HAL
open science

Approche expérimentale pour la compréhension des systèmes multi-agents réactifs

François Klein, Christine Bourjot, Vincent Chevrier

To cite this version:

François Klein, Christine Bourjot, Vincent Chevrier. Approche expérimentale pour la compréhension des systèmes multi-agents réactifs. Journées Francophones sur les Systèmes Multi-Agents - JFSMA 2006, Oct 2006, Annecy/France. inria-00104308

HAL Id: inria-00104308

https://inria.hal.science/inria-00104308v1

Submitted on 6 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Approche expérimentale pour la compréhension des systèmes multi-agents réactifs

KLEIN François, BOURJOT Christine et CHEVRIER Vincent

LORIA
BP 256
54506 Vandoeuvre-lès-Nancy Cedex
klein@loria.fr

RÉSUMÉ: Cet article s'intéresse à la compréhension et au contrôle des systèmes multi-agents réactifs. Nous proposons une approche pour approximer, de manière expérimentale, le comportement du système comme une fonction de ses paramètres. L'originalité de ce travail consiste en la génération automatique de données pour entraîner le modèle, ce qui permet de réduire le nombre d'exécutions nécessaires à la construction de cette approximation tout en maximisant sa précision. Nous détaillons notre proposition et fournissons un exemple d'implémentation. Quelques expériences sont présentées et nous commentons les résultats obtenus.

ABSTRACT: This paper is about the understanding and control of multi agent system. We propose an approach to experimentally approximate the behavior of the system as a function of its parameters. The originality of our proposition is the ability to dynamically generate data to train the model and therefore reducing the number of executions needed to build the approximation while maximizing its accuracy. We detail our proposition and provide an example of how to put it into practice. Some experiments are presented and the obtained results are discussed.

MOTS-CLÉS : Agent, émergence, auto-organisation, simulation multi-agents
KEYWORDS : Agent, emergence, self-organisation, multi-agents simulation

1 Introduction

Le contexte de ce travail est l'étude et l'analyse de systèmes capables de produire une réponse collective à partir des interactions entre des individus réactifs. L'essence de ces systèmes est leur capacité à présenter une réponse collective complexe qui dépasse la simplicité des individus, sans que le lien entre ces deux niveaux soit explicite. Cette complexité ne se situe donc pas dans les agents mais dans leurs interactions : les propriétés du système peuvent être qualifiées d'émergentes. Ces systèmes sont principalement caractérisés par une dynamique non linéaire (de petites fluctuations peuvent entraîner des modifications significatives de la dynamique du système), et par une forte sensibilité aux conditions initiales et aux paramètres (présence de différents bassins d'attractions et de zones de bifurcation). Par conséquent, une étude réductionniste du système ne peut aboutir : les propriétés globales ne peuvent pas être comprises en examinant simplement les composants du système séparément.

Par exemple, les équations classiques suivantes modélisent la dynamique d'une population, qui varie entre 0 et une valeur maximale normalisée à 1, en fonction du temps : $x_{n+1} = R.x_n(1-x_n)$ où R est un paramètre assimilable au taux de reproduction. Dans cet exemple, des bifurcations apparaissent pour différentes valeurs de R. La taille de la population vers laquelle le système converge est un attracteur, fixé par l'auto-organisation du système réel. Pour des valeurs données de R, différents attracteurs peuvent être identifiés, conduisant à une oscillation du système entre eux.

Les systèmes multi-agents (Weiss 99) font partie de ces systèmes potentiellement émergents, et permettent de les décrire comme un ensemble d'agents qui interagissent directement ou au travers d'un environnement partagé. Il sont ainsi de bons candidats (Fromm 04) pour modéliser les autres systèmes complexes grâce à leur ouverture à l'expérimentation virtuelle. On peut utiliser un système multi-agents pour expliquer les mécanismes qui sous-tendent leur dynamique ou celle de tout système complexe qu'il modélise (on peut trouver des exemples de la modélisation de sociétés auto-organisées en biologie : Camazine et al. 01), et pour prédire le comportement du système quand les conditions initiales changent ou quand des perturbations surviennent.

Cet article présente une approche pour comprendre et contrôler les systèmes multi-agents (SMA). Nous définissons notre problématique dans le chapitre suivant. Nous décrivons notre proposition dans le chapitre 3 puis fournissons un exemple d'utilisation au chapitre 4. Nous mettons enfin en relief les avantages et inconvénients d'une telle approche et proposons des améliorations envisageables au chapitre 5.

2 Position du problème

Le comportement du SMA est défini par l'ensemble des paramètres locaux, qui fixent

le comportement des agents et les règles de la dynamique de l'environnement. Etant donné un ensemble de paramètres et de conditions initiales, le système produit une propriété ou un schéma collectif remarquable (Campagne 05). Notre problème est alors d'établir une relation entre ces valeurs locales et des caractéristiques du comportement global. Cet objectif peut être divisé en trois parties : la *compréhension* : connaître le comportement global du système sur un sous-espace de ses paramètres; le *contrôle* : capacité à déterminer un ensemble de valeurs pour les paramètres, de manière à obtenir un comportement global désiré; l'*optimisation* : trouver, comme pour le contrôle, un ensemble de valeurs de paramètres qui fournissent un comportement global désiré et qui soit optimal selon un critère d'évaluation de ce comportement.

Etant données les propriétés propres aux systèmes multi-agents, en particulier leur caractère potentiellement complexe, trouver des réponses à de telles questions est a priori une tâche difficile. Deux approches principales peuvent être envisagées pour les traiter (Kazadi et al. 04), (Savit et al. 98) : i) utiliser un cadre formel dans lequel le SMA peut être exprimé, et étudier le comportement dans ce cadre, et/ou; ii) étudier expérimentalement le SMA en exécutant plusieurs simulations avec différents réglages de paramètres, et approximer son comportement à partir de ces données (une simulation du système avec des paramètres fixés est appelée *réplication*). De nombreux auteurs (Bourjot et Chevrier 04), (Mamei et al. 04), (Edmonds 04), (Edmonds et al. 04), (Holvoet et al. 05) affirment que l'étude des systèmes multi-agents nécessite une approche expérimentale. Par exemple, (Wooldridge et al. 98) fait explicitement référence à la démarche expérimentale : « *the development of any agent system, however trivial, is essentially a process of experimentatio*n ».

3 Proposition

3.1 Généralités

Dans la suite de cet article, on ne considère que les problèmes de compréhension et de contrôle du système en utilisant une approche expérimentale. Plusieurs travaux s'occupent du même problème en caractérisant la fonction globale du système (Holvoet et al. 05), ou en fournissant quelques étapes clés à suivre lorsque l'on étudie le système (Puppe et al. 05). La caractéristique essentielle de notre proposition est la réduction du nombre de réplications par la définition de plans d'expériences dynamiques, évoqués dans (Amblard 03). De notre point de vue, le comportement du système repose sur :

– un ensemble P de paramètres, qui sont les entrées du système, et qui correspondent aux valeurs que le concepteur peut fixer dans le système avant de l'exécuter ; on note $P=\{p_1, p_2,..., p_n\}$ où chaque paramètre p_i est défini sur un domaine $dom(p_i)$. Dans la suite, $dom(p_i)$ sera noté dom_i.

– un ensemble O de sorties qui correspondent aux valeurs observables du système, $O=\{o_1, o_2,..., o_m\}$ où o_j est défini sur un domaine dom_j. Ces sorties sont les descripteurs

du comportement du système.

Ce comportement peut donc être vu comme une fonction F qui lie les entrées aux sorties du système. Pour une valeur $p \in P$ donnée, l'exécution du SMA calcule la valeur de sortie $o \in O$, ce qui définit la fonction F *in extenso*. Le comportement F est une fonction qui va du produit cartésien des domaines des paramètres dans le produit cartésien des domaines des descripteurs : F(p)=o.

Dans ce contexte, on peut définir plus précisément ce que sont la compréhension et le contrôle du système. La compréhension est la capacité à répondre à cette question : pour un ensemble de valeurs p donné, quelle sera la réponse du système représentée comme un ensemble de valeurs o ? Le contrôle correspond à la question complémentaire : on fixe o, et l'on souhaite connaître un ensemble de valeurs p tel que F(p)=o.

Dans chacun de ces cas, le problème est de construire une approximation de F ou de F^{-1}. Pour y parvenir, nous proposons une approche en quatre étapes :

– restreindre l'espace de recherche aux zones susceptibles de présenter un intérêt, en considérant des sous-espaces de P et de O en accord avec notre objectif d'étude et en se basant sur la connaissance que l'on peut avoir a priori du système,

– sélectionner un modèle adapté selon cette connaissance et notre objectif,

– entraîner le modèle à partir de données expérimentales,

– et enfin le valider.

3.2 Simplification : restriction du domaine et du co-domaine

Bien qu'elle corresponde à une connaissance complète du système, la construction d'un modèle sur l'ensemble de l'espace des paramètres P à valeur dans tout l'espace de sortie O souffre de plusieurs inconvénients. De manière pratique, même si l'on suppose que P et O sont des espaces discrets, ou s'ils sont discrétisés, la taille de l'espace sera au minimum de Card(P)*Card(O), et elle peut être encore bien plus grande si l'on considère le temps ou le caractère stochastique du système. Le temps nécessaire au déroulement des réplications pour le système rend cette approche irréaliste. Dans certaines zones de P, le comportement du système ne présente aucun intérêt, que ce soit parce que ces combinaisons de paramètres ne correspondent à aucun cas réel ou bien sont impossibles, ou encore parce que la réponse du système en ces points est triviale ou déjà connue. Parmi toutes les entrées p_i et les sorties o_i, seules certaines d'entre elles peuvent présenter un intérêt pour notre étude : certaines entrées peuvent n'avoir que peu ou pas d'influence sur le comportement du système, et certaines sorties peuvent ne pas apporter d'informations originales sur ce comportement, dans l'optique que l'on s'est fixée. Par exemple, dans un système de dynamique de population, la position de chaque agent ne présente que peu d'intérêt pour le contrôle de la population totale, et peut être avantageusement remplacée par le calcul d'une ou plusieurs valeurs indiquant la

dispersion de ces agents. Le comportement du système peut varier qualitativement de manière différente sur plusieurs intervalles de paramètres, auquel cas on peut envisager de considérer séparément plusieurs approximations pour chacun de ces comportements.

C'est pourquoi nous pouvons supprimer des zones de P et de O de notre étude, en restreindre d'autres, et remplacer des dimensions de l'espace O par d'autres plus adaptés à notre étude, qui seraient des agrégations de valeurs o_i et des descripteurs des régularités du comportement du système, de manière à synthétiser l'information. Le but de cette première étape est de remplacer P par un ensemble E de paramètres pertinents, ceux qui permettent de contrôler le système, toujours en accord avec l'objectif de l'étude, et de remplacer O par un ensemble I de descripteurs pertinents du comportement. Pour y parvenir, du moins en partie, il existe un outil statistique très utile : les plans d'expériences. Ils consistent en une étude analytique de l'espace des paramètres, en se contentant le plus généralement de ses bornes, afin de déterminer quels sont les paramètres qui ont statistiquement une réelle influence sur le comportement qualitatif du système, et quels sont ceux dont l'influence est assimilable à du bruit.

3.3 Sélection du modèle

Nous nous intéressons maintenant à établir une relation F' entre des paramètres dirigeant le comportement local du système et certaines caractéristiques du phénomène global $F: E \rightarrow I$. Si l'on suppose que les domaines sont discrets, il est possible de représenter F de manière extensive, en assimilant F' à chaque couple composé d'un points et de la réponse associée : $F \equiv (p, o) = F'$. Comme nous l'avons vu ci-dessus, cette approche se limite aux systèmes les plus simples à cause de la taille de ces données.

Nous devons donc l'approximer, c'est-à-dire trouver une représentation de F à la fois plus simple et plus commode vis-à-vis de notre objectif. Comme il existe différentes possibilités pour cette représentation, il faut sélectionner une famille de modèles en fonction de sa précision de représentation, sa simplicité et sa facilité d'utilisation, sa compacité, le temps nécessaire pour le construire, etc. Un exemple de représentation est une expression paramétrique, comme un polynôme fonction des paramètres du système, dont il faut déterminer les coefficients ; une famille de modèles est alors un type de polynômes, et un modèle de cette famille est un ensemble de coefficients. Le type du modèle peut aussi être un estimateur statistique, tel qu'un réseau de neurones (Aussem et al. 99), une famille de modèles est alors un réseau particulier, et une instance de cette famille est décrite par le poids de chaque connexion. Choisir une représentation signifie donc déterminer une famille de modèles, en accord avec l'étude préliminaire du système : il n'est pas pertinent de supposer que le système se comporte linéairement, et de choisir une famille de modèles associée, si l'on peut démontrer en quelques réplications que ce n'est pas le cas. A l'inverse, il est également inutile de choisir un modèle très élaboré, qui permet de représenter de nombreux comportements complexes, si on ne se sert que d'une petite partie de ses capacités.

3.4 Entraîner le modèle : les plans d'expériences dynamiques

Une fois la représentation choisie, elle doit être instanciée à l'aide du système étudié. Comme nous l'avons remarqué, il faut tenir compte à la fois de la précision du modèle et du temps nécessaire pour le construire ou l'entraîner. En d'autres termes, nous souhaitons minimiser le nombre de réplications nécessaires pour le construire tout en maximisant la précision de ses prévisions.

Le terme de plans d'expériences, dont nous avons vu le but de déterminer les paramètres influents, désigne également des outils destinés à étudier de manière exhaustive et planifiée le domaine d'entrée d'un système pour obtenir une connaissance du comportement de ce système sur tout ce domaine. On différencie ces deux objectifs par les termes de screening et de méthodologie des surfaces de réponse (Vivier 02). On remarque que le second correspond parfaitement à ce que nous souhaitons faire ici (Kleijnen 87). Toutefois la plupart de ces plans d'expériences sont adaptés pour des systèmes simples, typiquement linéaires avec peu de paramètres, or nous considérons des systèmes dont le comportement complexe dépend de nombreux paramètres ; nous avons besoin par conséquent d'un nouveau type de plans d'expériences, qui permette de trouver des surfaces de réponses, tout en restant adaptés aux systèmes multi-agents que nous étudions. Nous proposons pour cela d'établir la planification des expériences à effectuer de manière dynamique, c'est-à-dire de déterminer le point de l'espace des paramètres où aura lieu la prochaine réplication en fonction des résultats déjà obtenus (Amblard 03). L'intérêt d'un plan d'expériences dynamique est qu'il homogénéise la précision du modèle sur l'espace des paramètres en choisissant toujours un réglage de paramètres pertinent, donc il permet soit de réduire le nombre de réplications nécessaires pour entraîner le modèle, soit d'augmenter sa précision pour un nombre donné de réplications. L'exécution du plan d'expériences dynamique s'achève donc lorsqu'une précision désirée est atteinte, ou quand un nombre maximal de réplications a été mené.

Donc au sein d'une zone donnée de l'espace des paramètres E, soit le modèle est construit et nous savons que la précision désirée est atteinte, soit le modèle ne peut pas fournir de réponse fiable et nous savons que la cause en est que la construction de tout modèle dans cette zone nécessite un nombre de réplications supérieur à celui que nous avons autorisé pour des questions de temps. La manière dont la précision du modèle est rendue homogène sur E dépend fortement du modèle choisi : pour que les plans d'expériences dynamiques soient applicables, on doit supposer que le modèle n'est pas sensible à l'ordre dans lequel on lui procure ses données d'entraînement, et que la précision du modèle peut être estimée pendant son entraînement même, éventuellement de manière approximative à l'aide d'heuristiques.

Le principe à retenir est que l'on génère dynamiquement le plan d'expériences en évaluant de manière itérative la précision, et si elle n'est pas suffisante, on choisit de nouveaux réglages de paramètres pour une réplication.

3.5 Validation et contrôle

Une fois le modèle entraîné, il doit être validé, donc reconnu comme une bonne approximation du système initial. On compare pour cela la prédiction du modèle avec la réponse du système sur un grand nombre de points aléatoires de l'espace d'entrée. Si nous avons un modèle valide, nous avons construit un lien entre les paramètres locaux et le comportement global du système multi-agents, et nous pouvons dire que nous avons compris ce système, selon la définition que nous avons proposé pour la notion de compréhension. Il nous reste donc à contrôler ce système.

La famille de modèles que nous avons choisie n'a aucune raison d'être inversible, on entend par là de fournir directement un point de l'espace E tel que la réponse du système en ce point puisse être représentée par des valeurs de descripteurs données, donc de permettre le contrôle du système. Il est heureusement possible de se passer de cette inversibilité, en la créant artificiellement à l'aide de méta-heuristiques, comme des algorithmes génétiques, des méthodes de décente de gradient ou de recuit simulé. L'avantage d'appliquer ces algorithmes au modèle du système plutôt qu'au système lui-même est un fort gain de temps : les méta-heuristiques sont des méthodes qui fonctionnent par approximations et essais successifs de points de l'espace d'entrée d'une fonction de satisfaction, elles demandent donc de nombreuses valeurs de cette fonction.

Dans notre cas, elles demanderaient surtout la réponse du système, donc de la fonction F, sur de nombreux points de l'espace E. C'est pourquoi la connaissance d'une fonction approchée F' permet d'accélérer ces méthodes en rendant les simulations du système multi-agents inutiles lors de la phase de contrôle. Bien entendu, ces mêmes simulations auront été nécessaires lors de la phase de modélisation, mais si nous désirons pouvoir contrôler rapidement le système, ou encore effectuer ce contrôle de nombreuses fois, il est plus avantageux de passer par un modèle, comme nous l'avons fait.

4 Instanciation de la proposition

Notre proposition est basée sur des principes généraux que nous nous proposons d'illustrer dans ce chapitre sur un exemple. Il faut préciser que nous ne cherchons pas ici à obtenir un modèle qui représente parfaitement un système multi-agents, mais nous souhaitons plutôt souligner la manière dont on met en pratique les principes proposés. Toutefois, la famille de modèles utilisée et l'algorithme de planification dynamique des expériences sont suffisamment génériques pour être réutilisés avec d'autres SMA.

4.1 Système d'étude : une dynamique de population

Le problème que nous avons choisi d'étudier est celui d'une dynamique de

population. Des agents (des moutons dans un pré) vivent dans un environnement partagé aux ressources limitées. Cet environnement est une grille torique où chaque cellule contient une quantité de nourriture déterminée qui augmente de manière linéaire au cours du temps. Les agents se déplacent aléatoirement dans cet environnement, et ils peuvent y puiser de la nourriture pour augmenter leur niveau d'énergie. Pour un agent donné, si son niveau d'énergie est trop bas, il meurt, et si le niveau d'énergie dépasse un certain seuil, il donne naissance à un nouvel agent. Nous nous intéressons au comportement de la population : dans quelles conditions la population s'éteint-elle ? Se stabilise-t-elle après un certain nombre de cycles ? Evolue-t-elle entre certaines limites ?

Ce problème est bien connu et a été implémenté à l'aide de nombreux outils de simulation, et il peut être décrit par l'équation suivante, communément appelée équation logistique : $x_{n+1} = Rx_n(1-x_n)$ qui nous renseigne sur le comportement du système. Même si une telle équation nous fournit des informations sur le comportement global, elle ne peut pas nous aider à répondre aux questions liées aux paramètres individuels et locaux.

4.2 Etapes de l'approche

4.2.1 Restriction des domaines de recherche

Le système multi-agents étudié est décrit par des conditions initiales (le nombre initial d'agents, leur niveau initial d'énergie) et par de nombreux paramètres : le taux de croissance Tx, la quantité maximale de nourriture consommée à chaque cycle Qt, l'énergie perdue par les agents à chaque cycle Dp, le seuil de naissance Se, la taille de l'environnement, la quantité maximale de nourriture disponible sur une cellule de l'environnement, etc.

Les valeurs o_j mesurables pendant l'exécution du système sont, au niveau local, toutes les caractéristiques des agents et des cellules, et au niveau global, toute valeur qui peut être calculée à partir d'une agrégation de valeurs locales d'un ensemble d'éléments du système pour une réplication (telle que la taille de la population, les valeurs maximale, minimale ou moyenne d'une caractéristique), ou encore une valeur statistique calculée à partir de ces précédentes valeurs pour plusieurs réplications en un même point de l'espace des paramètres.

Le choix des paramètres et des descripteurs à étudier est lié à ce que l'on souhaite connaître sur le comportement du système. L'étude de l'équation logistique nous indique que les conditions initiales de ce système n'ont pas d'influence sur son comportement global, à condition qu'elles ne soient pas trop proches de zéro. On remarque que nous pouvons nous passer de paramètres tels que la taille de l'environnement ou la quantité maximale de nourriture disponible par cellule, et nous décidons de fixer leurs valeurs : nous obtiendrons des résultats pour des valeurs données de ces paramètres, mais qui seront facilement généralisables à d'autres cas. D'autres valeurs (l'orientation initiale des

agents par exemple) n'ont pas d'influence sur le comportement global, comme nous pouvons le voir par exemple à l'aide d'un plan d'expériences classique, et nous les choisissons aléatoirement à chaque réplication afin de ne pas introduire de biais.

Finalement, nous ne conservons que quatre paramètres pour notre ensemble E : le taux de croissance $Tx \in [0.01, 1]$, la quantité maximale de nourriture consommée à chaque cycle $Qt \in [2, 10]$, l'énergie perdue par les agents à chaque cycle $Dp \in [1, 9]$, le seuil de naissance $Se \in [10, 100]$.

Les intervalles sont choisis de manière à ce que le comportement soit intéressant et les paramètres n'entrent pas en contradiction les uns avec les autres, conformément à ce que nous avons vu à la section 3.2, et selon les observations que l'on a pu faire du système. Par exemple, il est normal que les agents puissent consommer davantage de nourriture à chaque cycle qu'ils n'en dépensent. Nous nous plaçons donc dans le cas le plus difficile pour l'utilisation d'un plan d'expériences dynamique, ce qui ajoute de la valeur à notre étude.

Pour les descripteurs du comportement, nous choisissons d'effectuer plusieurs réplications en chaque point d'expérience dans l'espace des paramètres, de les arrêter après un nombre fixé de cycles et de considérer, pour l'ensemble de ces réplications, la moyenne μ de la population, son écart-type σ, et la proportion cv0 de réplications qui convergent vers zéro. En dépit du comportement stochastique des agents, ces descripteurs suffisent à discriminer les différents comportements du système selon le point de vue que l'on s'est fixé : le modèle logistique nous apprend en effet que la population peut, de manière simplifiée, soit s'éteindre, soit converger vers une valeur fixe, soit osciller entre plusieurs valeurs, soit enfin adopter un comportement chaotique ; et l'observation nous montre que dans ce dernier cas, il y a de nouveau un risque qu'elle s'éteigne. En comparant nos trois descripteurs, nous sommes donc en mesure de déterminer dans quelle situation le système se trouve.

4.2.2 Choix du modèle

Comme nous souhaitons vérifier si un plan d'expériences dynamique permet de réduire effectivement le nombre de réplications lors de la construction du modèle, nous décidons d'instancier un modèle dédié à chaque descripteur. Nous faisons l'hypothèse que le système se comporte de telle manière qu'il est possible d'interpoler linéairement ses résultats entre deux points connus, en fonction d'un seuil arbitraire, comme présenté sur la figure 1.

Les domaines des paramètres et des descripteurs sont numériques, nous décidons par conséquent d'un seuil arbitraire δ associé au système et lié à la pente moyenne des indicateurs. Comme nous l'expliquons en détail au paragraphe suivant, nous considérons que le modèle est précis si sa pente moyenne est plus petite que δ. Nous construisons l'ensemble minimal de points $S=\{s \in E \times P\}$ tel que les propriétés de précision soient respectées. Des points adjacents dans S définissent une maille.

4.2.3 Plan d'expériences dynamique

Nous présentons ici l'algorithme qui permet d'entraîner les modèles, en ne considérant qu'un seul descripteur pour clarifier l'explication, mais le principe reste le même pour nos trois descripteurs. Initialement, S correspond aux sommets de E et aux valeurs du descripteur correspondant à ces points. S n'est composé que d'une seule maille qui n'est pas considérée comme explorée.

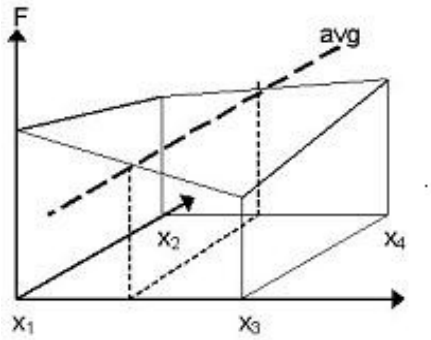

Figure 1. Approximation de F sur une maille. Sur l'axe x_1x_2 on calcule les valeurs $F(x_1) – F(x_2)$ et $F(x_1+\varepsilon) - F(x_2+\varepsilon) = F(x_3) – F(x_4)$ et on compare leur moyenne à δ.

Le principe de l'algorithme est de vérifier la validité de la condition de précision du modèle sur toutes les mailles non explorées du modèle, et si elle n'est pas respectée, de diviser cette maille en sous-mailles plus fines et de les considérer récursivement. Comme nous ne pouvons pas prévoir la taille des mailles telle que la précision du modèle sur cette maille soit nécessairement satisfaisante, nous fixons une taille minimale limite aux mailles. Par conséquent, lorsque la taille d'une maille considérée est inférieure à cette limite, la maille est considérée comme inexploitable.

L'algorithme termine et construit un modèle dont la complexité en temps et en espace est bornée. A la fin de son exécution, une maille est soit explorée, soit inexploitable.

L'algorithme fonctionne de la manière suivante :

pour chaque maille non explorée $m \in S$
  si la taille de m est inférieure à la taille limite, on la marque comme inexploitable
  sinon, pour chaque dimension de E
    calculer la moyenne avg de la pente de F sur la maille le long de cette dimension
    calculer l'écart-type σ entre avg et la pente de F le long de chaque arrête de la maille m selon cette dimension (voir figure 1)
    si $\sigma < \delta$, la maille m est considérée comme explorée
    sinon, diviser m en sous-mailles et les ajouter à S en tant que mailles non explorées

Pour diviser une maille en sous-mailles, nous considérons simplement le milieu de chaque segment entre deux sommets de la maille, comme pour une division en quadtree/octree. Dans notre étude, nous choisissons de ne considérer au plus que 16 segments par dimension de l'espace E (donc par paramètre d'étude). Comme nous avons 4 paramètres, le nombre de points à considérer pour construire S est d'au plus 17^4.

4.2.4 Validation

La validation du modèle n'a de signification que pour les mailles explorées et exploitables, elle s'effectue de manière statistique en comparant les valeurs approximées des descripteurs fournies par le modèle à celles fournies par l'exécution du système.

4.3 Résultats et commentaires

Nous avons effectué différentes expériences avec les trois indicateurs μ, σ et cv0. Pour vérifier la réponse du système en évitant d'introduire un biais dans l'expérimentation, nous avons effectué 10 réplications en chaque point d'expérience. Nous analysons les résultats suivant deux points de vue concernant:

– l'approche, nous répondons à la question : peut-on réduire significativement le nombre de réplications nécessaires pour approximer le système ?

– le modèle lui-même : les prédictions du modèle sont-elle satisfaisantes ?

Pour ce qui est de la qualité du modèle, l'entraînement s'arrête soit quand la taille d'une maille atteint la limite minimale, auquel cas les résultats sur cette maille ne sont pas exploitables, soit quand la maille est explorée et peut être utilisée pour prédire la réponse du système sur la zone de E qu'elle couvre. Nous devons donc considérer uniquement les réponses du modèle sur ses parties explorées. Pour cela, il faut distinguer le cas où cv0=100% (donc toutes les expériences montrent que la population s'éteint). Les descripteurs σ et μ sont alors nuls et peuvent être déduits de cv0 : un modèle, même bon, qui détaille ces descripteurs, n'apporte pas plus d'information qu'un modèle qui ne représente que l'indicateur cv0. Lorsque cv0 $\neq$ 100%, l'estimation de l'erreur de prévision ne peut se faire que pour les réplications qui ne tendent pas vers l'annulation de la population.

Pour résumer, les résultats doivent être analysés sur trois points :

– la couverture du modèle, soit la proportion d'espace exploré et exploitable,

– la quantité d'information pertinente du modèle, c'est-à-dire, la proportion de l'espace exploré où cv0 $\neq$ 100%

– et l'erreur de prédiction dans les zones pertinentes.

Nous avons utilisé pour nos expériences un seuil d'erreur $\delta = 0.05$, soit une erreur relative acceptable de 5%.

Une première étude expérimentale nous a fourni des résultats en grande partie inexploitables à cause de l'étendue des intervalles des paramètres. Nous nous sommes donc encore concentrés sur un espace E plus restreint, où le système présentait le comportement le plus intéressant, c'est-à-dire avec le plus de variations. Nous avons choisi comme domaines Tx $\in$ [0.44, 0.57], Qt $\in$ [5, 6], Dp $\in$ [1.5, 2.5] et Se $\in$ [49, 61], puis avons effectué une nouvelle campagne d'expériences.

Les résultats en sont donnés dans les tables 1 et 2 :

Descripteur	***Espace exploré***
μ	100%
σ	19.3%
cv0	100%

Tableau 1. Couverture du modèle (proportion de l'espace prise en charge par le modèle)

Nous avons également calculé la réduction du nombre de réplications nécessaires par rapport à un plan d'expériences non dynamique, et nous avons obtenu une réduction de 66%. Le modèle est totalement pertinent sur la partie de l'espace considéré (cv0 $\neq$100% en tout point, tableau 2). De plus, il approche particulièrement bien le système pour deux indicateurs, μ et cv0, avec une erreur relative de prédiction inférieure à 3%. Quant à σ, pour lequel nous obtenons de moins bons résultats, la raison peut être que les mailles considérées ne sont pas encore assez fines pour représenter efficacement le système dans ce sous-espace.

L'objectif principal de ces expériences était d'étudier la faisabilité dans l'utilisation de plans d'expériences dynamiques, les résultats obtenus nous encouragent donc à poursuivre cette voie : il est possible de réduire, de manière significative, le nombre de réplications nécessaires pour comprendre un système multi-agents en le modélisant. Les expériences montrent que notre modèle est capable de fixer l'essence du phénomène, et de fournir ainsi des résultats avec de faibles taux d'erreur dans des parties pertinentes de l'espace des paramètres E.

Descripteur	***Zone pertinente dans la couverture du modèle***	***Erreur relative de prédiction***
μ	100%	2.9%
σ	100%	11%
cv0	100%	0%

Tableau 2. Proportion des zones pertinentes de chaque modèle et erreurs de prédiction

Toutefois, nous avons fait des choix grossiers dans ce travail préliminaire : la division des mailles est systématique et non contextuelle, nous pouvons imaginer

construire des mailles en sélectionnant des points de manière à optimiser à la fois la pertinence des données utilisées pour entraîner le modèle et la taille des mailles, et ainsi augmenter encore le nombre de mailles explorées. Par ailleurs, le seuil δ utilisé pour définir le modèle est arbitraire, et il pourrait être intéressant de le sélectionner en fonction des données obtenues.

C'est pourquoi il nous reste à compléter cette étude préliminaire avec d'autre résultats expérimentaux pour confirmer les idées exposées dans l'article.

5 Conclusion et travaux à venir

Le travail présenté dans cet article se concentre sur la compréhension et le contrôle des systèmes multi-agents réactifs. Nous proposons une approche pour réaliser une approximation expérimentale du comportement du système, vu comme une fonction de ses paramètres, en dépit de son caractère éventuellement complexe. Nous cristallisons ainsi le lien qui existe entre les paramètres locaux du système et son comportement global. L'originalité de notre proposition réside dans la capacité à générer dynamiquement les données d'entraînement du modèle, et par conséquent à réduire le nombre, potentiellement rédhibitoire, de réplications nécessaires tout en maximisant la précision de notre approximation. L'approche proposée a en outre l'avantage d'être anytime (Zilberstein 96). La faisabilité de cette approche a été vérifiée au travers de quelques expériences sur un phénomène de dynamique de population, et son implémentation est suffisamment générique pour être appliquée à d'autres phénomènes.

Des expériences à venir nous permettront de vérifier plus précisément les propriétés du modèle entraîné pour ce même phénomène. Nous envisageons également d'utiliser la même approche pour créer d'autres modèles, et pour vérifier leur généricité sur différents phénomènes.

Références

Amblard F., Comprendre le fonctionnement de simulations individus-centrées : application à des modèles de dynamique d'opinion, thèse de doctorat, 2003.

Aussem A. et D. Hill, Wedding connectionnist and algorithmic modelling towards forecasting caulerpa taxifolia development in *the north-western mediterranean sea, Ecological Modelling*, pages 225–236, 1999.

Bourjot C. and Chevrier V., A platform for the analysis of artificial self-organized systems. *In Proceedings of Advances in Intelligent Systems - Theory and Applications AISTA 2004*, Luxembourg, 2004.

Camazine S., Deneubourg J.-L., Franks N., Sneyd J., Theraulaz G., and Bonabeau E., *Self-Organization in Biological Systems*, Princeton University Press., 2001.

Campagne J.-C., Systèmes multi-agents et morphologie, thèse de doctorat, Université de Paris 6, 2005.

Edmonds B. Using the experimental method to produce reliable self-organised systems, *In Proceedings of the 2nd International Workshop on Engineering Self-Organising Applications (ESOA 2004) at 3rd AAMAS*, New York, 2004.

Edmonds B. and Bryson J. The insufficiency of formal design methods the necessity of an experimental approach for the understanding and control of complex mas, *In Proceedings of the 3rd Internation Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'04)*, New York, pages 936–943, ACM, 2004.

Fromm J., *The emergence of complexity*, Kassel University Press, 2004.

Holvoet T., DeWolf T. and Samaey G., Engineering self-organising emergent systems with simulation-based scientific analysis, In S. Brueckner, G. Di Marzo Serugendo, D. Hales, and F Zambonelli, editors, *Proceedings of the Fourth International Workshop on Engineering Self-Organising Applications*, pages 1–15, 2005.

Kazadi S., Chung M., Lee B., and Cho R., On the dynamics of puck clustering systems, *Robotics and Autonomous Systems*, 46(1):1–27, 2004.

Kleijnen J.P.C., *Statistical tools for simulation practionners*, Marcel Dekker, 87.

Mamei M. Zambonelli F., Gleizes M.P. and R. Tolksdorf, Spray computers: Frontiers of selforganization, *In 1st International Conference on Autonomic Computing*, pages 268–269, May 2004.

Puppe F., Fehler M., Kluegl F., Techniques for analysis and calibration of multi-agent simulations, In Springer-Verlag GmbH, editor, *Engineering Societies in the Agents World* V: 5th International Workshop, ESAW 2004, Toulouse, France, October 20-22, 2004, Revised Selected and Invited Papers, number 3451, 2005.

Savit R. Riolo R.L. Parunak, H.V.D. Agent-based modeling vs. equation-based modeling: A case study and users guide, *In Proceedings of Multi-agent systems and Agent-based Simulation (MABS'98)*, Springer, LNAI 1534, pages 10–25, 1998.

Vivier S., Stratégies d'optimisation par la méthode des plans d'expériences et application aux dispositifs électrotechniques modélisés par éléments finis, thèse de doctorat, Université des sciences et technologies de Lille, 2002.

Weiss G., *Multi-agent systems a modern approach to distributed artificial intelligence*, MIT Press, 1999.

Wooldridge M. and Jennings N. R., Pitfalls of agent-oriented development, *In Proceedings of the 2nd International Conference on Autonomous Agents*, pages 385–391, 1998.

Zilberstein S. Using Anytime Algorithms in Intelligent Systems. *AI Magazine*, 17(3):73-83, 1996.