'ensemble des moyens d'action à explorer. Cette question peut être approfondie en s'appuyant sur une bibliographie que nous avons en partie citée au chapitre II.B.1.2, et en particulier [Fehler 04].

Nous avons considéré que toutes les informations sur l'état du SMA sont disponibles, c'est-à-dire que les moyens d'observation sont illimités. À cause du caractère décentralisé du SMA, il est possible que seule une partie de l'état du système soit observable. Dans l'expérience 18 (en annexe B), nous essayons de contrôler un SMA avec uniquement des informations locales. Cette expérience montre que la connaissance d'états décrits au niveau local est insuffisante pour assurer la maîtrise du comportement global. Un axe de recherche pourrait s'avérer intéressant : estimer les comportements globaux à partir d'informations locales, et considérer la dynamique du SMA comme un POMDP.

CHAPITRE VI - CONCLUSION ET TRAVAUX FUTURS

« Même pour des systèmes physiques assez simples, il n'est pas possible de négliger l'effet du composé sur les composants et il faut reconnaître que la relation entre le tout et ses parties n'est pas unilatérale. »

« Les notions macroscopiques appartiennent à un autre cadre conceptuel que celui de la mécanique microscopique, et ne sauraient émerger spontanément au sein d'une théorie dont elles ne relèvent pas. » - Jean-Marc Lévy-Leblond [Lévy-Leblond 96]

Ce travail se situe dans le cadre des systèmes multi-agents, et traite de la maîtrise de leur comportement global. Le problème est de trouver comment faire – à la conception du système ou lors de son fonctionnement – pour garantir l'obtention d'un comportement pour lequel il est construit. Dans cette thèse nous proposons une approche de contrôle au niveau global. Il s'agit d'effectuer des actions sur le SMA au cours de son évolution, lesquelles sont choisies en fonction du comportement global qu'il présente.

Nous proposons une approche de contrôle en deux phases.

Lors de la première phase, la dynamique du SMA est modélisée puis apprise : on représente l'évolution du comportement global du SMA, en fonction des actions de contrôle qui sont appliquées. Cette évolution est représentée sous la forme d'un graphe dans lequel les nœuds correspondent aux comportements globaux et les arcs aux changement de comportements lors de l'application d'une action de contrôle. Des outils d'apprentissage par renforcement permettent de construire ce graphe et de calculer une politique de contrôle, qui indique quelle action effectuer en fonction de l'état courant.

La seconde phase consiste à exploiter cette politique pour contrôler le SMA.

L'originalité de notre proposition est de s'appuyer sur la dynamique du SMA décrite à son niveau global. Ainsi, les différents comportements du SMA sont exprimés dans notre proposition au même niveau de description que celui du comportement cible.

Nous avons montré sur un exemple que cette approche fonctionne et qu'elle présente plusieurs avantages. Elle permet d'explorer sans intervention humaine l'ensemble des possibilités et de proposer une politique qui soit optimale (selon un critère particulier). Son caractère dynamique permet de sortir de comportements non souhaités (par exemple obtenus suite à une perturbation).

Néanmoins, nous avons vu que le choix des comportements à modéliser a une influence sur les performances de la politique de contrôle. Par ailleurs la caractérisation de ces comportements est primordiale. Il faut être capable de proposer un moyen qui, à partir d'une description locale du SMA, fasse correspondre un comportement global : sans cette correspondance, il est difficile voire impossible d'appliquer notre proposition. A ce titre nous avons proposé un algorithme de clustering original pour effectuer cette correspondance sur le système étudié.

Nous envisageons la poursuite de ce travail selon plusieurs directions.

Nous avons proposé un cadre pour le contrôle dont la généralité doit être éprouvée en appliquant cette proposition sur d'autres exemples de SMA, à la fois académiques et industriels[1].

Par ailleurs, il sera intéressant d'étudier d'autres mises en œuvre en proposant des types d'actions autres que des modifications de valeur de paramètres. Nous avons esquissé un premier travail dans cette direction avec l'utilisation de leurres.

Une extension de ce travail concerne la mise en œuvre et l'évaluation des performances de la proposition lorsque celle-ci s'appuie sur une estimation locale du comportement global.

Enfin à plus long terme, nous envisageons la proposition d'une méthodologie pour contrôler le comportement d'un SMA. En particulier, nous voulons fournir un guide pour faciliter le choix des états de contrôle appropriés à un problème, et pour les réviser en fonction des résultats de contrôle.

1 En particulier, dans le cadre de la société *Intuitive Machine* qui est une SARL créée par cinq associés, dont l'auteur, en décembre 2007, distinguée par la médaille du Loria 2007. Son principal projet est le développement d'un ADIC (Assistant Domestique Intelligent et Communiquant), c'est-à-dire un appareil capable de communiquer en langue naturelle avec un utilisateur humain et d'effectuer des tâches non critiques et répétitives.

CHAPITRE VII - BIBLIOGRAPHIE

[Agogino 04] Agogino, A.K., Tumer, K. : *Unifying temporal and structural credit assignment problems*. In the Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004, pp 980 – 987, 2004.

[Amblard 03] Amblard, F. : *Comprendre le fonctionnement de simulations sociales individus-centrées: application à des modèles de dynamiques d'opinions.* Thèse de doctorat en Informatique, Université Blaise Pascal – Clermont II, 2003.

[Aras 07] Aras, R., Dutech, A., Charpillet., F. : *Une méthode de programmation linéaire mixte pour les POMDP décentralisé à horizon fini*. Journées Françaises de Planification, Décision, Apprentissage (JFPDA'07), Grenoble, 2007.

[Arlabosse 04] Arlabosse F., Gleizes M.-P., Occello M. : *Méthodes de Conception de Systèmes Multi-agents.* In Systèmes Multi-Agents - Collection ARAGO - N° 29, 2004.

[Bernon 05] Bernon, C., Camps, V., Gleizes, M.-P., Picard, G. : *Engineering Adaptive Multi-Agent Systems: the ADELFE Methodology*. In B. Henderson-Sellers and P. Giorgini (Eds.), Agent-Oriented Methodologies. Idea Group Pub, June 2005, pp.172-202.

[Bernon 06] Bernon, C., Gleizes, M.-P., Picard, G. : *Enhancing Self-organising Emergent Systems Design with Simulation*. ESAW 2006: 284-299.

[Boissier 04] Boissier; O., Gitton, S., Glize, P. : *Caractéristiques des Systèmes et des Applications.* Systèmes Multi-Agents, Observatoire Français des Techniques Avancées, ARAGO 29, Diffusion Editions TEC & DOC, p. 25-54, 2004.

[Bourjot 01] Bourjot, C., Chevrier, V. : *De la simulation de construction collective à la détection de régions dans des images à niveaux de gris : l'inspiration des araignées sociales*. In proc. JFIADSMA'01, Hermès, pp253-263, Montréal, 2001.

[Brueckner 03] Brueckner, S., Parunak, H. V. D. : *Resource-aware exploration of the emergent dynamics of simulated systems*. AAMAS 2003: 781-788.

[Calvez 05] Calvez, B., Hutzler, G. : *Automatic tuning of agent-based models using genetic algorithms*. In: Proceedings of the 6th International Workshop MABS'05. Volume 3891 of lnai., 2005.

[Calvez 07] Calvez, B., Hutzler, G. : *Ant Colony Systems and the Calibration of Multi-Agent Simulations: a New Approach*. In : MA4CS'07 Satellite Workshop of ECCS 2007.

[Campagne 04a] Campagne, J.C., Cardon, A., Collomb, E., Nishida, T. : *Using morphology to analyse and control a Multi-Agent system, an example*. In: STAIRS ECAI'2004, August 2004.

[Campagne 04b] Campagne, J.C., Cardon, A., Collomb, E., Nishida, T. : *Massive Multi-Agent System Control*. FAABS III 2004, IEEE Workshop on Formal Approaches on Agents-based Systems, LNCS 3228, NASA Goddard Space Center, Greenbelt MA, USA, April 2004.

[Campagne 05] Campagne, J.-C. : *Systèmes multi-agents et morphologie*. Thèse de doctorat, Université Pierre et Marie Curie – Paris VI, 2005.

[Chades 02] Chadès, I., Scherrer, B., Charpillet, F. : *A Heuristic Approach for Solving Decentralized-POMDP : Assessment on the Pursuit Problem*. In ACM Symposium on Applied Computing - SAC'2002, Madrid, Spain, 2002.

[Contet 08] Contet, J.-M., Gechter, F., Gruer, P., Koukam, A. : *Evaluation of global system state thanks to local phenomenona*. ECAI 2008: 865-866, 2008.

[Culioli 94] Culioli, J. C. : *Introduction à l'optimisation*. Ellipses 1994.

[Deguet 08] Deguet, J. : *Intégration de l'émergence au sein des systèmes multi-agents. Une étude appliquée à la recherche heuristique*. Thèse de l'Université Joseph Fourier, 2008.

[Demazeau 95] Demazeau, Y. : *From interactions to collective behavior in agent-based systems*. In proceedings of the First European Conference on Cognitive Science, pp. 117-132, 1995.

[Demazeau 01] Demazeau, Y. : *Voyelles.* Rapport d'habilitation à diriger des recherches, Tech. report, Institut National Polytechnique de Grenoble, Laboratoire Leibniz, Grenoble, Avril 2001.

[DeWolf 05a] De Wolf, T., Holvoet, T. : *Emergence Versus Self-Organisation: Different Concepts but Promising When Combined*. Engineering Self Organising Systems: Methodologies and Applications (Brueckner, S. and Di Marzo Serugendo, G. and Karageorgos, A. and Nagpal, R., eds.), Lecture Notes in Computer Science, 2005, Volume 3464, May 2005, pages 1 – 15.

[DeWolf 05b] De Wolf, T., Holvoet, T. : *Towards a Methodolgy for Engineering Self-Organising Emergent Systems*, In Self-Organization and Autonomic Informatics (I), Volume 135 of Frontiers in Artificial Intelligence and Applications. H. Czap, R. Unland, C. Branki and H. Tianfield (editors), pp 18 - 34. ISBN: 1-58603-577-0, IOS Press. Proceedings of the International Conference on Self-Organization and Adaptation of Multi-agent and Grid Systems (SOAS 2005), Glasgow, Scotland, UK, 2005.

[DeWolf 05c] De Wolf, T., Samaey, G., Holvoet, T. : *Engineering Self-Organising Emergent Systems with Simulation-based Scientific Analysis*. Proceedings of the Third International Workshop on Engineering Self-Organising Applications (Brueckner, S. and Di Marzo Serugendo, G. and Hales, D. and Zambonelli, F., eds.), pp. 146-160, Utrecht, The Netherlands, 2005.

[Dorigo 92] Dorigo, M. : *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italie, 1992.

[Dreo 06] Dréo, J., Petrowski, A., Taillard, E., Siarry, P.: *Metaheuristics for Hard Optimization Methods and Case Studies*. Springer, 2006.

[Drogoul 04] Drogoul A., Ferrand N., Müller J.P. : *Emergence : L'articulation du local au global*. In : Observatoire français des techniques avancées. Systèmes multi-agents . Paris : OFTA, p. 105-136, 2004.

[Dutech 03] Dutech, A., Samuelides, M. : *Un algorithme d'apprentissage par renforcement pour les processus décisionnels de Markov partiellement observés*. RSTI - RIA. Volume 17 – n°4/2003, pages 559 à 589.

[Edmonds 04a] Edmonds, B. : *Using the Experimental Method to Produce Reliable Self-Organised Systems*. In Brueckner, S. et al. (eds.) Engineering Self Organising Sytems: Methodologies and Applications, Springer, Lecture Notes in Artificial Intelligence, 3464:84-99, 2004.

[Edmonds 04b] Edmonds, B., Bryson, J. : *The Insufficiency of Formal Design Methods - the necessity of an experimental approach for the understanding and control of complex MAS*. In Jennings, N. R. et al. (eds.) Proceedings of the 3rd International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS'04), July 19-23, New York, ACM Press, 938-945, 2004.

[Faubourg 01] Faubourg, L. : *Construction de fonctions de Lyapunov contrôlées et stabilisation non linéaire*. Thèse de doctorat, Université de Nice Sophia Antipolis, décembre 2001.

[Fehler 04] Fehler, M., Klügl, F. : *Techniques for analysis and calibration of multiagent simulations.* In: Engineering Societies in the Agents World V: Proceedings of the ESAW 2004. Number 3451 in LNCS (2005) 305–321.

[Ferber 89] Ferber J. : *Eco Problem Solving: how to solve a problem by interactions*. Proceedings of the Ninth Workshop on Distributed Artificial intelligence, Seattle, 1989.

[Ferber 95] Ferber, J. : *Les systèmes multi-agents. Vers une intelligence collective*. InterEditions, Paris, 1995.

[Ferber 96] Ferber, J., Müller, J.-P. : *Influences and Reaction: a Model of Situated Multiagent Systems.* In Second International Conference on Multi-Agent Systems, icmas'96 , Kyoto (Japon), décembre 1996.

[Ferber 06] Ferber, J. : *Introduction aux concepts et méthodologies de conception multi-agents.* In : Amblard F. Phan D. eds.(2006) Modélisation et simulation multi-agents pour les Sciences de l'Homme et de la Société : une introduction, Londres, Hermes-Sciences, 414 p. ISBN : 2-7462-1310-9. chapitre 1, p.11-36, 2006.

[Gage 92] Gage, D. W. : *Command control for many-robots systems*. Proceedings of AUVS-92, Huntsville AL, 22-24 June 1992. Reprinted in Unmanned Systems, Fall 1992, volume 10, nbr 4, pp 28-34.

[Gaud 08] Gaud, N., Galland, S., Gechter, F., Hilaire, V., Koukam, A. : *Holonic multilevel simulation of complex systems: Application to real-time pedestrians simulation in virtual urban environment*. Simulation Modelling Practice and Theory 16(10): 1659-1676, 2008.

[Gechter 05] Gechter F., Simonin O. : *Conception de systèmes multi-agents réactifs pour la résolution de problèmes : une approche basée sur l'environnement*. JFSMA 05, volume 8, 2005.

[Girgin 08] Girgin S., Preux P. : *Feature Discovery in Reinforcement Learning using Genetic Programming*. Eleventh European Conference on Genetic Programming (EuroGP), 2008.

[Gleizes 04] Gleizes, M.-P. : *Vers la résolution de problèmes par émergence*. Habilitation à Diriger des Recherches, Université Paul Sabatier - Toulouse III, 2004.

[Goldman 2004] Goldman, C.V., Zilberstein, S. : *Decentralized Control of Cooperative Systems: Categorization and Complexity Analysis*. Journal of Artificial Intelligence Research, 22:143-174, 2004.

[Goldstein 99] Goldstein, J. : *Emergence as a construct : History and issues*. In Emergence, volume 1, pp. 49-71, 1999.

[Halloy 07] Halloy, J., Sempo, G., Caprari, G., Rivault, C., Asadpour, M., Tâche, F., Saïd, I., Durier, V., Canonge, S., Amé, J-M., Detrain, C., Correll, N., Martinoli, A., Mondada, F., Siegwart, R., Deneubourg J-L. : *Social integration of robots into groups of cockroaches to control self-organized choices*. *Science*, 318, 1155-1158, 2007.

[Handl 04] Handl, J., Knowles, J. : *Multiobjective clustering with automatic determination of the number of clusters*. In: Technical Report TR-COMPSYSBIO-2004-02. UMIST, Manchester, 2004.

[Hassas 03] Hassas, S. : *Systèmes Complexes à base de Multi-Agents Situés*. Mémoire pour l'habilitation à diriger les recherches, Université Claude Bernard – Lyon 1, 2003.

[Helbing & Molnar 95] Helbing, D., Molnàr, P. : *Social force model for pedestrian dynamics*. Physical Review E, 51: 4282-4286, 1995.

[Helbing 00] Helbing, D., Farkas, I.J., Vicsek, T. : *Freezing by heating in a driven mesoscopic system*. Physical Review Letters, 84: 1240-1243, 2000.

[Jain 99] Jain, A.K., Murty, M.N., Flynn, P.J. : *Data Clustering: A Review*. ACM Computer Survey 31(3): 264-323, 1999.

[Jean 97] M.R. Jean (Nom collectif) : *Emergence et SMA*. Journées Francophones IAD et SMA, Nice, Hermès, 1997.

[Khalil 02] Khalil, H. K. : *Nonlinear Systems*. Third Edition, Prentice Hall, ISBN 0-13-067389-7, 2002.

[Klein 05] Klein, F., Bourjot, C., Chevrier, V. : *Dynamic design of experiment with MAS to approximate the behavior of complex systems*. In Multi-Agents for modeling Complex Systems (MA4CS'05) Satellite Workshop of the European Conference on Complex Systems (ECCS'05), 2005.

[Klein 06] Klein, F., Bourjot, C., Chevrier, V. : *Approche expérimentale pour la compréhension des systèmes multi-agents réactifs*. JFSMA06, Annecy, 2006.

[Klein 08] Klein, F., Bourjot, C., Chevrier, V. : *Controlling the Global Behaviour of a Reactive MAS : Reinforcement Learning Tools*. In the 9th Annual International Workshop "Engineering Societies in the Agents World" - ESAW 08.

[Klein 09] Klein, F., Bourjot, C., Chevrier, V. : *Contribution to the Control of a MAS's Global Behaviour: Reinforcement Learning Tools*. In LNAI09 : Engineering Societies in the Agents World IX - 9th International Workshop, ESAW 2008, Revised Selected Papers, Springer-Verlag Berlin Heidelberg (Ed.), 2009.

[Lacroix 06] Lacroix, B., Mathieu, P., Picault, S. : *Time and Space Management in Crowd Simulations*. Proceedings of the European Simulation and Modelling Conference (ESM'06), pp. 315-320, Toulouse, France, 2006.

[Lee 04] Lee, C.F., Wolpert, D.H. : *Product distribution theory for control of multi-agent systems*. In Proceedings of the Third International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS 2004), IEEE Press, New York, USA, p. 522-529, 2004.

[Lehalle 05] Lehalle, C.-A. : *Contrôle non linéaire et réseaux de neurones formels : les perceptrons affines par morceaux*. Thèse de doctorat, Université Paris 6 Pierre et Marie Curie, juin 2005.

[Lévy-Leblond 96] Lévy-Leblond, J.-M. : *Aux Contraires: L'exercice de la pensée et la pratique de la science*. Editions Gallimard,1996.

[Manneville 00] Manneville, P. : *Dynamique non linéaire appliquée au chaos et à son contrôle*. Lecture notes/notes de cours, DEA de Dynamique des Fluides et des Transferts (Paris-Sud) et DEA de Mécanique (Paris VI), 2000-2001.

[Narzisi 06] Narzisi, G., Mysore, V., Bud Mishra, B. : *Multi-objective evolutionary optimization of agent-based models: An application to emergency response planning*. In Kovalerchuk, B., ed.: The IASTED International Conference on Computational Intelligence, 2006.

[Parunak 99] Parunak, H. V. D. : *Synthetic Ecosystems: A Perspective for Multi-Agent Systems*. Tutorial PAAM 1999.

[Picard 06] Picard, G., Glize, P. : *Model and Analysis of Local Decision Based on Cooperative Self-Organization for Problem Solving*. In : Multiagent and Grid Systems, IOS Press, Vol. 2(3), 253-265, septembre/September 2006.

[Prouvost 04] Prouvost, P. : *Automatique Contrôle et régulation*. Dunod, 2004.

[Reynolds 87] Reynolds, C.W. : *Flocks, Herds and Schools : a distributed behavioral model.* Computer Graphics, vol.21, n°4, pp.289-296, 1987.

[Ricordel 01] Ricordel, P.-M. : *Programmation orientée multi-agents : développement et déploiement de systèmes multi-agents voyelles*. Thèse de Doctorat de l'INPG, 2001.

[Rondepierre 06] Rondepierre, A. : *Algorithmes hybrides pour le contrôle optimal des systèmes non-linéaires.* thèse de doctorat, INPG, 2006.

[Sauter 01] Sauter, J.A., Parunak, H.V.D., Brueckner, S., Matthews, R. : *Tuning Synthetic Pheromones Withe Evolutionary Computing*. In: Genetic and Evolutionary Computation Conference Workshop Program (GECCO 2001), San Fransisco, CA, 2001.

[Scherrer 04] Scherrer, B. : *Approche connexionniste du contrôle optimal*. In JEDAI, Journal électronique d'intelligence artificielle*.* Volume 4, Août 2004.

[Shalizi 01] Shalizi, C. : *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*. PhD thesis, University of Wisconsin, Madison, USA, 2001.

[Sichman 95] Sichman, J. S. : *Du raisonnement social chez les agents : une approche fondée sur la théorie de la dépendance*. Thèse de doctorat, Institut National Polytechnique de Grenoble, 1995.

[Siebert 08] Siebert, J., Ciarletta, L., Chevrier, V. : *Impact du comportement des utilisateurs dans les réseaux pair-à-pair (P2P) : modélisation et simulation multi-agents*. In proc JFSMA08, Brest, pp129-138, Cépaduès, 2008.

[Sierra 02] Sierra, C., Sabater, J., Agusti, J., Garcia, P. : *Evolutionary Computation in MAS Design*. In: Proceedings ECAI, pp188-192, 2002.

[Sigaud 08] Sigaud, O., Garcia, F. : *Processus décisionnels de Markov en intelligence artificielle (volume 1)*, chapter Apprentissage par renforcement, pages 53-88. Hermes-Lavoisier. 2008.

[Simonin 02] Simonin O., Michel J., Chapelle J., Ferber J. : *Un simulateur de systèmes multi-robots dans MadKit*. In Xèmes journées Francophones pour l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents, JFIADSMA'02, (Lille France), 28-30 Octobre 2002.

[Sutton & Barto 98] Sutton, R., Barto, A. : *Reinforcement Learning : an introduction*. MIT Press, Cambridge, 1998.

[Szer 06] Szer, D., Charpillet, F. : *Point-based Dynamic Programming for DEC-POMDPs*. In Proceedings of the Twenty-First AAAI National Conference on Artificial Intelligence – AAAI'2006.

[Thomas 02] Thomas, V., Bourjot, C., Chevrier, V., Desor, D. : *MAS and RATS : Multi-agents simulation of social differenciation in rats groups. Interest for the understanding of a complex biological phenomenon*. International Workshop on Self-Organization and Evolution of Social Behaviour September 8-13, 2002.

[Vercouter 01] Vercouter, L. : *Une Gestion Distribuée de l'Ouverture dans un Système Multi-Agent*. Journées Francophones d'Intelligence Articifielle Distribuée et des Systèmes Multi-Agents (JFIADSMA'01), Hermes, Montréal, Canada, Novembre 2001.

[Vercouter 04] Vercouter, L. : *MAST : Un modèle de composants pour la conception de SMA*. 1ère Journée Multi-Agents et Composants (JMAC'04), Paris, France, 23 Novembre 2004.

[Weyns 04] Weyns, D., Parunak, H. V. D., Michel, F., Holvoet, T., Ferber, J. : *Environments for Multiagent Systems State-of-the-Art and Research Challenges*. E4MAS 2004: 1-47, 2004.

[Wooldridge 01] Wooldridge, M., Ciancarini, P. : *Agent-oriented software engineering : the state of the art*. In P. Ciancarini and M. Wooldridge, editors, *Agent-Oriented Software Engineering*. Springer-Verlag Lecture Notes in AI Volume 1957, January 2001.

[Wooldridge 02] Wooldridge, M. : *An Introduction to Multiagent Systems*. John Wiley and Sons Ltd, February 2002.

CHAPITRE VIII - ANNEXES

A. Choix de l'ensemble $\mathcal{A}$ des moyens d'action

L'objectif est ici d'étudier une variation d'un problème de contrôle. L'idée est de simplifier ce problème, en sélectionnant un sous-problème plus simple, afin de réaliser l'apprentissage plus rapidement.

Pour illustrer l'influence de l'ensemble des moyens d'action sur le contrôle, nous considérons un nouveau problème, plus difficile que les précédents, c'est-à-dire pour lequel une politique aléatoire offre des résultats notablement moins bons. Cela nous permet d'observer des différences significatives de performances entre l'utilisation de deux ensembles d'actions différents. En effet, les bons résultats obtenus sur un problème plus facile avec peu d'actions risquent de ne pas être améliorés notablement en ajoutant d'autres actions de contrôle.

Expérience 17: variation de l'ensemble d'actions du MDP

Problème

Cible du contrôle : 1 bloc et 2 lignes

Actions de contrôle : sur la largeur, les facteurs d'évitement, de mouvement et de séparation

Mise en oeuvre

États de contrôle $\mathcal{S}$: 12 états, nombre de blocs$\in${0, 1, 2 ou plus} et nombre de lignes$\in${0, 1, 2, 3 ou plus}

Ensemble $\mathcal{A}$ des actions du MDP : trois solutions testées

1. toutes conjugaisons, 2100 actions (35 × 3 × 4 × 5)
2. actions sur le comportement conjuguées, mais pas avec celles sur l'environnement, 95 actions (35 + 3 × 4 × 5).

Apprentissage de la politique en respectivement 8500 et 6000 simulations, politique proportionnelle aux Q(s,a).

Résultats

Ensemble d'actions $\mathcal{A}$	π	v	τ	γ
1	58	22	13000	280
2	88	15	6000	115

Tableau 25: Comparaison de deux ensembles d'actions

Dans l'expérience 17, nous souhaitons savoir s'il est intéressant pour atteindre la cible [un bloc, deux lignes] de conjuguer deux ensembles d'actions :

- Le premier est celui que nous avons appelé l'ensemble n°1, qui influe sur la largeur de l'environnement, et qui contient 35 moyens d'action.
- Le second correspond au choix des facteurs d'évitement, de mouvement et de séparation des agents. Les valeurs de ces paramètres sont choisis respectivement dans les ensembles {10, 15, 20} (équivalent à l'ensemble n°5), {2, 3, 4, 5} (ensemble n°6), et {20, 25, 30, 35, 40}. Un action de cet ensemble revient à fixer une valeur pour chacun de ces trois paramètres, il y en a 60 en tout.

Si ces deux ensembles sont conjugués, une action de contrôle revient à choisir simultanément un moyen d'action de chaque ensemble, et l'ensemble $\mathcal{A}$ du problème contient $35 \times 60 = 2100$ moyens d'action. Mais nous pouvons choisir d'influer soit sur le premier ensemble, soit sur le second. Il ne reste alors que $35 + 60 = 95$ moyens d'action à explorer.

L'apprentissage avec 2100 actions est arrêté par excès de temps, après 8500 simulations et 290000 pas de simulation en 24 heures. 6000 simulations suffisent pour apprendre la politique avec le second ensemble actions.

On voit que le contrôle avec l'ensemble d'actions simplifié (2) donne de biens meilleurs résultats qu'avec l'ensemble complet. L'apprentissage est trop compliqué pour l'ensemble 1, l'exploration de toutes les actions trop longue, donc incomplète si une limite de temps est imposée.

En conclusion, les performances de l'approche proposée dépendent du problème étudié, et sont limitées par la durée de l'apprentissage. Une perspective est d'étudier la réduction de l'ensemble d'actions afin d'améliorer le contrôle. C'est un problème difficile, qui peut faire l'objet d'une étude à part entière.

Ce n'est en outre que l'un des aspects du choix de l'ensemble d'actions. Par exemple, si le problème de contrôle autorise de diriger comme on le souhaite un paramètre, il faut décider du découpage de l'intervalle des valeurs possibles de ce paramètre en moyens d'action.

B. Contrôle avec des informations locales

La décentralisation du SMA limite l'observation qu'on peut en faire. S'il n'est pas possible de mesurer le comportement global du SMA, on peut se demander si un contrôle local possède de bonnes performances de contrôle. Il s'agit de contrôler chaque agent indépendamment d'informations globales sur le SMA. Nous avons vu qu'une telle approche pose un problème de complexité. Nous choisissons d'essayer un contrôle au niveau local avec des informations en petit nombre.

Pour cela, nous considérons un contrôle identique pour tous les agents. Ils ne possèdent que deux états locaux, selon qu'il appartiennent à un bloc ou à une ligne d'agents. Dans l'expérience 18, les agents choisissent un action en fonction de cette seule information, afin de résoudre les problèmes p_2 et p_3.

Expérience 18: utilisation d'états locaux pour les problèmes p_2 et p_3 avec des contraintes de décentralisation

Problèmes p_2 et p_3

Mise en oeuvre

Apprentissage et application de la politique en ne prenant en compte que des états locaux : l'appartenance des agents à un bloc ou à une ligne.

Résultats

Problème	***p_2***	***p_3***
Contrôle décentralisé (évalué)	80	59
Contrôle centralisé (proposition de base)	94	67

Tableau 26: Taux de convergence π pour un contrôle décentralisé, et comparaison à l'application de la proposition.

L'apprentissage se fait en suivant la méthode décrite dans [Chades 02]. On fixe l'action à réaliser lorsque l'agent se trouve dans une ligne, et on apprend la meilleure action à faire lorsqu'il est bloqué. Puis cette action est fixée et les agents apprennent quelle action effectuer lorsqu'ils appartiennent à une ligne. Ces deux opérations sont répétées plusieurs fois (3 dans notre expérience) afin d'atteindre un équilibre.

Le taux de convergence π obtenu pour les problèmes p_2 et p_3 avec cette approche n'atteint pas les performances de la proposition. Le contrôle au niveau global semble donc plus intéressant pour ces problèmes.

Une solution à approfondir, pour compenser l'observation partielle du SMA, consisterait à *estimer* les états globaux à partir d'informations locales partielles au lieu de le *mesurer* grâce à toutes les informations locales nécessaires. Il serait alors possible d'appliquer, de manière approchée et incertaine, un contrôle au niveau global malgré le manque d'informations. Le cadre des POMDP pourrait s'avérer intéressant pour cette approche.