INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Formalisme pour la construction automatique d'interactions dans les SMA réactifs - version étendue

Vincent Thomas — Christine Bourjot — Vincent Chevrier

N° ????

Mai 2005

Thème NUM

Rapport de recherche

ISSN 0249-6399

Formalisme pour la construction automatique d'interactions dans les SMA réactifs - version étendue

Vincent Thomas , Christine Bourjot , Vincent Chevrier

Thème NUM — Systèmes numériques
Projet Maia

Rapport de recherche n° ???? — Mai 2005 — 15 pages

Résumé : Nous proposons un nouveau formalisme de représentation des actions et des interactions dans les SMA réactifs inspiré des processus de décision Markoviens décentralisés (DEC-MDP). Ce formalisme appelé Interac-DEC-MDP permet de représenter dans un même cadre homogène les actions individuelles et les interactions directes entre agents. Ainsi il permet de calculer automatiquement les prises de décisions des agents relativement aux actions et aux déclenchements et résolution des interactions par l'introduction de la rationalité au travers de la notion de récompense. Un premier problème simple de partage de ressources impliquant 2 agents a été modélisé selon le formalisme proposé et les comportements des agents ont été construits automatiquement par Q-learning. Les premiers résultats bien qu'obtenus avec des hypothèses limitatives montrent qu'il est possible à partir d'apprentissages simples de construire automatiquement des comportements collectifs pertinents.

Mots-clés : Interaction, Apprentissage, SMA réactifs, Processus Décisionnels de Markov

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

A formalism for automatic computation of interactions in reactive multi-agent systems - extended version

Abstract: In this paper, we propose an original formalism inspired by Markov Decision Process in order to represent homogeneously actions and direct interactions between agents. Thanks to the introduction of individual rewards characterising the problem to be solved, this formalism is a first step towards the automatic computation of policies and use of available interactions. A simple problem of distributing resources among 2 agents has been modelled and solved by a Q-learning based approach. First results show that it is possible to benefit from simple individual learnings to produce efficient collective behaviour.

Key-words: Interaction, Learning, reactive Multi-agent system, Markov Decision process

1 Introduction

Le contexte de notre travail est celui des systèmes multi-agents réactifs. Les agents qui nous intéressent ne disposent pas de mémoire, de représentation globale du système, de la tâche à effectuer et des autres agents. Leurs prises de décision sont simples, de type stimulus réponse et se fondent uniquement sur des perceptions locales. Dans ce cadre, nous nous focalisons sur la résolution collective de problèmes. Il s'agit de définir les comportements individuels et les interactions du système pour que celui-ci se comporte comme un tout cohérent et parvienne à résoudre à l'exécution et de manière décentralisée le problème posé à la collectivité. La résolution de la tâche est alors le produit des couplages des comportements des agents.

Ces couplages peuvent être la conséquence d'interactions médiées par l'environnement comme dans les approches stigmergiques [LF94] [BCT03] [DDCG99]. L'environnement est modifié par les actions effectuées par les agents et ces modifications ont en retour une influence sur les comportements des autres agents. De la même manière la position d'un agent dans l'environnement peut être perçue par les autres agents et influencer leurs comportements [Rey87] [GCC03]. Ces couplages peuvent aussi être la conséquence d'interactions ponctuelles, non médiées par l'environnement, lorsqu'un agent essaie d'influencer directement le comportement d'un autre agent [SF00] [BDFJ91].

Le problème consistant à déterminer les comportements et les interactions locales du système est d'autant plus difficile que, généralement, les comportements et les interactions sont décrits à un niveau d'abstraction local, celui de l'agent, différent de celui dans lequel la résolution collective est constatée. De plus, la résolution de la tâche apparaît à l'exécution du système du fait des couplages et ne peut se déduire directement des composants élémentaires.

Une stratégie possible est de s'inspirer de phénomènes observés dans la nature pour tenter de tirer parti de leurs propriétés. De telles approches sont parvenues à produire des résultats probants sur des problèmes complexes comme le modèle ant colony optimisation [DDCG99].

Une autre approche, celle abordée dans cet article, consiste à construire un formalisme mathématique pour caractériser un problème à partir d'un critère de qualité globale et à développer des algorithmes permettant de construire automatiquement le comportement des agents du système. Les modèles markoviens et leurs extensions [BZI00] en introduisant de la rationalité au niveau individuel permettent à des agents d'apprendre automatiquement les comportements à adopter pour résoudre une tâche donnée [Buf03]. Cependant, pour le moment, ces modèles ne représentent pas de manière explicite les interactions entre les composants du système. Ceci nécessite de développer des algorithmes complexes ou d'introduire de nouveaux mécanismes pour prendre en compte la multiplicité des prises de décision comme des communications [PT02].

Dans cet article, nous proposons un formalisme intitulé Interac-DEC-MDP. Ce formalisme constitue une extension aux modèles markoviens décentralisés (DEC-MDP) [BZI00] dans lequel la notion d'interaction directe entre agents a été ajoutée. Il permet ainsi de représenter des systèmes dans lesquels les capacités pour résoudre la tâche collective et les prises de décisions sont décentralisées comme les DEC-MDP. Mais, en outre, il est possible de représenter explicitement des interactions directes entre les agents. Les agents peuvent alors mettre à jour leurs comportements vis à vis de celles-ci et prendre des décisions collectives. Ainsi, les interactions introduites peuvent permettre de résoudre des conflits d'intérêt entre les agents au bénéfice de l'avancement de la tâche collective en cours. Ces soucis sont relativement proches de ceux d'autres approches centrées sur l'interaction comme [SF00] et [GGG03].

Cet article se décompose en cinq parties. Tout d'abord, les modèles markoviens seront présentés de manière formelle et nous spécifierons notre cadre de travail. Dans la seconde partie, nous présenterons le formalisme Interac-DEC-MDP en développant la manière dont les instances d'interaction peuvent être définies. La partie suivante proposera un mécanisme très simple fondé sur un Q-learning permettant un premier apprentissage tirant parti du formalisme. Nous présenterons enfin un exemple simple d'utilisation des Interac-DEC-MDP et discuterons du formalisme.

2 Cadre Markovien

2.1 DEC-MDP

Le formalisme DEC-MDP est défini par $< S, A, T, R, J, O, n >$ avec

- n le nombre d'agents du système
- S un ensemble fini d'états.
- $A = \cup_{i<n} A_i$ un ensemble fini d'actions jointes. A_i désigne l'ensemble des actions possibles pour l'agent i.
- $T : S \times A \to P(S)$ la fonction de transition du système. $T(s, a)$ fournit la densité de probabilité de l'état d'arrivée suite à une action jointe a dans l'état s.
- $R : S \times A \times S \to P(\Re)$ la fonction de récompense globale du système.
- J_i des ensembles finis d'observations.
- $O_i : S \times A \times S \to P(J_i)$ la fonction d'observation de chaque agent. $O(s, a, s')$ fournit une densité de probabilité sur les observations. L'ensemble des observations des agents permet de retrouver l'état s du système.

L'originalité du formalisme DEC-MDP est de pouvoir représenter des systèmes pour lesquels la prise de décision est décentralisée et par conséquent des systèmes multi-agents. Chaque agent est caractérisé par un comportement réactif stochastique sans mémoire π_i : $J_i \to A_i$. Résoudre un DEC-MDP consiste à déterminer les politiques individuelles réactives π_i des agents pour maximiser au cours de l'exécution décentralisée du système la somme des récompenses pondérées $\sum_t \gamma^t . R_t$. Ce problème reste ouvert et le formalisme ne stipule rien quant aux contraintes concernant le calcul de ces politiques.

2.2 Nos contraintes sur le calcul des politiques

Nous cherchons à construire de manière **décentralisée** les comportements des agents du système (politiques π_i) à l'exécution (par apprentissage). Dans un premier temps, nous considérons que les agents ont une perception globale du système. Nous souhaitons en outre que chaque agent n'ait accès qu'à une évaluation partielle de l'avancement de la tâche globale. Ainsi, nous supposerons par la suite que la fonction de récompense globale du système R peut se décomposer en récompenses locales additives ($R = \sum_i r_i$). Chaque agent i n'accède qu'à r_i. Cette décomposition permet de faire le lien entre le niveau global (résolution de la tâche caractérisée par R) et le niveau local de l'agent. Cependant de nouveaux problèmes apparaissent dans des situations analogues à la "tragédie des communs" [Har68] pour lesquelles maximiser la récompense individuelle peut conduire à la ruine du système.

Nous nous intéressons donc à un problème de **coopération** (à savoir que l'objectif est de maximiser une récompense globale) dans un système **perçu globalement** par chaque agent, pour lequel les récompenses sont **partiellement observées** et dont la résolution se fera de manière **décentralisée** à l'exécution. Afin de résoudre ce problème en introduisant des considérations collectives, nous avons ajouté dans le cadre présenté précédemment un nouveau concept : celui d'interactions directes entre agents.

3 Interac-DEC-MDP

Cette partie décrit le formalisme Interac-DEC-MDP qui permet de prendre en compte la notion d'interaction directe. Une interaction directe peut être définie comme une "influence mutuelle réciproque". Dans notre cas, nous utiliserons des interactions directes réactives. Nous définissons dans notre cadre ce concept comme :
"des envois de signaux impliquant deux agents et conduisant à une prise de décision collective suivie de l'émission d'une action jointe coordonnée faisant évoluer l'état du système". Nous détaillons ensuite la structure du module d'interaction permettant d'introduire différentes instances d'interactions dans le système, puis nous décrivons formellement l'exécution du système.

3.1 Le formalisme Interac-DEC-MDP

Un Interac-DEC-MDP est constitué de deux modules :

- un premier module est appelé "module d'action". Il est défini par $< S, A, T, R, n >$ et correspond au formalisme présenté dans la partie précédente.
- un second module est un "module d'interaction". Il permet aux agents d'effectuer des interactions directes avec d'autres agents. Un Interac-DEC-MDP sans aucune interaction correspond au modèle présenté en première partie.

Les interactions que nous considérerons ont plusieurs composantes :

- un agent émetteur : l'agent à l'origine de l'interaction
- un agent receveur : l'agent vers lequel l'interaction est dirigée
- un moyen : des échanges de signaux numériques entre agents
- une influence réciproque correspondant à une prise de décision collective
- un effet : une évolution de l'état du système dûe à une action jointe coordonnée choisie par les deux agents impliqués

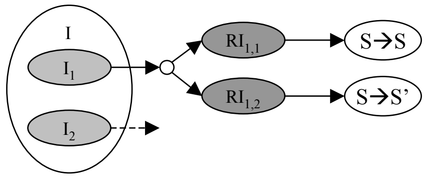

FIG. 1 – Structure du module d'interaction : a) les interactions b) les résultats possibles c) les matrices de transitions associées

L'intérêt d'une telle interaction réside dans la prise de décision collective permettant à deux agents de prendre en compte des considérations plus globales alors qu'ils n'ont accès qu'à des récompenses individuelles.

3.2 Structure du module d'interaction

Le module d'interaction admet plusieurs instances d'interactions (cf Figure 1), les interactions considérées ne durent qu'un cycle. I_k désigne l'instance d'interaction k et $I = \{I_k\}_k$ l'ensemble des interactions directes possibles dans ce système. Pour une interaction I_k, il existe plusieurs résultats possibles $RI_{k,l}$ (RI_k en désigne l'ensemble), chaque résultat correspond à une action jointe coordonnée et y est associée une matrice de transition $T_{RI_{k,l}} : S \times [0..n] \times [0..n] \to P(S)$ faisant évoluer le système en fonction des indices des agents impliqués dans l'interaction.

3.3 Exécution du module d'interaction

L'exécution du module d'interaction se décompose en une séquence de plusieurs cycles. Chaque cycle consiste à se concentrer sur un des agents du système qui sera l'agent émetteur d'interaction pour ce cycle. Une exécution du module d'interaction permettra à chaque agent d'être l'agent émetteur d'interaction.

Chaque cycle d'interaction peut se décomposer en plusieurs étapes :

Une étape de déclenchement consistant pour l'agent émetteur à choisir l'interaction qu'il va déclencher et quel agent constituera l'agent receveur. Le déclenchement d'une interaction est décidé de manière autonome par un agent (cf Figure 2.a). Cette prise de décision est représentée par la politique $\pi_{trig,i} : S \to P(I, [0..n])$. Celle-ci fournit pour un état donné l'interaction I_i déclenchée et l'indice $n_i \in [0..n]$ de l'agent receveur. $(I_i, n_i) \leftarrow \pi_{trig,i}(s)$

Une étape de résolution des interactions consistant en une prise de décision collective effectuée par les deux agents impliqués dans l'interaction. Cette seconde étape consiste à décider du résultat de l'interaction I_i qui a été déclenchée (cf Figure 2.b). Il s'agit d'une prise de décision collective impliquant l'agent émetteur et l'agent receveur de l'interaction. Cette prise de décision collective est représentée par une politique $\Pi_{I_k} : S \times [0..n] \times [0..n] \to P(RI_k)$.

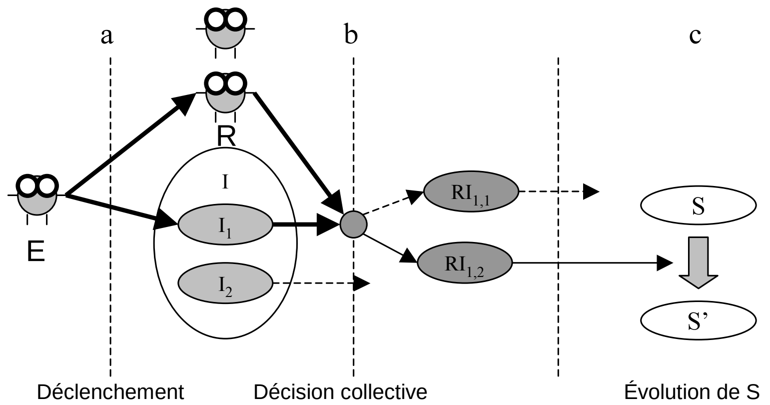

FIG. 2 – L'exécution d'un Interac-DEC-MDP se décompose en trois phases : a) une phase de déclenchement de l'interaction (E désigne l'agent émetteur, R le receveur), b) une phase de résolution et c) une phase d'évolution

Cette dernière est calculée en fonction des deux agents impliqués et retourne une distribution sur les résultats possibles RI_k de l'interaction considérée I_k : $RI_{i,l} \leftarrow \Pi_{I_i}(s, i, n_i)$

Une transition faisant évoluer l'état du système. Enfin, le résultat d'interaction choisi correspond à l'exécution d'une action jointe coordonnée. Cette dernière conduit à une évolution de l'état courant du système en fonction de la matrice de transition associée au résultat d'interaction choisi (cf Figure 2.c) $s \leftarrow TRI_{RI_{i,l}}(s, i, n_i)$

3.4 Exécution de interac-DEC-MDP

Le fonctionnement général d'un interac-DEC-MDP est décrit par l'algorithme 1.

3.5 Bilan

Les Interac-DEC-MDP permettent ainsi de représenter des actions individuelles et des interactions directes dans un même formalisme. Ils permettent de définir une nouvelle classe de problèmes dans laquelle les agents peuvent interagir directement. Résoudre une instanciation, consiste à trouver l'ensemble des politiques individuelles π_i pour tout i, l'ensemble des politiques individuelles de déclenchement $\pi_{trig,i}$ pour tout i, l'ensemble des politiques de résolution d'interaction Π_{i,j,I_k} pour tout i, j, k.

Le formalisme n'impose pas de contraintes sur les politiques à introduire, il peut s'agir de politiques données a priori dans une approche plutôt ascendante (le formalisme constitue

Algorithme 1 Execution d'un Interac-DEC-MDP

Entrées: n agents :

```
tant que (execution) faire
    (*Modèle d'action*)
    pour i from 1 to n faire
        a_i = π_i(s)
    fin pour
    s = T(s, {a_i}_i)
    (*Modèle d'interaction*)
    pour i from 1 to n faire
        (I_k, n_i) = π_{trig,i}(s)
        RI_{k,l} = Π_{I_k}(s, i, n_i)
        s = TRI_{k,l}(s)
    fin pour
fin tant que
```

alors une plate-forme d'évaluation des politiques données) ou construites automatiquement en fonction d'un objectif comme dans une approche descendante. Dans la partie suivante, nous proposons une première méthode basée sur l'apprentissage par renforcement pour tirer parti des interactions introduites dans le système.

4 Processus de résolution

4.1 Q-learning

Pour un Interac-DEC-MDP, nous cherchons à construire les comportements individuels (π, π_{trig}) et collectifs (Π) permettant de maximiser la récompense collective reçue par le système. Des approches comme [Buf03] ont montré qu'il était possible d'apprendre dans un cadre décentralisé à coordonner des comportements de manière implicite par Q-learning [SB98]. L'avantage d'une telle approche est de ne pas avoir à représenter au sein d'un agent les comportements des autres agents. La partie suivante présente pour cela les représentations que nous allons manipuler.

4.2 Représentations internes des politiques

Politiques individuelles Les politiques π_i peuvent être représentées de la même manière que dans un Q-learning classique. Chaque agent dispose d'une table de $Q-valeurs$: $Q_i : S \times A_i \rightarrow \Re$. La $Q-valeur$ $Q_i(s, a_i)$ représente l'espérance de gain de l'agent i lorsqu'il se trouve dans l'état s et effectue l'action a_i. La politique individuelle de l'agent i peut alors être calculée en fonction de ses $Q-valeurs$. Par exemple, une politique gloutonne sera définie par : $\pi_i(s) = argmax_{a_i}(Q_i(s,a_i))$. Bien entendu, cette espérance peut être biaisée du fait de la présence de plusieurs agents dans l'environnement.

Politiques de déclenchement L'étape de déclenchement d'interaction fait appel à une prise de décision autonome, ces politiques peuvent donc être représentées de manière analogue à l'aide de $Q-valeurs$ individuelles de déclenchement : $Q_{trig,i} : S \times I \times [0..n] \rightarrow \Re$. Une politique gloutonne consistera à choisir i_i et n_i maximisant ces $Q - valeurs$.

Politiques de résolution Les politiques de résolution d'interaction sont des politiques collectives impliquant deux agents. Dans un système composé de n agents, il faut donc $n*(n-1)$ politiques de résolution $\Pi_{i,j}$ (une politique par couple d'agents). Nous sommes partis sur l'hypothèse qu'un simple échange des $Q-valeurs$ des agents impliqués permettait de prendre une décision collective efficace. Ainsi, les politiques de résolution vont être représentées de manière distribuée au sein des agents. Chaque agent dispose de $Q - valeurs$ d'interaction $Q_{I_k,i} : S \times RI_{I_k} \times \{R,E\} \rightarrow \Re$ et à chaque fois que deux agents interagissent ensemble, la politique d'interaction correspondante est reconstruite à partir des $Q-valeurs$ d'interaction de ces deux agents. Ces $Q - valeurs$ d'interactions dépendent du rôle de l'agent vis à vis de l'interaction (émetteur représenté par E, ou receveur, représenté par R).

Au cours d'une interaction entre un agent émetteur a et un agent receveur b, on reconstruit des $Q-valeurs$ associées à l'interaction $Q_I(RI_{k,l}) = Q_{I,a}(s, RI_{k,l}, E) + Q_{I,b}(s, RI_{I_k}, R)$, basées sur une évaluation collective de l'utilité associée au groupe $\{a,b\}$

Une politique gloutonne consiste alors à choisir le résultat d'interaction $\Pi_{a,b,I_k} = argmax_{RI_{k,l}}(Q_I(RI_{k,l}))$. Ces $Q - valeurs$ Q_I peuvent alors être comprises comme les influences exercées par chacun des agents pour que le résultat de l'interaction soit le résultat RI considéré. Le fait de maximiser la somme des $Q - valeurs$ d'interaction introduit des considérations collectives et peut conduire un agent à émettre un comportement altruiste pour améliorer les performances globales du système (la récompense globale reçue) au détriment de la sienne.

4.3 Apprentissages dépendants

Une première approche rudimentaire a été conduite pour concevoir automatiquement les différentes politiques nécessaires à l'exécution du système : il s'agit d'effectuer trois apprentissages successifs.

Apprentissage des politiques individuelles Dans un premier temps, tous les agents apprennent simultanément et indépendamment leurs politiques individuelles π_i en fonction des récompenses individuelles r_i reçues, sans qu'aucune interaction ne soit émise. Comme l'environnement d'un agent contient les autres agents du système en train d'apprendre en parallèle, cet environnement n'est plus stationnaire et il n'existe aucune garantie de convergence [SPG03].

A chaque pas de temps, le système évolue en fonction de l'action jointe A émise par les agents. Chaque agent effectue l'action a_i part de l'état s et arrive dans l'état s' suite à l'action jointe A. Il reçoit la récompense individuelle r_i et peut ainsi mettre à jour ses $Q - valeurs$: $Q_i(s,a_i) \leftarrow (1-\alpha).Q_i(s,a_i) + \alpha.(\gamma.max_{a'}(Q_i(s',a')))$

Apprentissage des politiques jointes $\Pi_{i,j}$ Les politiques individuelles π_i sont ensuite figées, le déclenchement des interactions est aléatoire et les agents apprennent à résoudre conjointement les interactions déclenchées.

A l'issue de chaque interaction, un résultat d'interaction est choisi. Ce résultat RI_i est connu par les deux agents impliqués dans l'interaction et fait évoluer le système de l'état s vers l'état s'. Chaque agent sait alors quelle est sa récompense escomptée (lorsque les interactions ne sont pas considérées) en fonction de l'état s' dans lequel il se trouve : $max_a(Q(s',a))$ appris dans la phase précédente. Les agents peuvent donc apprendre l'utilité escomptée en fonction des résultats d'interaction choisis et mettre à jour leurs $Q - valeurs$ d'interactions Q_I à l'aide des $Q - valeurs$ précédentes. $Q_{I_k,i}(s, RI_{k,l}, E) \leftarrow (1-\alpha).Q_{I_k,i}(s, RI_{k,l}, E) + \alpha.(r_i + \gamma.max_{a'}(Q_i(s',a'))$.

Apprentissage des déclenchements Les politiques individuelles π_i et les politiques collectives $\Pi_{i,j}$ sont figées et les agents apprennent à déclencher les interactions. Quand un agent déclenche une interaction dans un état s, le système évolue vers un état s' en fonction des politiques $\Pi_{i,j}$ qui ont été apprises (et donc des Q_I). Chaque agent peut donc savoir a posteriori la récompense escomptée lorsqu'il déclenche une certaine interaction dans un certain état et mettre à jour de cette manière ses Q-valeurs de déclenchement : $Q_{trig,i}(s, I, n) = (1-\alpha).Q_{trig,i}(s, I, n) + \alpha.(\gamma.max_{a'}(Q_i(s,a')))$

4.4 Bilan

Dans cette partie, nous avons présenté un processus de résolution consistant à apprendre les politiques π, π_{trig} et Π.

Cet apprentissage est un apprentissage relativement simple puisqu'il ne nécessite pas de modéliser le comportement des autres agents et réutilise des apprentissages individuels décentralisés. Cependant, pour le moment cet apprentissage ne remet pas en cause les politiques individuelles π_i qui ont été apprises en absence d'interactions dans le système. Il consiste simplement à tirer parti de politiques individuelles déjà apprises par le passé pour améliorer le comportement collectif du système.

5 Exemple

Afin de tester le formalisme et le processus de résolution, nous avons modélisé un problème simple impliquant deux agents. Ceux-ci sont situés dans une cage et peuvent accéder à la nourriture en traversant un couloir immergé. Chaque agent a besoin de se nourrir mais traverser l'élément liquide coûte à chacun d'entre eux (le coût peut dépendre des agents). De plus, il est possible pour les agents d'échanger leur nourriture. La question qui se pose alors est de savoir comment organiser la collectivité pour maximiser la récompense globale.

5.1 Modèle

Ce problème a été formalisé dans le cadre des Interac-DEC-MDP de la façon suivante :

Les deux agents sont caractérisés par leur faim et leur possession de nourriture. Chaque agent peut soit **plonger** pour aller chercher de la nourriture, soit **attendre** dans la cage ou **manger** la nourriture qu'il pourrait avoir en sa possession. Les actions des agents sont indépendantes les unes des autres. Les fonctions de récompenses sont identiques pour les deux agents. Si un agent plonge, il reçoit une récompense négative α dépendant de la catégorie donnée a priori de l'agent (pour un bon plongeur $|\alpha|$ vaut 10, pour un mauvais plongeur 150). Si un agent mange de la nourriture, il reçoit une récompense positive (+100). A chaque pas de temps où l'agent ne mange pas, sa faim augmente de 1 et l'agent reçoit une récompense négative ($-10.faim_i$).

Enfin, les agents peuvent échanger de la nourriture. Il s'agit d'une interaction directe (I_1) avec deux résultats possibles : soit l'échange a effectivement lieu entre les deux agents ($RI_{1,1}$), soit l'échange n'est pas effectif ($RI_{1,2}$). Cette interaction sera mise en concurrence avec l'absence d'interaction pour la politique de déclenchement.

5.2 Scénario

Deux scénarios sont envisagés :

- Dans le premier, les deux agents sont des agents bons plongeurs. Dans ce cas, l'échange de nourriture est inutile puisque le coût associé à la plongée est identique pour les deux agents. L'objectif de ce scénario est de vérifier la disparition des interactions lorsque celles-ci ne sont pas nécessaires.
- Dans le second scénario, les deux agents mis en présence n'ont pas les mêmes capacités : le coût de plongée associé à un agent, l'agent qualifié d'agent mauvais plongeur, est plus important. Ce scénario avait pour objectif de vérifier à quelle amélioration globale des performances du système les interactions pouvaient conduire.

Pour chacun de ces scénarios, les trois phases dépendantes d'apprentissage présentées précédemment ont été conduites : apprentissage des politiques individuelles π_i sans déclencher d'interactions pendant 20000 pas de temps, puis apprentissage des politiques $\Pi_{i,j}$ en déclenchant ces interactions de façon aléatoire ; enfin apprentissage de la politique de déclenchement $\pi_{trig,i}$. Enfin, ces résultats sont comparés aux résultats optimaux obtenus pour un PDM centralisé équivalent (avec prise en compte d'interactions de manière centralisée). Vu le faible nombre d'agents, il est en effet encore possible d'utiliser une approche centralisée. Cependant, celle-ci ne pourra plus être envisagée si le nombre d'agents croît.

5.3 Résultats

Agents homogènes Le comportement optimal dans ce cas (calculé à partir du PDM centralisé) consiste à plonger puis manger sans interagir. L'apprentissage effectué conduit aux mêmes résultats. En outre, cet apprentissage conduit à la disparition des interactions qui ne sont pas utiles.

Agents hétérogènes La présence d'agents hétérogènes constitue une situation plus intéressante (cf Tableau 1). Lors de la première phase, les deux agents apprennent le comportement optimal en absence d'interaction. Par la suite, quand les interactions sont sys-

Agent(s)	r après π_i	nb_I	r après $\Pi_{i,j}$	nb_I	r après $\pi_{trig,i}$	nb_I
Bon Plongeur	40	0	16.6	1	16.6	0
Mauvais Plongeur	-27.5	0	23.3	1	23.3	0.33
Groupe	12.5	0	40	2	40	0.33

TAB. 1 – Récompenses (r) et nombre d'interactions (nb_I) aux différentes étapes d'apprentissage avec deux agents hétérogènes, chaque étape se situe à la suite de l'apprentissage des π, Π et π_{trig}

tématiquement déclenchées, l'agent bon plongeur apprend à donner sa nourriture quand l'agent mauvais plongeur est à l'initiative d'une interaction et ne dispose pas de nourriture. Le résultat de l'interaction peut alors être compris comme une action "altruiste" de la part de l'agent bon plongeur (ses récompenses reçues passent de 40 à 16.6) intéressante pour le système (la récompense globale passe de 12.5 à 40). Enfin, la phase 3 permet de limiter les interactions émises dans le système aux interactions utiles. Il est à noter que le comportement optimal n'est pas atteint. Celui-ci consiste pour l'agent bon plongeur à plonger systématiquement sans chercher à se nourrir et pour l'agent mauvais plongeur à prendre et manger systématiquement la nourriture du plongeur. Ce comportement n'est pas atteint car le comportement individuel de l'agent mauvais plongeur n'est pas remis en cause lors de l'ajout d'interactions dans le système. Afin de résoudre cette difficulté, d'autres apprentissages plus élaborés sont nécessaires.

5.4 Discussion

Cet exemple montre qu'il est possible de construire des comportements collectifs pertinents à partir d'apprentissages simples sans représentation explicite des agents. Un simple échange de Q-valeurs entre agents est suffisant pour décider de manière efficace de l'issue d'une interaction.

De plus, l'exemple met en évidence l'intérêt de la notion d'interaction par rapport à celle d'action classique. Une action, lorsqu'elle est exécutée, a un effet indépendant des caractéristiques comportementales de l'agent qui en subit les conséquences (point de vue purement local et égoïste). Le concept d'interaction permet aux agents un accroissement de la somme de leurs récompenses espérées et répond partiellement à des problématiques proches de la tragédie des communs (en limitant pour le moment le nombre d'agents impliqués simultanément à 2).

L'exemple que nous avons présenté reste relativement simple : il n'implique que deux agents et les actions d'un agent n'ont pas d'influence sur le résultat des actions de l'autre agent. Nous envisageons par la suite d'autres problèmes plus complexes impliquant un plus grand nombre d'agents comme le "problème des dockers". Après l'apprentissage des interactions, les politiques individuelles ne sont pas mises à jour et ceci empêche d'atteindre le comportement optimal. Envisager un apprentissage cyclique semble alors une bonne solution pour obtenir des comportements collectifs plus efficaces.

6 Conclusion

Dans cet article, nous nous sommes intéressés à la construction automatique des comportements individuels et des interactions d'un système composé d'agents réactifs afin que celui-ci se comporte comme un tout cohérent et parvienne à résoudre un problème posé à la collectivité. Un formalisme original a été présenté : les interac-DEC-MDP. Ce formalisme est constitué de deux modules, un module d'action permettant aux agents de choisir simultanément des influences à exercer sur le système et un module d'interaction permettant des prises de décisions collectives. Ce dernier module permet ainsi d'autres considérations que de simples considérations égoïstes et peut conduire de ce fait à une amélioration globale du système par rapport à des apprentissages purement individuels.

Ce formalisme permet de construire automatiquement des comportements collectifs à partir d'évaluations locales de l'avancement de la tâche globale (principe de rationalité limitée). Un premier problème simple de partage de ressources impliquant 2 agents a été modélisé selon le formalisme proposé et les comportements des agents ont été construits automatiquement par Q-learning. Les premiers résultats montrent qu'il est possible à partir d'apprentissages simples de générer des comportements collectifs pertinents, en limitant lors de l'exécution les interactions utilisées à celles utiles pour le système. Un apprentissage plus élaboré est envisagé dans le futur avec un plus grand nombre d'agents pouvant interagir entre eux de manière indirecte. L'objectif à long terme de ces travaux consiste à vérifier s'il est possible, à partir d'interactions locales, de générer automatiquement des structures utiles par apprentissage.

Références

[BCT03] Christine Bourjot, Vincent Chevrier, and Vincent Thomas. A new swarm mechanism based on social spiders colonies : from web weaving to region detection. *Web Intelligence and Agent Systems : An International Journal - WIAS*, 1(1) :47–64, Mar 2003.

[BDFJ91] Stéphane Bura, Alexis Drogoul, Jacques Ferber, and Eric Jacopin. Eco-résolution : un modèle de résolution par interactions. In *8ème congrès RFIA*, pages 1299–1308, 1991.

[Buf03] Olivier Buffet. *Une double approche modulaire de l'apprentissage par renforcement pour des agents intelligents adaptatifs.* PhD thesis, Université Henri Poincaré - Nancy 1, 2003.

[BZI00] Daniel Bernstein, Shlommo Zilberstein, and Neil Immerman. The complexity of decentralized control of markov decision processes. In *Proceedings of the*

16th Conference on Uncertainty in Artificial Intelligence (UAI'00), Stanford, California, pages 32–37, 2000.

[DDCG99] Marco Dorigo, Gianni Di Caro, and Luca M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2) :137–172, 1999.

[GCC03] Franck Gechter, Vincent Chevrier, and François Charpillet. Une architecture réactive pour la localisation en robotique mobile. In *JFSMA 2003*, pages 345–358, 2003.

[GGG03] Jean-Pierre Georgé, Marie-Pierre Gleizes, and Pierre Glize. Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie amas. *Revue d'Intelligence Artificielle RIA*, 17(4) :591–626, 2003.

[Har68] Garrett Hardin. The tragedy of the commons. *Science*, 162 :1243–1248, 1968.

[LF94] Erik D. Lumer and Baldo Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the third international conference on Simulation of adaptive behavior : from animals to animats 3*, pages 501–508. MIT Press, 1994.

[PT02] David V. Pynadath and Milind Tambe. The communicative multiagent team decision problem : Analyzing teamwork theories and models. *Journal of AI research*, 16 :389–423, 2002.

[Rey87] Craig Reynolds. Flocks, herds, and schools : A distributed behavioral model. *SIGGRAPH, Computer Graphics*, 21 :25–34, 1987.

[SB98] Richard Sutton and Andrew Barto. *Reinforcement Learning : an introduction*. Bradford Book, MIT Press, Cambridge, MA, 1998.

[SF00] Olivier Simonin and Jacques Ferber. Modeling self satisfaction and altruism to handle action selection and reactive cooperation. In *6th conference of simulation and adaptive behaviour*, pages 314–323, 2000.

[SPG03] Yoav Shoham, Rob Powers, and Trond Grenager. Multi-agent reinforcement learning : a critical survey. Technical report, 2003.

Table des matières

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399