



HAL
open science

Actes des 28es Journées Francophones sur les Systèmes Multi-Agents

Nicolas Sabouret

► **To cite this version:**

Nicolas Sabouret. Actes des 28es Journées Francophones sur les Systèmes Multi-Agents. Plate-Forme Intelligence Artificielle, Association Française pour l'Intelligence Artificielle, 2020. hal-04573992

HAL Id: hal-04573992

<https://ut3-toulouseinp.hal.science/hal-04573992v1>

Submitted on 13 May 2024

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.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0
International License



AfIA

Association française
pour l'Intelligence Artificielle

JFSMA

Journées Francophones sur les Systèmes Multi-Agents

PFIA 2020

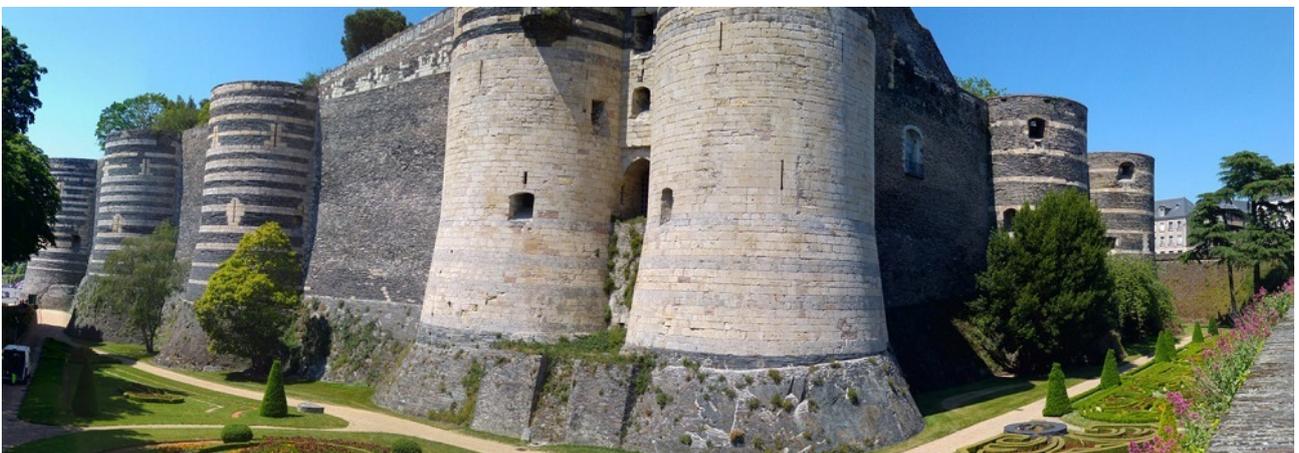


Table des matières

Nicolas SABOURET

Éditorial	4
Comité de programme	5
K. Aguilar Alarcon, M.-P. Gleizes, L. Caroux, E. Kaddoum, V. Camps, S. Combettes ReCoVac : Décision autonome de reprise de la conduite d'un véhicule autonome et connecté	6
J. Albouys Perrois, N. Sabouret, Y. Haradjic, M. Schumann Étude de différentes configurations d'autoconsommation collective de l'énergie à l'échelle du quartier à l'aide de la simulation multi-agent	16
M. Bourgeois, P. Taillandier, L. Vercoouter BEN : Une architecture pour des agents cognitifs, affectifs et sociaux dans la simulation	26
R. Chaput, O. Boissier, M. Guillermin, S. Hassas Apprentissage adaptatif de comportements éthiques	36
N. Cointe, A. Ghorbani, C. Chorus Un processus pour la sélection de stratégie de gestion de l'information dans les systèmes de confiance	46
P. Collard Micro-Motivations du Second Ordre : application à la ségrégation spatiale	56
B. Dato, M.-P. Gleizes, F. Migeon Stratégie d'apprentissage proactif endogène par systèmes multi-agents coopératifs	65
G. Picard, P. Rust Analyse des performances d'algorithmes DCOP pour l'association d'utilisateurs de réseaux HetNets	75
S. Picault, V. Sicard Les meilleurs agents sont ceux qu'on ne simule pas : vers des architectures de simulation multi-paradigmes ?	85
A.N. Rakotoarivelo, J.-M. Auberlet, R. Brémond Comportements hétérogènes de piétons et comportement émergent d'une foule	94
A. Rousset, P. Breugnot, B. Herrmann, C. Lang, L. Philippe Politiques de synchronisation dans les systèmes multi-agents distribués parallèles	103
J. Saunier, A. Pauchet Interactions dans les communautés mixtes humains-agents : état de l'art	113
F. Suro, J. Ferber, T. Stratulat, F. Michel Émergence de comportements collectifs basée sur l'apprentissage progressif individuel	123
I.F. Trentin, F. Ramparany, O. Boissier Intégration de la Gestion du Contexte et des Dispositifs IoT dans des Maisons Intelligentes basées sur des Systèmes Multi-Agents	133

Éditorial

Les Journées Francophones sur les Systèmes Multi-Agents (JFSMA) sont un moment privilégié d'échanges scientifiques transversaux. Elles réunissent des chercheurs qui étudient, utilisent et font évoluer le paradigme multi-agent pour adresser des problématiques issues de domaines liés à l'informatique (intelligence et vie artificielle, génie logiciel, robotique collective, etc.), à l'automatique et aux sciences humaines et naturelles (économie, sociologie, éthologie, etc.).

Le paradigme multi-agent, pluridisciplinaire, fournit un cadre conceptuel pour l'étude et la conception de systèmes dont la dynamique globale est le fruit d'entités autonomes — agents — qui interagissent dans un environnement commun. Les travaux de recherche qui sont associés à ce paradigme proposent ainsi des modèles, des méthodologies, des techniques et des outils qui permettent notamment de répondre à différents problèmes que l'on peut décliner selon quatre aspects :

- le développement de systèmes informatiques décentralisés où l'approche SMA permet l'intégration flexible et la coopération de logiciels et de services autonomes ;
- la résolution collective de problème pour laquelle il s'agit de résoudre de manière distribuée un problème qui se pose globalement à la collectivité d'agents ;
- la simulation de phénomènes complexes où la modélisation multi-agent apporte un cadre conceptuel permettant la représentation et la simulation de systèmes faisant intervenir différentes entités en interaction ;
- le développement de systèmes médiatisés où utilisateurs humains et agents artificiels interagissent directement ou indirectement, dans le cadre d'activités collectives de type éducatif, culturel ou social.

Les contributions relèvent de ces domaines en proposant des avancées conceptuelles, formelles ou logicielles. Elles peuvent également être analytiques (sur les problèmes, les enjeux, etc.) ou encore synthétiques (par exemple des retours sur expérimentations).

Par tradition, chaque édition des JFSMA met en exergue une thématique spécifique que les auteurs sont invités à prendre en compte dans leurs contributions, s'ils le souhaitent.

Les articles présentées correspondent aux versions de soumission à JFSMA 2020. Les actes définitifs ont été publiés chez Cépaduès.

Nicolas SABOURET

Comité de programme

Président

- Nicolas Sabouret, LIMSI

Membres

- Emmanuel Adam, LAMIH
- Carole Adam, LIG
- Flavien Balbo, LHC
- Aurélie Beynier, LIP6
- Olivier Boissier, LHC
- Grégory Bonnet, GREYC
- Kei Brousmiche, SystemX
- Valérie Camps, IRIT
- Florent Carlier, CREN
- Anne-Cécile Caron, CRISTAL
- François Charpillat, LORIA
- Vincent Chevrier, LORIA
- Caroline Chopinaud, CRAFT
- Rémy Courdier, LIM
- Yves Demazeau, LIG
- Stéphane Galland, UTBM
- Marie-Pierre Gleizes, IRIT
- Emmanuelle Grislin, LAMIH
- Zahia Guessoum, LIP6
- Salima Hassas, LIRIS
- Cédric Herpson, LIP6
- Guillaume Hutzler, IBISC
- Jean-Paul Jamont, LCIS
- Elsy Kaddoum, IRIT
- Jean-Daniel Kant, LIP6
- Christophe Lang, LIUFC
- Domitile Lourdeaux, UTC
- Rene Mandiau, LAMIH
- Cristina Maniu, L2S
- Nicolas Marilleau, UMMISCO
- Philippe Mathieu, CRISTAL
- Bruno Mermet, GREYC
- Frederic Migeon, IRIT
- Maxime Morge, CRISTAL
- Gildas Morvan, LIG2A
- Jean-Pierre Muller, CIRAD
- Antoine Nongaillard, CRISTAL
- Michel Occello, LCIS
- Wassila Ouerdane, MICS
- Alexandre Pauchet, LITIS
- Denis Payet, LIM
- Sebastien Picault, CRISTAL
- Suzanne Pinson, LAMSADE
- Éric Platon, CosmosX
- Lilia Rejeb, ISG Tunis
- Valérie Renault, CREN
- Julien Saunier, LITIS
- Patrick Tailliandier, COGIT
- Mahdi Zargayouna, IFSTTAR

ReCoVac : Décision autonome de reprise de la conduite d'un véhicule autonome et connecté

K. Aguilar Alarcon¹, M-P. Gleizes¹, L. Caroux², E. Kaddoum¹, V. Camps¹, S. Combettes¹

¹ Université Toulouse III – Paul Sabatier, IRIT

² Université Toulouse II – Jean Jaurès, CLLE

prenom.premierNom-deuxiemeNom@irit.fr/@univ-tlse2.fr

Résumé

La société des ingénieurs de l'automobile a identifié 5 niveaux de conduite autonome dont le niveau 3 appelé "l'automatisation conditionnée" où le conducteur doit être capable de reprendre le volant en cas de nécessité. Plusieurs études en Sciences Humaines et Sociales (SHS) ont souligné l'existence d'un laps de temps important nécessaire pour la reprise du contrôle. Ce laps dépend de l'état mental, physique et émotionnel du conducteur ainsi que du contexte de conduite. L'objectif des recherches menées en collaboration avec des chercheurs en SHS vise à la mise en place d'un système adapté à chaque conducteur facilitant la reprise de contrôle. Cet article présente une première étape de ce travail qui correspond à la conception d'un système multi-agent pour superviser la conduite autonome et détecter des situations où la reprise est nécessaire.

Mots-clés

Véhicule autonome et connecté, reprise de contrôle, système multi-agent

Abstract

The Society of Automotive Engineers identified 5 levels of autonomous driving including level 3 called "conditioned automation" where the driver must be able to resume driving if necessary. In order to achieve this, several studies in the Human and Social Sciences (HSS) highlighted the existence of a significant period of time required to regain control. This time depends on the mental, physical and emotional state of the driver in addition to the driving context. The research work carried out in collaboration with SHS aims at the design of a system adapted to each driver to facilitate the regaining of control. This paper presents a first step concerning the design of multi-agent system to supervise autonomous driving and detect situations where regaining control is required.

Keywords

Autonomous and connected vehicle, retaking control, multi-agent system.

1 Contexte de l'étude

Si le concept de conduite autonome peut sembler vague, il existe pourtant une nomenclature officielle qui classe les

véhicules selon leur capacité à assister les conducteurs humains. La société internationale des ingénieurs de l'automobile a ainsi identifié 5 niveaux de conduite autonome. Dans ce travail, nous considérons les véhicules de niveau 3 également appelé "l'automatisation conditionnée". Dans certaines situations prédéfinies, comme de longues lignes droites avec marquage au sol bien visible, le véhicule peut automatiquement gérer sa trajectoire et sa vitesse, éviter les obstacles, etc. Le conducteur humain n'est plus obligé de garder les yeux sur la route, mais il doit être capable de reprendre le volant en cas de nécessité. Les procédures de reprise en main sont considérées comme des périodes particulièrement critiques où le niveau de sécurité dépend de la performance combinée de l'humain et du système d'automatisation. En effet, les études en Sciences Humaines et Sociales (SHS) [24, 1, 9, 22] ont montré qu'un laps de temps important est nécessaire pour permettre au conducteur de reprendre le contrôle de la conduite. Ce laps de temps dépend de l'état mental, physique et émotionnel du conducteur ainsi que du contexte de conduite et notamment de la situation rencontrée. De plus, les experts du domaine estiment qu'avec l'introduction du véhicule autonome, le conducteur perdra en expertise, ce qui implique une croissance forte de ce laps de temps pour assurer une transition sûre et efficace.

Le niveau 3 de conduite autonome est le sujet de plusieurs projets de recherche souvent centrés sur les facteurs humains pour comprendre et modéliser la réaction humaine afin de mieux la prendre en compte. Néanmoins, la multitude des profils de conducteurs existants rend cette modélisation difficile à mettre en place. De ce fait, le déploiement de systèmes de conduite autonome de niveau 3 se limite à des situations bien spécifiées.

Dans le cadre du projet interdisciplinaire ReCoVac (Reprise du Contrôle de la conduite au sein d'un Véhicule Autonome et Connecté) mené en collaboration avec des chercheurs du laboratoire CLLE¹ de l'Université de Toulouse - Jean Jaurès, nous nous intéressons à la mise en place d'un système adapté à chaque conducteur permettant une reprise de contrôle sûre. Les expérimentations menées par le CLLE s'intéressent à l'étude de l'attention de l'humain notamment dans le cadre de la conduite autonome. Elles ont

1. Laboratoire de Cognition, Langues, Langage, Ergonomie <https://clle.univ-tlse2.fr/>

souligné que plusieurs facteurs influent sur l'attention mais qu'il est difficile d'étudier l'impact de leur combinaison. L'originalité du travail mené au sein de ce projet concerne l'intégration des facteurs humains dans un système de supervision afin que ce dernier soit adapté et accepté par son utilisateur final. Ainsi, le projet ReCoVac se décompose en trois étapes :

1. la modélisation et la conception d'un système multi-agent capable de superviser la conduite autonome et d'anticiper les situations anormales où la reprise du contrôle est nécessaire,
2. l'étude de l'impact des différents facteurs humains, environnementaux, et du véhicule sur la capacité de chaque conducteur à gérer une situation,
3. l'intégration des facteurs humains dans le système de supervision afin de mieux anticiper et expliquer les situations à risque et ainsi permettre une reprise efficace et sûre du contrôle de la conduite.

Cet article présente les travaux menés dans le cadre de la première étape et notamment ceux permettant d'apprendre à détecter les situations anormales afin de pouvoir par la suite intégrer les facteurs humains. Le travail présenté dans cet article ne traite pas des contraintes de temps réels et des certifications de sécurité nécessaires pour ce type de système, il se concentre sur la capacité du système à détecter les situations anormales. La section 2 présente un état de l'art sur les systèmes de supervision et de détection d'anomalies et montre l'intérêt d'utiliser des systèmes multi-agents. La section 3 détaille le modèle multi-agent ADORSiDe pour la détection de situations où la reprise de contrôle est nécessaire. Puis, la section 4 expose les résultats préliminaires obtenus, avant de conclure et proposer des perspectives en section 5.

2 État de l'art

Dans cette section, nous analysons et discutons les travaux ayant trait à la supervision, et au contrôle de système autonomes, ainsi qu'à la détection d'anomalies.

La supervision est communément définie comme l'action d'exercer un contrôle, et de surveiller en temps réel l'activité, le comportement et les informations des entités, dans le but d'interpréter les perceptions, de détecter des actions et de prévoir des événements [6, 25]. Dans ce travail, nous nous intéressons aux systèmes informatiques de supervision qui analysent les comportements observés pour détecter des anomalies, menaces, comportements à risques, mais aussi capables d'apprendre à identifier de nouvelles anomalies, comme le traite [2].

Le travail [25] identifie quatre principaux processus lors de la supervision : la détection et la reconnaissance d'entités en évolution, le suivi, l'analyse de leur comportement et la récupération. Pour la détection et la reconnaissance, la répartition des dispositifs de détection et la fusion des données perçues individuellement par chaque dispositif sont importantes. Les systèmes de supervision ont été appliqués à la circulation automobile pour des systèmes de vision autonomes pour le suivi de véhicules, de conducteurs et de

personnes [18, 26] et dans la surveillance des chemins de fer [19].

La détection et le suivi d'une entité conduisent souvent à une analyse comportementale pour détecter des anomalies. Certains travaux modélisent des comportements anormaux ou non autorisés dans leurs systèmes, notamment pour identifier les menaces maritimes asymétriques dans les ports et les voies navigables [23, 12]. A l'inverse, certains travaux modélisent les comportements autorisés et lèvent une alerte pour les comportements anormaux comme dans [14] pour la détection de conduites anormales à partir de la vitesse de conduite et du mouvement du véhicule.

Enfin, il existe des systèmes de supervision qui utilisent différentes techniques d'apprentissage afin de comprendre les comportements observés ; nous citerons par exemple [20] qui a conçu un système multi-agent pour détecter des intrusions dans les réseaux informatiques.

Par ailleurs, la diversité de constructeurs automobiles induit une multitude de systèmes de conduite autonomes. Il est donc nécessaire d'avoir un système de supervision générique qui soit capable de s'adapter au système observé. Ainsi, le système doit être capable d'apprendre par la collecte d'expériences et de prendre des décisions. Le modèle d'apprentissage fonctionne en apprenant à détecter automatiquement des objets, des situations complexes et des modèles. Plus, il dispose de données, plus le système est précis. Le système apprend par la répétition, en recevant continuellement des données d'entrée et en optimisant sa capacité à faire des prédictions sur la signification de ces données et sur ce qu'il faut en faire [21]. Le travail présenté ici concerne l'étape 1 du projet ReCoVac et consiste à développer un mécanisme d'apprentissage permettant de détecter des situations anormales car le conducteur autonome n'est plus en mesure de gérer la conduite ; un changement de contrôle de conduite devient alors nécessaire.

L'apprentissage artificiel (Machine Learning : ML) concerne le développement et la mise en œuvre de méthodes automatiques qui permettent à un système d'apprendre à partir d'informations non structurées. Les systèmes qui utilisent l'apprentissage automatique peuvent accomplir des tâches qu'ils ne peuvent pas traiter à l'aide d'algorithmes plus courants [16]. Il existe de nombreuses techniques de ML comme les règles d'association, les machines à vecteur de support, les réseaux bayésiens, etc. Parmi elles, deux approches sont adaptées pour l'apprentissage dans le contexte de systèmes complexes comme les véhicules autonomes : 1) les approches à base de réseaux neuronaux, et 2) l'approche constructiviste.

L'approche par réseaux de neurones s'effectue à l'aide d'une fonction qui admet un certain nombre d'entrées et de poids. Les poids sont liés à chacune des entrées et les pondèrent. Cette technique a été appliquée à différents travaux portant sur des véhicules autonomes : 1) méthode évolutive pour la création d'un contrôleur de véhicule terrestre autonome, pour laquelle ont été élaborées des techniques afin d'intégrer la recherche évolutive et la rétro-propagation d'erreurs [3] ; 2) mise en place de deux sous-systèmes dans le système d'aide à la conduite "Intelligent Stop & Go". Le

premier sous-système détecte les véhicules et estime la position du véhicule sur les autoroutes. Le second permet la reconnaissance de piétons [28]. Les réseaux neuronaux ont de bons résultats dans de nombreux domaines ce qui en fait des candidats intéressants. Cependant, le temps d'apprentissage est encore très important, ce qui empêche l'utilisation de ces approches pour obtenir un apprentissage temps réel sans infrastructure informatique efficace [15]. De plus, ce sont des systèmes difficiles à paramétrer, en raison de la multitude de variables, de paramètres différents et d'architectures à choisir [4]. Ce n'est donc pas une technique que nous avons retenue pour atteindre nos objectifs.

L'approche constructiviste cherche à imiter le mécanisme de construction des connaissances chez l'être humain à travers des interactions avec le monde réel. Les concepts acquis précédemment servent de base pour assimiler et interpréter de nouvelles expériences, et ces anciens concepts sont ajustés en fonction de ces observations [8]. Cette technique d'apprentissage a été appliquée à différents travaux : 1) ensemble de mécanismes pour améliorer les capacités d'adaptation des schémas dans le mécanisme d'apprentissage anticipatif constructif (CALM) [17]; 2) mise en place d'un système d'apprentissage multi-agent basé sur des schémas, conçu pour l'intelligence ambiante et adapté aux environnements continus [13]; 3) système d'apprentissage constructiviste, également basé sur un système multi-agent pour réaliser une activité de contrôle du trafic [11]. Les approches constructivistes offrent des caractéristiques intéressantes, car elles permettent d'ajouter facilement de nouvelles sources de données lors de l'exécution. Elles sont conçues pour apprendre en explorant l'environnement. De plus, l'apprentissage réalisé est indépendant de la tâche à effectuer. Cependant, la principale limitation est liée à l'explosion combinatoire lorsqu'elle doit passer à l'échelle. De plus, les schémas dans leur structure d'origine ne sont pas adaptés aux données continues [15]. Même si la technique d'apprentissage peut satisfaire nos besoins, le passage à l'échelle est un élément important à prendre en compte dans les systèmes de véhicules autonomes.

Pour résoudre notre problème nous avons choisi les systèmes multi-agents adaptatifs (AMAS) qui ont montré leur adéquation pour résoudre des problèmes d'apprentissage adaptatif en temps réel dans des conditions similaires au problème étudié. Un AMAS comporte des agents qui interagissent en coopération. Par leurs interactions, le système s'auto-organise et atteint sa fonction objectif [10]. La coopération est le mécanisme fondamental d'auto-organisation et d'auto-adaptation pour la conception des AMAS.

L'approche par AMAS possède des propriétés intéressantes pour aborder des systèmes complexes : 1) l'absence d'un algorithme connu qui répond au problème; 2) un grand nombre d'entités corrélées et coopératives qui interagissent entre elles; 3) un environnement dynamique, évolutif, ouvert et partiellement observable; 4) la nature distribuée et décentralisée des systèmes obtenus; 5) la capacité de ces systèmes à apprendre et à s'adapter tout au long de la vie du système.

Cette approche a été appliquée avec succès à une grande variété d'applications, notamment pour l'apprentissage par démonstrations [27], le contrôle d'un système complexe [5] et la détection d'anomalies en surveillance maritime [6]. Ces systèmes ont montré leur capacité à réagir aux bruits, aux changements brusques et aux défaillances. Pour ces raisons nous avons utilisé ce paradigme pour concevoir le système ADORSiDe.

3 Contributions

Dans cette section, nous présentons le système *Autonomous Driver Observation for Risky Situation DEtection* (ADORSiDe) mis en place pour superviser la conduite autonome et anticiper les situations où la reprise de contrôle est nécessaire. Ce système se base sur l'approche par AMAS [10] et s'inspire du modèle Agent Qualas [7]. Son objectif est d'observer le système de conduite autonome d'un véhicule afin d'en extraire par apprentissage un modèle et ainsi pouvoir anticiper l'occurrence d'anomalies.

Le fonctionnement d'un véhicule totalement autonome se décompose en 4 étapes comme c'est le cas pour la voiture autonome du CEA² :

1. la perception des informations permettant de capter toutes les informations extérieures utiles à la conduite;
2. la fusion et l'analyse des informations récoltées;
3. la prise de décision des actions à mener;
4. l'activation des commandes.

Dans notre travail, nous nous basons sur ce cycle de fonctionnement et nous cherchons à : 1) apprendre à prédire au moment de la prise de décision noté t l'état de l'environnement après l'activation des commandes noté $t+1$ en observant le conducteur autonome; 2) évaluer la différence entre l'état prédit de l'environnement et l'état de l'environnement souhaité par le conducteur autonome afin de juger de la capacité de ce dernier à gérer la situation.

3.1 Spécification du problème

Pour réaliser l'ensemble des activités précédentes, le problème étudié est décrit par :

- un **conducteur autonome** représentant le système de conduite autonome
- un **contexte de conduite**

$DC^t = \{EV_i^t\}$ définissant l'état perçu de l'environnement de conduite à l'instant t sous forme d'un ensemble de variables environnementales $\{EV_i^t\}$ avec $i = 1..n$. Ces variables décrivent les états internes (nombre de passagers, état du conducteur, etc.) et externes (nombre de véhicules, virage à droite, nombre de piétons, etc.) du véhicule. C'est en fonction du contexte de conduite DC^t et de l'objectif à atteindre que le conducteur autonome décide d'un ensemble d'actions $SoA^t = \{A_k^t\}$ à réaliser.

2. <http://www.cea.fr/comprendre/Pages/nouvelles-technologies/essentiel-sur-voiture-autonome.aspx>

- un **ensemble d’actions**
 $SoA^t = \{A_k^t\}$ avec $k = 1..m$ décrivant les actions (tourner, accélérer, décélérer, etc.) décidées et réalisées par le conducteur autonome à t en fonction de l’environnement de conduite perçu.
- le **contexte de conduite intentionnel**
 $(IDC^{t+1} = \{IEV_i^{t+1}\})$ avec $i = 1..n$ donné par le système de conduite autonome. Ce contexte représente l’état des variables environnementales que souhaite obtenir le conducteur autonome à $t + 1$ étant donné ses objectifs locaux et sa perception du contexte de conduite DC^t à t .
- le **contexte de conduite prédit**
 $(PDC^{t+1} = \{PEV_i^{t+1}\})$ avec $i = 1..n$ calculé par ADORSiDe à partir de l’observation du conducteur autonome. PDC^{t+1} est donc l’état de conduite estimé par le système de supervision en supposant que le conducteur réalise l’ensemble d’actions décidées SoA^t .
- le **contexte de conduite réel**
 $(RDC^{t+1} = \{REV_i^{t+1}\})$ avec $i = 1..n$. Ce contexte représente l’état perçu de l’environnement à $t + 1$ après l’exécution de l’ensemble des actions SoA^t dans le contexte de conduite DC^t . RDC^{t+1} est utilisé par ADORSiDe lors de l’apprentissage de l’effet des actions réalisées sur les différentes variables environnementales.

3.2 ADORSiDe : Autonomous Driver Observation for Risky Situation Detection

Ce système multi-agent est composé d’agents de prédiction des variables environnementales $EVPA$ et d’agents variables VA . A chaque variable environnementale $\{EV_i\}$, ADORSiDe associe un agent de prédiction $EVPA_i$ dont le rôle est de déterminer l’influence des autres variables environnementales et des états des actions sur sa variable. Ainsi, à chaque agent de prédiction de variable est associé à un agent variable VA_l avec $l = 1..n+m-1$ pour chaque variable environnementale $\{EV_j\}$ ($i \neq j, j = 1..n$) et chaque action $\{A_k\}$ ($k = 1..m$). Chaque agent variable a pour objectif de donner à l’agent de prédiction l’influence de sa variable environnementale ou de son action pour que ce dernier puisse prédire la valeur de sa variable environnementale.

Le fonctionnement général des agents dans ADORSiDe suit l’algorithme 1. Pour chaque contexte de conduite $DC^t = \{EV_i^t\}$ et ensemble d’actions $SoA^t = \{A_k^t\}$, chaque agent de prédiction $EVPA_i$ perçoit l’état des variables environnementales et des actions, ainsi que la valeur réelle de sa variable $\{REV_i^{t+1}\}$. Il communique ces valeurs aux agents variables VA_l qui le composent et leur demandent des ajustements tant que la valeur qu’il estime ne correspond pas à la valeur réelle attendue (avec un pourcentage d’erreur à atteindre E entre les deux valeurs). Les agents variables coopèrent pour aider leur $EVPA_i$ à estimer sa valeur. A chaque contexte de conduite DC^t , ADORSiDe exécute une phase d’apprentissage qui se traduit par plusieurs cycle de vie (perception, décision, action)

des agents $EVPA$ et VA .

Algorithm 1 Fonctionnement général des agents d’ADORSiDe

```

1: for all  $DC^t$  do
2:   for all  $EVPA_i$  do
3:     Percevoir  $REV_i^{t+1}$ ;
4:     Percevoir les  $EV_j^t$  avec  $j = 1..n, j \neq i$  et les  $A_k^t, k = 1..m$  et les communiquer aux  $VA_l$  avec  $l = 1..n+m-1$ ;
5:     Recevoir les influences,  $w_l * EV_j$  ou  $w_l * A_k$ , estimées par chaque  $VA_l$ ;
6:     Calculer  $PEV_i^{t+1} = \sum_{l,j=1}^{n-1} (w_l * EV_j^t) + \sum_{l=n, k=1}^{n+m-1, m} w_l * A_k^t$ 
7:     while  $REV_i^{t+1} \neq PEV_i^{t+1}$  do
8:       Demander aux  $VA_l$  d’ajuster leur influence en fonction de l’écart calculé ( $REV_i^{t+1} - PEV_i^{t+1}$ ) et les anciennes perceptions de l’agent
9:     end while
10:   end for
11:   Intégrer  $DC^t$  dans la base de connaissances de l’agent  $EVPA_i$ 
12: end for

```

3.3 Les agents d’ADORSiDe

Cette section décrit les deux types d’agents du système : l’agent prédiction de variable environnementale ($EVPA$) et l’agent variable (VA).

3.3.1 Agent prédiction de variable environnementale

Un agent $EVPA_i$ est responsables du modèle de prédiction de la variable environnementale PEV_i^{t+1} et cherche à minimiser sa différence avec sa valeur réelle REV_i^{t+1} . Pour atteindre son objectif, il coopère avec les agents variables VA_l le composant en leur fournissant les feedbacks nécessaires pour améliorer l’évaluation de leur influence. A t , l’agent $EVPA_i$ perçoit la valeur réelle de sa variable environnementale REV_i^{t+1} à $t + 1$ ainsi que les informations du contexte de conduite DC^t constitué de l’ensemble des autres variables environnementales $\{EV_j^t\}$ ($i \neq j$) et des actions $\{A_k^t\}$ qu’il communique à ses agents variables VA_l (voir Figure 1).

A la réception des influences calculées par les VA_l , l’agent $EVPA_i$ prédit par approximation linéaire la valeur de sa variable :

$$PEV_i^{t+1} = \sum_{l=1}^{n+m-1} (w_l * VA_l) \quad (1)$$

Ensuite, l’agent $EVPA_i$ évalue pour les différents contextes de conduite déjà rencontrés, les convergences des erreurs en calculant pour chaque contexte de conduite l’écart (Δ_i^{t+1}) entre sa valeur réelle dans ce contexte et la valeur prédite. (Δ_i^{t+1}) est comparée à un pourcentage d’erreur à atteindre (E).

$$|\Delta_i^{t+1} = REV_i^{t+1} - PEV_i^{t+1}| < E \quad (2)$$

Tant que la convergence de l'erreur maximale est supérieure à E , l'agent $EVPA_i$ demande aux agents VA_l actifs dans le contexte de conduite donnant cette erreur maximale d'actualiser leur poids, en leur fournissant toutes les erreurs pour les contextes dans lesquels ils sont impliqués.

Si la convergence de l'erreur maximale est proche de E , cela indique qu'une solution collective est trouvée.

Une phase d'apprentissage se termine si la convergence est inférieure ou égale à E , ou si tous les agents VA_l signalent des cycles dans leurs estimations et/ou l'agent $EVPA_i$ détecte un minimum local dans la fonction de prédiction du contexte de conduite avec l'erreur maximale. Ce minimum local est détecté quand l'agent $EVPA_i$ n'améliore pas l'erreur de prédiction malgré un certain nombre de demandes d'ajustements adressées aux agents VA_l .

Une phase d'apprentissage se conclut par l'ajout du nouveau contexte de conduite dans la base de connaissances de l'agent $EVPA_i$ qui constitue l'historique des différentes situations rencontrées. Cette base est utilisée par l'agent $EVPA_i$ pour s'assurer que les ajustements réalisés par les agents variables ne détériorent pas les situations déjà rencontrées.

3.3.2 Agent variable

L'agent variable VA_l est la représentation d'une variable environnementale (EV_j) présente dans le contexte de conduite, et/ou l'état d'une action de conduite A_k que le conducteur peut effectuer sur le véhicule. Il est associé et interagit avec l'agent $EVPA$. Son objectif est d'apprendre son poids et il coopère avec l'agent $EVPA_i$ pour résoudre les erreurs observées.

L'agent variable VA_l perçoit la valeur de sa variable environnementale ou l'état de l'action qu'il représente dans le contexte de conduite et y associe un paramètre de pondération, appelé poids, (w_l), représentant l'estimation de l'influence de sa variable dans le modèle de prédiction.

L'agent variable VA_l reçoit le feedback de l'agent $EVPA_i$, contenant les erreurs d'estimation (qui peuvent être positives ou négatives) pour tous les contextes de conduite dans lesquels il est actif. À partir de ces erreurs, l'agent VA_l détermine le contexte de conduite avec la plus grande erreur positive (c.-à-d. la surestimation maximale) $MaxE_{+l}$, et le contexte de conduite avec la plus grande erreur négative (c.-à-d. la sous-estimation maximale) $MaxE_{-l}$; son objectif est alors de minimiser ces deux situations les plus critiques. Par conséquent, l'agent variable VA_l cherche à équilibrer $MaxE_{+l}$ et $MaxE_{-l}$ en calculant une valeur α_l (équation (3)) prenant en compte la participation des autres agents variables également responsables de ces erreurs.

$$\alpha_l = \frac{-(MaxE_{+l} + MaxE_{-l})}{2 * nbVarAc_{Max(MaxE_{+l}, MaxE_{-l})}} \quad (3)$$

avec $nbVarAc$, le nombre de variables actives dans le contexte de conduite avec la plus grande erreur.

Puisque l'agent variable VA_l n'est pas le seul agent qui cherche à minimiser l'erreur (Δ_i^{t+1}), il calcule sa valeur

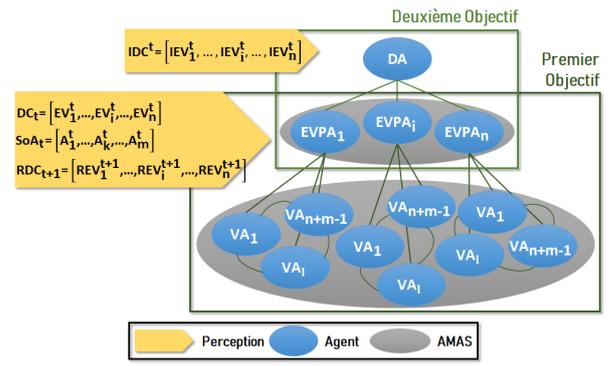


FIGURE 1 – Modèle multi-agent proposé pour ADORSiDe

d'importance dans la prédiction (équation (4)).

$$importance_l = \left| \frac{w_l * AV_l}{Max(MaxE_{+l}, MaxE_{-l})} \right| \quad (4)$$

Finalement, l'agent variable VA_l modifie son estimation de poids (w_l) à chaque étape de l'apprentissage en ajoutant α_l (équation (5)) en fonction de son $importance_l$, et crée un historique de ses estimations précédentes pour déterminer si localement il a ou non un comportement cyclique. Ce comportement est détecté quand sa valeur d'équilibre (α_l) n'est pas suffisamment importante pour générer un impact sur l'erreur de prédiction pour les situations dans lesquelles il participe, indiquant ainsi l'inutilité de modifier son poids. Ces informations sont transmises à l'agent $EVPA_i$ qui décide de continuer la phase d'apprentissage ou de l'arrêter.

$$(w'_l) = w_l + (\alpha_l * importance_l) \quad (5)$$

La figure 1 montre l'association de chaque agent $EVPA_i$ à autant d'agents variables VA_l qu'il y a de variables environnementales et d'état des actions.

3.4 Détection de situations anormales

Une fois que le modèle de chaque variable environnementale est appris, il est utilisé dans la deuxième étape qui consiste à évaluer la capacité du conducteur autonome à gérer la situation courante. Pour ce faire, ADORSiDe calcule à t le contexte de conduite prédit (PDC^{t+1}) et évalue la distance avec le contexte de conduite intentionnel (IDC^{t+1}) du conducteur autonome. Pour comparer ces deux contextes, la distance euclidienne est utilisée :

$$d(IDC^{t+1}, PDC^{t+1}) = \sqrt{\sum_{i=1}^n (|IEV_i^{t+1} - PEV_i^{t+1}|)} \quad (6)$$

Puisque chaque variable environnementale a individuellement un effet différent sur le comportement du conducteur autonome et que cela peut affecter la conduite, ADORSiDe associe une valeur d'importance imp_i à chaque variable du processus de conduite. En ajustant la métrique de distance comme suit :

$$d(IDC^{t+1}, PDC^{t+1}) = \sqrt{\sum_{i=1}^n (|IEV_i^t - PEV_i^{t+1}|) * imp_i} \quad (7)$$

Afin de décider de la capacité de conduite du conducteur autonome, cette distance est comparée à un seuil de tolérance. Le seuil permet de détecter si le conducteur autonome est capable de réagir de la manière attendue au contexte de conduite suivant, ou s'il s'agit d'une situation anormale que le conducteur autonome n'est pas en mesure de gérer et qu'un changement de contrôle de conduite est donc nécessaire. L'alerte correspondant à une anomalie est déclenchée lorsque la distance dépasse ce seuil.

$$d(IDC^{t+1}, PDC^{t+1}) > Threshold \quad (8)$$

Étant donné, que tous les contextes de conduite sont différents et que leur niveau de criticité est variable, certaines situations nécessitent une capacité de conduite plus ou moins grande pour les gérer, c'est pourquoi la valeur du seuil doit s'adapter au type de situations. Pour cette adaptation, une valeur de récompense (r^{t-1}) est ajoutée au seuil, représentant l'évaluation de la qualité de la décision prise précédemment. La décision est considérée comme bonne si le conducteur autonome respecte les règles de conduite, n'atteint pas une situation dangereuse et/ou est satisfait; elle est considérée comme mauvaise dans le cas contraire.

Pour évaluer la décision prise par ADORSiDe, celle-ci est envoyée à l'*Oracle* (voir section 4), qui fournit un retour sur la qualité de la décision : (1) **Faux positif**, le système détecte une situation anormale non classée comme anormale dans l'*Oracle*. (2) **Faux négatif**, l'*Oracle* identifie que le conducteur autonome n'est pas en mesure de gérer la situation alors que le système n'active pas l'alerte d'anomalie. (3) **Vrai positif**, une alerte d'anomalie a été générée et l'*Oracle* confirme l'existence d'une situation anormale. Et, (4) **vrai négatif**, le système détermine que le conducteur autonome est en mesure de poursuivre l'activité de conduite et l'*Oracle* confirme. Selon le retour, la valeur de la récompense (r^{t-1}) est différente.

$$Threshold^t = Threshold^{t-1} * r^{t-1} \quad (9)$$

Dans cette phase, le fonctionnement général est décrit par l'algorithme 2. ADORSiDe associe un agent de supervision au conducteur autonome (*DSA*) (qui dans cette version du système n'a pas encore de comportement coopératif), dont le rôle est d'évaluer la capacité du conducteur autonome à gérer un contexte de conduite. L'agent *DSA* perçoit les valeurs prédites des agents de prédiction des variables environnementales $EVPA_i$ avec $i = 1..n$, ainsi que les valeurs intentionnelles IEV_i^{t+1} du conducteur autonome et estime l'écart existant entre le contexte de conduite prédit PDC^{t+1} et le contexte intentionnel IDC^{t+1} . Ensuite, il compare cet écart avec le seuil (voir équation 8); s'il est supérieur au seuil l'agent décide que le conducteur n'est pas en mesure de continuer la conduite.

L'agent *DSA* échange avec l'*Oracle* pour évaluer la qualité de sa décision et ajuste le seuil en fonction (voir équation 9).

Algorithm 2 Fonctionnement général d'ADORSiDe pour l'évaluation de la capacité du conducteur autonome à gérer une situation

- 1: **for all** DC^t **do**
 - 2: Recevoir les valeurs prédites de chaque $EVPA_i$, avec $i = 1..n$;
 - 3: Percevoir les valeurs intentionnelles de chaque IEV_i^{t+1} avec $i = 1..n$;
 - 4: Calculer $d(IDC^{t+1}, PDC^{t+1}) = \sqrt{\sum_{i=1}^n (IEV_i^{t+1} - PEV_i^{t+1}) * imp_i}$
 - 5: **if** $d(IDC^{t+1}, PDC^{t+1}) > Threshold$ **then**
 - 6: Détecter une situation et demander un changement de contrôle.
 - 7: **else**
 - 8: L'activité de conduite autonome peut se poursuivre.
 - 9: **end if**
 - 10: Donner les variables les plus impliquées dans la décision, c.-à-d. EV_i avec les plus grandes valeurs de imp_i
 - 11: Demander à l'oracle l'évaluation de la situation.
 - 12: $Threshold^t = Threshold^{t-1} * r^{t-1}$
 - 13: **end for**
-

La figure 1 illustre le flux de données dans ADORSiDe. Pour évaluer la capacité du conducteur autonome à gérer une situation, l'agent *DSA* est associé à autant d'agents *EVPA* que de variables environnementales dans le contexte de conduite. Les agents *EVPA* envoient la valeur de prédiction de la variable environnementale qu'ils représentent à *DSA*, qui détermine si le conducteur autonome atteint ou non ses objectifs.

4 Expérimentations

Afin d'étudier les performances d'ADORSiDe, un environnement synthétique a été conçu. Cette section décrit d'abord l'environnement synthétique, puis le processus expérimental, et enfin présente les résultats obtenus.

4.1 Générateur synthétique de scénarios

Pour simuler un environnement de test du système, nous utilisons un générateur synthétique. Son utilisation est motivée par la difficulté d'accès aux données réelles et personnelles (RGPD) et permet d'évaluer ADORSiDe sur une grande diversité de cas, même avec des données qui pourraient être bruitées. Faire ce traitement en conditions réelles impliquerait un coût économique élevé, difficile à contrôler et qui pourrait amener à des situations dangereuses.

Le générateur défini permet de jouer différents scénarios représentant l'évolution et la variété des processus réels de conduite.

Comme le montre la figure 2, un scénario est décrit par un ensemble de contextes de conduite (DC_s^t) définis par

des variables environnementales (EV_i^t avec $i = 1..n$), et des états d'actions (A_k^t avec $k = 1..m$). Le nombre de contextes de conduite s , de variables d'environnement n et d'états d'actions m sont les paramètres de l'expérience. A noter que dans un cas réel, le nombre de variables environnementales est au moins égal à celui de l'environnement réel. Une étude, menée avec les partenaire SHS du projet, a pour objectif d'identifier les indicateurs qui caractérisent l'environnement de l'activité de conduite. Ces indicateurs se séparent en deux ensembles : le premier décrit l'environnement dans lequel se trouve le véhicule, le second décrit l'état des actions (valeur de vitesse, pression sur l'embrayage, etc.). Afin de simplifier leur génération, on suppose que ces valeurs passent par un processus de normalisation et sont représentées dans une plage de valeurs définies (dans notre cas de $[0, 10]$). Ainsi l'*Oracle* initialise de manière aléatoire les valeurs des n variables environnementales entre deux bornes $[value_{min}, value_{max}]$. La génération des valeurs de chacune des m variables d'action par le générateur synthétique est fonction d'un sous-ensemble de variables environnementales présentes dans le contexte et est réalisé de manière aléatoire. Suite à cette étape d'initialisation, deux fonctions différentes sont définies pour simuler une corrélation des données. La première calcule l'état intentionnel de l'environnement souhaité par le conducteur autonome (IEV_i^{t+1}) après application des actions désirées. La deuxième calcule l'état réel de cet environnement (REV_i^{t+1}). Comme en réalité, le conducteur autonome peut ne pas contrôler parfaitement le véhicule, la deuxième fonction «ressemble» à la première avec quelques variations sur les variables du contexte de conduite. Ces variations sont déterministes pour que le système apprenne. L'écart entre les valeurs de (IEV_i^{t+1}) et (REV_i^{t+1}) est utilisé pour évaluer la capacité de conduite du conducteur autonome. Nous avons utilisé la métrique de *troisième quartile d'une mesure d'ensemble de données* pour classer les contextes de conduite (comme normales ou anormales).

Scénario

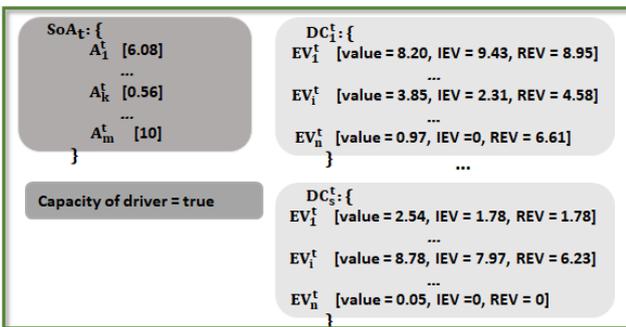


FIGURE 2 – Scénario d'un processus de conduite autonome

4.2 Résultats préliminaires

Cette section présente les résultats préliminaires concernant la prédiction de la valeur d'une variable environnementale. Nous évaluons 1) la capacité d'ADORSiDe à

ajuster dynamiquement ses poids en fonction de l'erreur de prédiction, en atteignant un certain niveau de précision, et 2) l'influence du nombre de contextes de conduite et de variables environnementales dans le nombre de phases d'apprentissage nécessaires pour atteindre le niveau de précision requis.

Dans cette optique, 7 scénarios de tests ont été générés avec le générateur de scénarios avec comme paramètres (voir Tableau 1) $s = 10, 30, 50, 70$, $n = 10, 30, 50, 70$, $m = 10$, $value_{min} = 0$ et $value_{max} = 10$. Pour chaque expérimentation, la valeur de précision à atteindre (E) entre la valeur estimée de la variable et sa valeur réelle attendue, est calculée.

D'abord, nous proposons d'évaluer le comportement d'ADORSiDe au cours du processus de résolution. La figure 3 présente la convergence de la plus grande erreur positive (courbe rouge) et de la plus grande erreur négative (courbe bleue) pour le troisième scénario ($s = 50$, $n = 10$). Cette figure illustre le processus de minimisation de l'erreur de prédiction de la variable environnementale par l'agent *EVPA* aidé par le comportement coopératif des agents *VAI*. Au fur et à mesure que le processus progresse, de nouveaux contextes de conduite sont ajoutés, qui peuvent être plus critiques (erreur positive ou négative plus importante) que les contextes de conduites précédents, ce qui explique certaines augmentations ou diminutions rapides de la courbe de l'erreur de prédiction.

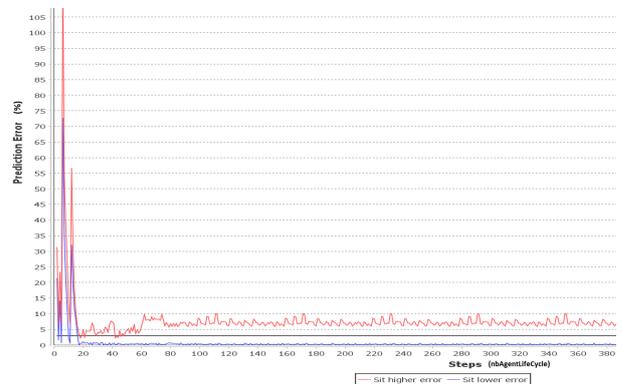


FIGURE 3 – Évolution de la criticité de la plus grande erreur positive et de la plus grande erreur négative pour le troisième scénario ($s = 50$, $n = 10$)

Pour ce scénario, les contextes de conduite proposés en entrée ont permis au système de réduire la plus grande erreur négative ($MaxE- = 0,727$) (courbe bleue), et atteindre la précision désirée (courbe noire). Cependant, la plus grande erreur positive (courbe rouge) reste légèrement supérieure à la valeur souhaitée, ce qui signifie que les agents d'ADORSiDe n'ont pas pu trouver les poids nécessaires pour satisfaire tous les contextes de conduite. Ces résultats peuvent être expliqués par le fait que le nombre de contextes en entrée du système n'est pas suffisant pour que les *VAI* aient assez d'informations pour contraindre leur poids respectifs.

Pour étudier la convergence du système, nous avons exé-

Paramètres du générateur	% de DC_t avec erreur $< E$	MaxE+	MaxE-	Raison d'arrêt de la dernière phase d'apprentissage
s=10, n=10, m=10	40	18,006	0,764	Comportement cyclique des VA_l
s=30, n=10, m=10	16,667	12,656	0,356	Erreur d'estimation $< E$
s=50, n=10, m=10	6	15,306	0,727	Erreur d'estimation $< E$
s=70, n=10, m=10	5,714	17,132	1,027	nb cycles apprentissage $>$ maximum
s=10, n=30, m=10	20	29,355	1,139	Comportement cyclique des VA_l
s=10, n=50, m=10	20	39,180	1,158	Comportement cyclique des VA_l
s=10, n=70, m=10	20	33,777	1,214	Comportement cyclique des VA_l

TABLE 1 – Résultats des moyennes des 10 expériences effectuées pour chaque scénario.

cuté 10 fois chaque scénario. La table 1 montre les moyennes des résultats obtenus selon 4 critères : le pourcentage de contextes de conduite DC_t ayant une erreur inférieure à la valeur de précision à atteindre (E , fixée ici à 3%), les valeurs de la plus grande erreur positive ($MaxE+$) et négative ($MaxE-$) ainsi que la raison ayant amenée l'arrêt de la dernière phase d'apprentissage liée à la prise en compte du dernier contexte de conduite. Cette raison peut être : 1) les agents VA_l ont détecté un comportement cyclique, 2) l'agent $EVPA$ a détecté un minimum local dans la prédiction (*i.e.* les ajustements effectués par les VA_l n'améliorent plus la situation), et/ou 3) la convergence du contexte de conduite actuellement traité a atteint la précision.

Cette table montre que le pourcentage de DC_t ayant une erreur inférieure à E est faible ce qui signifie que pour la plupart des DC_t , le système n'arrive pas à réduire l'erreur d'estimation en dessous de la précision souhaitée. De plus, nous notons que l'erreur de la situation la plus critique positive ($MaxE+$) est plus importante pour n élevé, car les agents VA ont plus de difficulté à ajuster significativement leur paramètre de poids, ce qui est également souligné par la détection du comportement cyclique des agents VA_l . Néanmoins, nous soulignons que la plus grande erreur négative ($MaxE-$) atteint la précision souhaitée. Des expérimentations plus approfondies sont nécessaires pour comprendre les raisons amenant $MaxE-$ à converger au détriment de $MaxE+$.

Ensuite, nous proposons d'évaluer la sensibilité d'ADORSiDe à la variation des paramètres s , et n . Pour cela, nous divisons l'expérimentation en deux : partie A) l'analyse de l'influence du nombre de contextes de conduite s sur le nombre de cycles de vie des agents pour la convergence du système (utilisation des 4 premiers scénarios de la table 1); et partie B) l'étude de l'influence du nombre de variables environnementales n sur le nombre de cycles de vie pour la convergence du système (utilisation du premier et des 3 derniers scénarios de la table 1).

La figure 4 présente l'influence du nombre de contextes de conduite s sur le nombre de cycles de vie des agents nécessaires pour atteindre la convergence du système. La figure montre que la relation entre les deux paramètres étudiés suit une tendance légèrement plus polynomiale (coefficient = 0,9831) que linéaire (coefficient = 0,9776). La

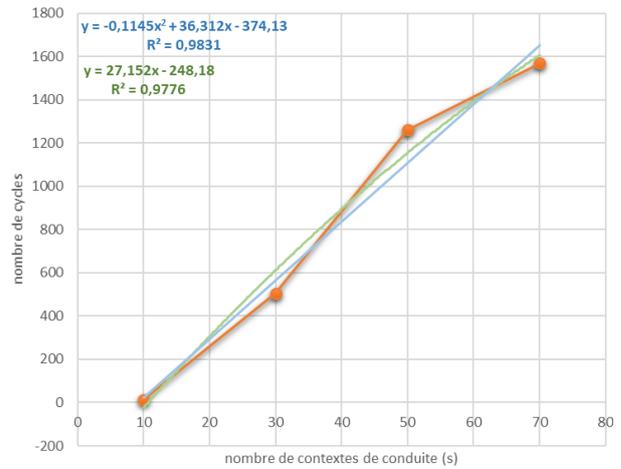


FIGURE 4 – Évolution du nombre de cycles de vie des agents en fonction du nombre de contextes de conduite s .

relation quadratique avec $-0,1145$ comme coefficient de x^2 et $36,312$ de x , permet de considérer la relation comme proche du linéaire, puisque le coefficient de x est plus important que celui de x^2 . Ainsi, nous considérons que la relation entre s et le nombre de cycles de vie des agents comme presque linéaire.

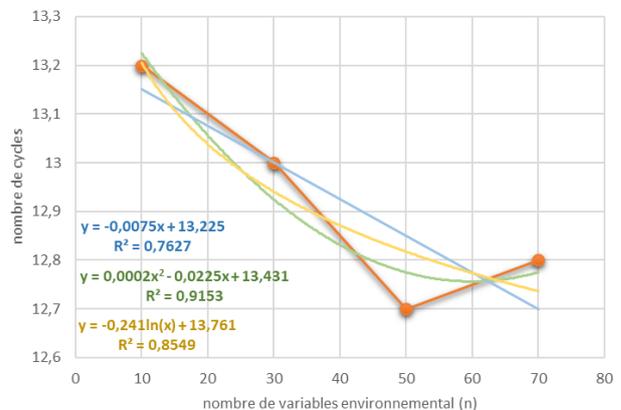


FIGURE 5 – Évolution du nombre de cycles de vie des agents en fonction du nombre de variables environnementales n .

La figure 5 illustre les résultats de l'étude pour la partie B. La relation entre le nombre de cycles de vie des agents et le nombre de variables environnementales n suit une tendance plutôt polynomiale, puisque le coefficient de détermination est de 0,7627 pour la linéaire, 0,8549 pour la logarithmique, et 0,9153 pour la polynomiale. Ceci est potentiellement expliqué par l'approximation linéaire de fonctions non linéaires impliquant une plus forte interaction entre les agents afin de converger. Néanmoins, les agents VA_i sont plus en difficulté et la détection des comportements cycliques est plus importante. Là aussi, le faible nombre de contexte de conduite $s = 10$ peut expliquer le manque d'informations nécessaires aux agents VA_i pour leur permettre de converger.

5 Conclusion et Perspectives

Dans cet article, nous avons introduit un modèle de supervision du système artificiel d'un véhicule autonome et de sa capacité à gérer un contexte de conduite, basé sur l'apprentissage par l'observation de l'activité de conduite. ADORSiDe est capable d'estimer l'état réel attendu de chaque variable environnementale qui constitue un contexte de conduite. Cette valeur d'estimation dépend de l'influence des autres variables environnementales et de l'état de l'ensemble des actions de conduite. L'adaptation du modèle se fait par des ajustements des valeurs de poids des agents variables. Le système converge lorsque la précision souhaitée est atteinte. Pour évaluer la capacité du conducteur à gérer un contexte de conduite, ADORSiDe utilise les valeurs précédemment apprises et les compare aux valeurs de l'état souhaité par le conducteur autonome, et si cette différence est supérieure à un seuil fixé, ADORSiDe conclut que le contexte de conduite est anormal et que le conducteur autonome n'est pas en mesure de le gérer.

Les expériences réalisées avec un générateur synthétique ont montré à la fois la capacité d'ADORSiDe à atteindre un certain niveau de précision dans l'ajustement des poids et l'existence d'une relation presque linéaire entre le nombre de paramètres s et n , et le nombre de cycles pour trouver une solution.

Cependant, les performances du système doivent être évaluées en prenant en compte un contexte beaucoup plus riche (plus grand nombre de situations et de variables environnementales). De plus, la capacité d'ADORSiDe à fournir des résultats en un temps très restreint pour respecter le cycle de décision du véhicule autonome doit être étudiée. La pertinence de l'approximation linéaire pour approcher les modèles des variables environnementales doit également être approfondie. Nous devons réaliser davantage plus d'expérimentations sur le modèle qui effectue l'évaluation de la capacité du conducteur tout en respectant un temps contraint. La prochaine étape consiste à intégrer les résultats fournis par les études menées en SHS afin d'expliquer les décisions du système pour faciliter la reprise du contrôle de la conduite et le rendre acceptable par l'humain.

Remerciements

Nous remercions l'opération scientifique neOCampus de l'Université Toulouse 3 et la région Occitanie qui financent ce travail de thèse.

Références

- [1] Jose Aguilar, Kristell Aguilar, Danilo Chávez, Jorge Cordero, and Eduard Puerto. Different intelligent approaches for modeling the style of car driving. In *ICINCO (2)*, pages 284–291, 2017.
- [2] Javier Albusac, David Vallejo, Luis Jimenez-Linares, Jose Jesus Castro-Schez, and Luis Rodriguez-Benitez. Intelligent surveillance based on normality analysis to detect abnormal behaviors. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(07) :1223–1244, 2009.
- [3] Shumeet Baluja. Evolution of an artificial neural network based autonomous land vehicle controller. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(3) :450–463, 1996.
- [4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning : A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8) :1798–1828, 2013.
- [5] Jérémy Boes. *Apprentissage du contrôle de systèmes complexes par l'auto-organisation coopérative d'un système multi-agent*. PhD thesis, 2014.
- [6] Nicolas Brax. *Self-adaptive multi-agent systems for aided decision-making : an application to maritime surveillance*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2013.
- [7] Fabrice Crasnier, Jean-Pierre Georgé, and Marie-Pierre Gleizes. Real-time learning of power consumption in dynamic and noisy ambient environments. In *International Conference on Computational Collective Intelligence*, pages 443–454. Springer, 2019.
- [8] Gary L Drescher. *Made-up minds : a constructivist approach to artificial intelligence*. MIT press, 1991.
- [9] Florian Eyben, Martin Wöllmer, Tony Poitschke, Björn Schuller, Christoph Blaschke, Berthold Färber, and Nhu Nguyen-Thien. Emotion on the road—necessity, acceptance, and feasibility of affective computing in the car. *Advances in human-computer interaction*, 2010, 2010.
- [10] Pierre Glize. L'adaptation des systèmes à fonctionnalité émergente par auto-organisation coopérative. *Hdr, Université Paul Sabatier, Toulouse III*, 2001.
- [11] Maxime Guériau. *Systèmes multi-agents, auto-organisation et contrôle par apprentissage constructiviste pour la modélisation et la régulation dans les systèmes coopératifs de trafic*. PhD thesis, 2016.
- [12] Michal Jakob, Ondrej Vanek, Stepán Urban, Petr Benda, and Michal Pechoucek. Agentc : agent-based

- testbed for adversarial modeling and reasoning in the maritime domain. In *AAMAS*, pages 1641–1642, 2010.
- [13] Sébastien Mazac. *Approche décentralisée de l'apprentissage constructiviste et modélisation multi-agent du problème d'amorçage de l'apprentissage sensorimoteur en environnement continu : application à l'intelligence ambiante*. PhD thesis, 2015.
- [14] Ishak Mohamad, Mohd Alauddin Mohd Ali, and Mahamod Ismail. Abnormal driving detection using real time global positioning system data. In *Proceeding of the 2011 IEEE international conference on space science and communication (IconSpace)*, pages 1–6. IEEE, 2011.
- [15] Julien Nigon. Apprentissage artificiel adapté aux systèmes complexes par auto-organisation coopérative de systèmes multi-agents. *Université Paul Sabatier de Toulouse III*, 2017.
- [16] Fannia Pacheco, Ernesto Exposito, Mathieu Gineste, Cedric Baudoin, and Jose Aguilar. Towards the deployment of machine learning solutions in network traffic classification : a systematic survey. *IEEE Communications Surveys & Tutorials*, 21(2) :1988–2014, 2018.
- [17] Filipo Studzinski Perotto. A computational constructivist model as an anticipatory learning mechanism for coupled agent–environment systems. *Constructivist Foundations*, 9(1) :46–56, 2013.
- [18] Paolo Remagnino, Adam Baumberg, Tom Grove, David C Hogg, TN Tan, Anthony D Worrall, Keith D Baker, et al. An integrated traffic and pedestrian model-based vision system. In *BMVC*, 1997.
- [19] Nino Ronetti and Carlo Dambra. Railway station surveillance : the italian case. In *Multimedia Video-Based Surveillance Systems*, pages 13–20. Springer, 2000.
- [20] Arturo Servin and Daniel Kudenko. Multi-agent reinforcement learning for intrusion detection. In *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, pages 211–223. Springer, 2005.
- [21] G Silberg et al. I see, i think, i drive (i learn), 2016.
- [22] Niklas Strand, Josef Nilsson, IC MariAnne Karlsson, and Lena Nilsson. Semi-automated versus highly automated driving in critical situations caused by automation failures. *Transportation research part F : traffic psychology and behaviour*, 27 :218–228, 2014.
- [23] Kok S Tan. A multi-agent system for tracking the intent of surface contacts in ports and waterways. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 2005.
- [24] Fabio Tango, Caterina Calefato, Luca Minin, and Luca Canovi. Moving attention from the road : A new methodology for the driver distraction evaluation using machine learning approaches. In *2009 2nd Conference on Human System Interactions*, pages 596–599. IEEE, 2009.
- [25] Maria Valera and Sergio A Velastin. Intelligent distributed surveillance systems : a review. *IEE Proceedings-Vision, Image and Signal Processing*, 152(2) :192–204, 2005.
- [26] David Vallejo, Javier Albusac, Luis Jimenez, Carlos Gonzalez, and Juan Moreno. A cognitive surveillance system for detecting incorrect traffic behaviors. *Expert Systems with Applications*, 36(7) :10503–10511, 2009.
- [27] Nicolas Verstaevel. *Self-organization of robotic devices through demonstrations*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2016.
- [28] Christian Wöhler and Joachim K Anlauf. Real-time object recognition on image sequences with the adaptable time delay neural network algorithm—applications for autonomous vehicles. *Image and Vision Computing*, 19(9-10) :593–618, 2001.

Étude de différentes configurations d'autoconsommation collective de l'énergie à l'échelle du quartier à l'aide de la simulation multi-agent

J. Albouys
Perrois^{a, b, c, d}
albouys@limsi.fr

N. Sabouret^b
nicolas.sabouret@limsi.fr

Y. Haradji^c
yvon.haradji@gmail.com

M. Schumann^{c, d}
mathieu.schumann@edf.fr

B. Charrier^{c, d}
benoit.charrier@edf.fr

C. Inard^{da, d}
christian.inard@univ-lr.fr

^a Laboratoire des sciences de l'ingénieur pour l'environnement,
Université de La Rochelle, France

^b Université Paris-Saclay, CNRS, LIMSI, 91400, Orsay, France

^c EDF R&D, France

^d 4evLab, EDF R&D, CNRS, LaSIE, La Rochelle Université, Avenue Michel Crépeau, La Rochelle Cedex 1, 17042,
France

Résumé

Cet article présente un modèle de simulation multi-agent à deux niveaux d'autoconsommation collective de l'énergie. Le premier niveau de ce modèle est une simulation multi-agent de l'activité humaine couplée à une simulation thermique du bâtiment qui permet d'obtenir la consommation électrique d'un foyer. Le second niveau est une modélisation d'un groupement de foyers pratiquant l'autoconsommation collective d'énergie. Nous présentons une formalisation de cette notion de groupement ainsi que différentes organisations pour échanger de l'énergie. Nous étudions ensuite ces différentes organisations et montrons leur fort impact sur la répartition de l'énergie lorsque la production est faible face à la consommation.

Mots-clés : *Simulation multi-agent, Simulation, Consommation énergétique, Autoconsommation collective*

Abstract

We present a multi-level multi-agent simulation of collective energy self-consumption. The first level of this model is a

multi-agent simulation of human activity coupled with an energy simulation of the building, calculating the household's energy consumption. The second level is a model of a group of households practising collective self-consumption of energy. We present a formalisation of this notion of grouping as well as different organizations to exchange energy. Last, we study these different organisations and show their strong impact on the distribution of energy when production is low compared to consumption.

Keywords: *Multi-agent simulation, Energetic consumption, Collective self-consumption*

1 Introduction

L'autoconsommation collective (ACC) d'énergie associe une production d'énergie électrique locale, par exemple avec des panneaux photovoltaïques (PV) ou une éolienne, avec la consommation de toute ou d'une partie de cette énergie par un ensemble de foyers situés à proximité des moyens de production [7, 8]. L'un des principaux intérêts est de maximiser la consommation d'énergie

locale et renouvelable sans devoir faire appel à des sources d'énergie centralisées. Complémentaire avec les expérimentations sur site, la simulation numérique est un outil permettant d'étudier la pertinence technico-économique des nombreuses configurations d'ACC possibles et l'évolution des usages de l'énergie dans ces situations. [8].

Dans nos précédents travaux [1, 2] nous avons déjà mis en avant la nécessité de prendre en compte la diversité de l'activité des habitants des foyers dans les problématiques de l'ACC, en mettant en évidence l'impact positif sur le taux d'autoconsommation collective, c'est-à-dire la part de la production électrique locale qui est consommée immédiatement sur place par les foyers. Cependant, les premiers résultats obtenus étaient basés sur une seule configuration d'ACC simplifiée : dans notre étude, les foyers s'échangeaient l'énergie produite localement de manière séquentielle i.e. un foyer B recevait l'énergie que le foyer A ne consommait pas. Il existe toutefois de nombreuses configurations possibles d'autoconsommation collective (en termes d'acteurs, d'organisation des échanges d'énergie, de cadre contractuel, ou d'équipements de production ou de stockage d'énergie) que notre étude précédente n'avait pas encore mis en avant.

Dans cet article, nous proposons un modèle de simulation multi-agent à deux niveaux, afin de rendre compte de la diversité des configurations d'ACC et d'étudier leur impact sur le taux d'autoconsommation. Comme nous l'avons montré dans nos travaux précédents [3], la prise en compte de l'activité humaine est importante pour étudier avec justesse la dynamique des consommations dans le secteur résidentiel. C'est pourquoi nous utiliserons, dans un premier niveau, la plateforme SMACH pour simuler les foyers et, par co-simulation, le modèle énergétique du bâtiment. Dans cet article nous présentons le second niveau de notre modèle qui est aussi un système multi-agent (SMA) et dont le rôle est de simuler le fonctionnement de la configuration d'ACC étudiée. L'objectif de ce

niveau est de permettre de simuler un grand nombre configuration d'ACC différentes. Ceci permettra de fournir un outil aux experts de l'énergie afin d'étudier et d'anticiper les futures situations et problématiques liées à l'ACC, comme le dimensionnement des équipements liés à l'ACC.

Nous présentons la conception, la formalisation et l'implémentation actuelle de notre modèle. Dans la section 2, nous discutons des travaux antérieurs portant sur la simulation de regroupement d'agents et sur les problèmes d'allocation par des SMA. La section 3 est dédiée à la présentation de notre modèle d'ACC. La section 4 est dédiée aux modèles d'échange de l'énergie. Enfin, la section 5 présente un cas d'étude précis et ses résultats.

2 État de l'art

2.1 Agent, groupe et ACC

Une des caractéristiques de l'ACC est la possibilité pour des foyers de se regrouper afin d'échanger entre eux de l'énergie produite localement. En SMA, la formation de groupe d'agents et leur optimisation vis-à-vis d'un ou plusieurs objectifs sont des sujets largement étudiés, avec des notions comme les coalitions, les équipes ou encore les congrégations [6].

Cependant, dans nos travaux, les regroupements de foyers pratiquant l'ACC sont définis et fixés de manière contractuelle. Nous ne cherchons donc pas à optimiser ces regroupements pratiquant l'ACC, mais à fournir un outil visant à étudier l'impact de différents types de groupement (en termes de foyers, d'habitants, de moyens de production et d'échange d'énergie). Nous ne sommes ni dans un problème de formation de coalition ni d'optimisation de la consommation.

2.2 Échanges de ressources

L'énergie produite localement sera amenée à être échangée entre les différents acteurs

pratiquant l'ACC. L'échange (ou allocation) de ressources est un problème fréquemment étudié dans les systèmes multi-agents. Un problème d'allocation peut être défini comme un problème dans lequel un ensemble d'agents (coopératifs ou non) doit trouver une répartition de ses ressources [4]. Cette répartition doit correspondre au mieux aux préférences des agents qui sont souvent en compétition.

Pour résoudre ces problèmes d'allocation, les auteurs de [4] proposent la mise en place d'un système multi-agent décentralisé basé sur les processus décisionnels de Markov (PDM). Dans ce modèle, les agents vont être capables de faire ou d'accepter des propositions d'échanges de ressources avec d'autres agents. L'objectif des PDM est de rechercher une optimisation et une planification des différents échanges de ressources. Or aujourd'hui, l'ACC impose une organisation contractuelle précise pour régir les échanges d'énergie entre les foyers. Dans nos travaux notre objectif est de voir comment différentes méthodes pour échanger de l'énergie impactent la consommation au sein d'un groupement sans chercher à les optimiser.

D'autres méthodes se sont intéressées à la mise en œuvre de systèmes centralisés afin de réaliser l'allocation de ressources. C'est le cas par exemple du « Contract-Net Protocol » [11]. Ce protocole consiste en l'interaction d'un manager qui va propager une offre i.e. un ensemble de ressources à d'autres agents qui réaliseront une proposition (d'achat par exemple) au manager. Le manager sélectionne ensuite la meilleure proposition puis va attribuer les ressources. Le fonctionnement avec des agents en concurrence pour l'obtention de ressources est similaire au fonctionnement des enchères dites scellées. Cependant, une seule ressource ne sera attribuée qu'à un seul acheteur. Cela ne correspond pas forcément à la façon dont l'énergie électrique peut être répartie dans le cas d'enchères. D'autres algorithmes existent visant à reproduire des systèmes d'enchères pour allouer des ressources entre vendeurs

[12].

Constatant le besoin d'un modèle capable de répondre aux contraintes de l'ACC réelle, cet article présente le modèle que nous utilisons pour simuler l'ACC entre un ensemble de foyers, ainsi que les différentes organisations d'autoconsommation d'énergie associées à l'allocation de l'énergie produite localement entre les différents foyers. Ce modèle sera basé sur la simulation multi-agent à deux niveaux pour simuler l'ACC. Au centre de modèle, on retrouvera le **Groupement**. Un des apports de ce modèle est de permettre le test et la comparaison de différentes organisations d'ACC sur la consommation électrique des foyers et sur les flux financiers.

3 Le modèle de groupement

Dans cette section et dans la suivante, nous présentons notre modèle de configurations d'ACC. Nous présentons tout d'abord la notion de groupement et le modèle SMA à deux niveaux qui y est rattaché. La section 4 présentera l'organisation des échanges.

3.1 Le groupement : un SMA à deux niveaux

Les acteurs sont des agents présents dans le groupement, ces derniers représenteront les foyers. Les foyers sont des systèmes multi-agents qui sont simulés par la plateforme SMACH dont les développements successifs ont déjà été abordés [2]. La plateforme permet d'obtenir la consommation électrique des foyers en simulant l'activité humaine grâce à un système multi-agent. Dans ce SMA, chaque agent représente un habitant du foyer qui va sélectionner des tâches quotidiennes à réaliser. Les activités sélectionnées sont associées à des appareils qui consomment de l'énergie électrique. Nous pouvons ainsi obtenir la consommation électrique de chaque foyer. Les foyers peuvent ainsi avoir un besoin en énergie qui peut être comblé en partie ou totalement par un moyen de production individuel (situé au niveau du foyer). Cela est évalué par une

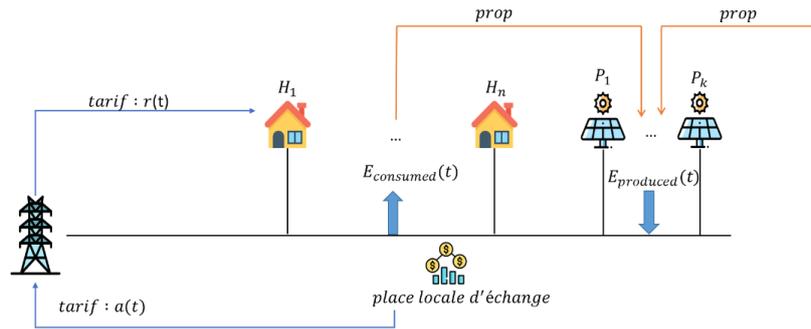


Figure 1 : Représentation schématique d'un groupement

nouvelle entité intégrée dans chacun des foyers pratiquant l'ACC : **le module d'autoconsommation**. Le rôle de ce module est de calculer le besoin ou le surplus en énergie du foyer et de le communiquer au groupement à un pas de temps défini. Les foyers peuvent également être copropriétaires d'un moyen de production d'énergie mutualisé, c'est-à-dire situé au niveau du groupement.

Un groupement est également composé d'équipements : moyens de production et de stockage. Les moyens de production à l'échelle du groupement sont dits mutualisés. Ces derniers sont à la disposition du groupement et l'ensemble de leur production d'énergie sera amenée à être répartie entre les foyers du groupement. Ces moyens de production sont modélisés grâce au langage Modelica et à la bibliothèque BuildSysPro comme présenté dans nos précédents travaux [9]. Pour ce faire, une co-simulation entre le groupement et les moyens de production mutualisés s'appuyant sur le standard d'interopérabilité logicielle FMI a été développée [1]. Ces moyens de production mutualisés peuvent appartenir à un ou plusieurs acteurs du groupement, c'est-à-dire que chacun a un taux de propriété qui sert à la répartition des bénéfices.

Enfin, la place locale d'échange (PLE) est un agent présent dans notre groupement, dont le rôle principal est de définir comment l'énergie produite au sein du groupement est répartie entre les différents foyers. Pour cela, elle calcule la répartition de cette énergie en fonction de règles contractuelles définies entre

les foyers. Cet ensemble de règles est appelé **organisation de l'autoconsommation** et pourra être différent d'un groupement à l'autre, mais toujours identique au sein d'un groupement. Une fois l'énergie produite localement répartie, la place locale d'échange calcule également la quantité d'énergie que chaque foyer a consommée auprès du réseau électrique.

3.2 Définition formelle du groupement

Soit H l'ensemble des foyers qui composent un groupement. Pour chacun de ces foyers, nous notons $e_{consumed}(h, t)$ la quantité d'énergie en kWh appelée par un foyer h sur le précédent pas de temps de notre simulation. Dans notre groupement, $Prod$ permet de représenter l'ensemble des moyens de production mutualisés du groupement. Nous notons $e_{produced}(t): Prod \rightarrow R^+$ la quantité d'énergie en kWh produite par un moyen de production mutualisé du groupement sur le pas de temps précédent. Nous notons également $prop: H \times Prod \rightarrow [0,1]$, le taux de propriété d'un acteur du groupement envers un moyen de production. Ce taux de propriété permettra de calculer la répartition des bénéfices. Par définition, les valeurs de $prop$ doivent toujours satisfaire la contrainte suivante : $\forall j \in Prod, \sum_{i \in H} prop(i, j) = 1$.

Nous notons également $E_{produced}(t) = \sum_{j \in Prod} e_{produced}(j, t)$ la quantité totale d'énergie produite par l'ensemble des moyens de production du groupement sur le pas de temps précédent.

Dans notre groupement, nous notons $r(t): H \rightarrow \mathbb{R}^+$ le prix de l'énergie provenant de l'extérieur du groupement pour un foyer au temps t . Nous notons également $a(t) \in \mathbb{R}^+$ le prix auquel l'énergie produite pourra être vendue à l'extérieur du groupement. Actuellement, nous considérons que seul le réseau peut vendre ou acheter de l'énergie depuis l'extérieur du groupement.

Pour un groupement, nous notons $pol = \{TauxFixe, TauxProportionnel, Concurrentielle\}$ l'organisation des échanges d'énergie mise en place dans le groupement, tel que défini dans la section 4.

Nous avons pour chaque foyer la fonction $e_{local}(t): H \rightarrow \mathbb{R}^+$ qui représente la quantité d'énergie obtenue à l'aide des moyens de production mutualisés du groupement. Nous avons également la contrainte $e_{consumed}(t) \geq e_{local}(t)$, c'est-à-dire qu'un foyer ne peut pas recevoir plus d'énergie qu'il n'en consomme.

Le groupement (Figure 1) est donc une simulation à deux niveaux de l'ACC. Le premier niveau permet de simuler l'activité humaine et la consommation d'énergie à l'échelle des foyers. Le second permet la simulation des différentes configurations d'ACC.

4 Modélisation des échanges

4.1 Principe général

Dans cette section nous présentons la notion d'organisation des échanges d'énergie et comment ils sont implémentés. Formellement, nous définissons l'organisation des échanges d'énergie comme l'ensemble des règles contractuelles qui vont permettre d'échanger l'énergie produite localement au sein d'un groupement. En entrée, ces organisations prennent les quantités d'énergie produite et appelée dans le groupement. En sortie, une organisation doit fournir la répartition de l'énergie produite localement c'est-à-dire la quantité d'énergie que chaque foyer a reçu et, si cela s'applique, le prix payé par chacun des

foyers pour cette énergie. Dans le groupement, c'est la place locale d'échange qui est responsable de la mise en place de ces règles.

Nos premières enquêtes et études [2] nous ont permis d'identifier trois types d'organisation des échanges entre les foyers que nous allons formaliser :

- Taux fixe : chaque foyer reçoit une quantité d'énergie selon un taux fixe défini à l'avance
- Taux proportionnel : chaque foyer reçoit une quantité d'énergie en fonction de sa consommation
- Concurrentielle : l'énergie est achetée ou vendue sur un marché où vendeurs et acheteurs interagissent entre eux.

4.2 Taux fixe

Dans le cadre d'une organisation à taux fixe, les différents foyers qui composent un groupement reçoivent une part de l'énergie d'un ou plusieurs moyens de production.

Nous définissons $taux_{energie}$ une fonction telle que $taux_{energie}: H \times Prod \rightarrow [0,1]$, correspondant au taux d'énergie qu'un foyer peut récupérer auprès d'un moyen de production, par rapport à sa production totale. Ce taux peut être contractuellement défini en fonction de différents paramètres : par exemple proportionnellement à la surface des foyers (au tantième) ou en fonction du nombre de foyers dans le groupement. Nous notons $\forall j \in Prod, \sum_{h \in H} taux_{energie}(h, j) = 1$ la contrainte qui exprime le fait que pour un moyen de production donné, la somme de ses taux d'énergie doit être égale à 1.

Cependant, avec ce type d'organisation il peut arriver qu'un foyer reçoive plus d'énergie qu'il n'en a besoin. Ainsi, nous notons $surplus_{fixe}(t)$ le surplus d'énergie qui apparaît lorsque des foyers ne peuvent pas consommer toute l'énergie que le groupement leur fournit. Il faut alors développer une stratégie pour gérer ce surplus d'énergie. Nous avons identifié deux stratégies possibles : la

revente complète du surplus ou une nouvelle répartition avec un algorithme itératif comme présenté à la Figure 2. Cet algorithme calcule le surplus une fois les échanges au sein du groupement réalisés puis va répartir ce surplus entre les foyers ayant encore des besoins.

```

1 Soit n le nombre de foyer tel que besoin(h, c) > 0;
2 Soit surplus_fixe(c) > 0;
3 tant que surplus_fixe(c) ≥ 0 et n ≥ 0 faire
4   residu = surplus_fixe(c);
5   pour chaque foyer h faire
6     si besoin(h, c) >  $\frac{\text{surplus\_fixe}(c)}{n}$  alors
7       besoin(h, c) =  $\frac{\text{surplus\_fixe}(c)}{n}$ ;
8       residu =  $\text{surplus\_fixe}(c) - \frac{\text{surplus\_fixe}(c)}{n}$ ;
9       e_local(h, c) = e_local(h, c) +  $\frac{\text{surplus\_fixe}(c)}{n}$ ;
10    fin
11   sinon si besoin(h, c) ≥ 0 alors
12     e_local(h, c) = e_local(h, c) + besoin(h, c);
13     besoin(h, c) = 0;
14     residu = residu - besoin(h, c);
15     n = n - 1;
16   fin
17 fin
18 surplus_fixe(c) = residu
19 fin

```

Figure 2: Algorithme de répartition itérative du surplus d'énergie pour une organisation à échanges à taux fixe

Pour chaque foyer, nous calculons $avoir(h, t) = \sum_{j \in Prod} [taux_{\text{energie}}(h, j) * e_{produced}(j, t)]$ la quantité d'énergie produite localement et qu'un foyer peut recevoir et $besoin(h, t)$ son besoin en énergie. Deux cas sont alors identifiables :

- $avoir(h, t) < besoin(h, t)$
 - $e_{local}(h, t) = avoir(h, t)$
- $avoir(h, t) > besoin(h, t)$
 - $e_{local} = besoin(h, t)$
 - $surplus(h, t) = avoir(h, t) - besoin(h, t)$

4.3 Taux proportionnel

Au sein d'un groupement pratiquant une organisation à taux proportionnel, les différents foyers reçoivent une partie de l'énergie produite localement proportionnellement à leur besoin en énergie.

Pour la mise en place de cette organisation nous utilisons $E_{produced}(t)$ la quantité totale d'énergie produite au sein d'un groupement.

Nous utilisons également $E_{consumed}(t)$ la quantité totale d'énergie consommée par les foyers du groupement.

Nous sommes alors confrontés à deux cas pour répartir l'énergie :

$E_{consumed}(t) > E_{produced}(t)$, dans ce cas pour chaque foyer nous avons :

$$e_{local}(h, t) = \frac{besoin(h, t)}{E_{consumed}(t)} * E_{produced}(t)$$

$E_{consumed}(t) \leq E_{produced}(t)$, dans ce cas pour chaque foyer nous avons :

$$e_{locale}(h, t) = e_{consumed}(h, t)$$

Ce type d'organisation permet de répartir toute l'énergie entre les foyers sans leur attribuer plus que ce qu'ils ne demandent.

4.4 Concurrentielle

Dans une organisation concurrentielle, les différents foyers s'échangent de l'énergie à travers un marché et des processus d'achat et de vente. Nous considérons deux fonctionnements possibles dans nos travaux. Un premier dans lequel les prix sont fixés par les acheteurs et un second qui considère que les vendeurs fixent les prix. Dans les deux cas, nous considérons deux tableaux : $offres_{achat}$ et $offre_{vente}$ contenant chacune des offres de vente et d'achat d'énergie.

Positionnons-nous par exemple dans le cas où les acheteurs fixent les prix. Le tableau $offres_{achat}$ sera composé d'ensembles : $achat = \langle acheteur, quantité_{achat}, prix \rangle$

Le tableau $offre_{vente}$ quant à lui sera composé d'ensembles : $vente = \langle vendeur, quantité_{vente} \rangle$.

L'algorithme de calcul de cette organisation va faire correspondre les offres d'achat proposant le prix le plus élevé avec les différents acheteurs. Ce processus est répété jusqu'à ce qu'il n'y ait plus d'énergie à vendre ou que les offres des acheteurs soient satisfaites.

Le processus est identique dans le cas où les vendeurs fixent les prix, sauf que ce sont les

	Foyer A	Foyer B	Foyer C	Foyer D	Foyer E	Foyer F	Foyer G	Foyer H	Foyer I	Foyer J
Surface	115m ²	68m ²	72m ²	50m ²	257m ²	95m ²	55m ²	206m ²	59m ²	77m ²
Réglementation thermique	2020_H1H2	2012	1989	Avant 1974	2000_H1H2	1974	1974	BBC	1974	2000H1H2
Chauffage électrique	Non	Oui	Non	Oui	Non	Oui	Oui	Non	Non	Oui
ECS électrique	Non	Oui	Oui	Oui	Non	Oui	Oui	Non	Non	Oui
Heures creuses	Base	22h30-6h30	23h00-7h00	23h30-7h30	Base	22h30-6h30	23h30-7h30	22h30-6h30	Base	2h00-7h00 / 14h00-17h00
Panneaux PV	Oui	Oui	Oui	Non	Non	Non	Non	Non	Non	Non
Nh Habitants	1	1	2	1	2	4	2	2	1	2

Tableau 1: Paramètres des foyers

offres de vente avec les prix de vente les plus bas qui seront considérées en priorité.

Cette organisation pose également la question de la formation des prix pour les acheteurs et les vendeurs. Nous n'abordons pas cette problématique dans cet article, mais un mécanisme de formation des prix sera intégré dans le futur. Par mesure de simplification, les prix sont pour le moment sélectionnés aléatoirement dans un ensemble donné.

L'organisation concurrentielle proposée ici est un modèle simple, d'autres systèmes concurrentiels seront intégrés dans le futur comme des enchères à doubles carnets d'ordres [12] pour permettre l'étude de différentes organisations concurrentielles d'échanges.

5 Cas d'étude et résultats

5.1 Contexte de la simulation

Nous considérons un groupement composé de 10 foyers dans la ville de Nice. Dans le Tableau 1 nous donnons les principaux paramètres des foyers étudiés. En plus des caractéristiques du bâtiment, nous fournissons deux informations sur les familles à savoir le nombre d'habitants et s'il y a au moins un retraité ou une personne inactive, c'est-à-dire présente dans le foyer pendant les jours ouvrés. Les activités des individus de ces foyers ont été générées en s'appuyant sur l'Enquête Emploi du Temps de l'INSEE [10].

Dans le groupement, les foyers A, B et C produisent de l'énergie locale grâce à des panneaux PV. Ces trois foyers possèdent chacun une installation PV identique avec les caractéristiques suivantes :

- Surface : 20 m²
- Inclinaison : 30°
- Azimut : Sud

La puissance électrique de chacune de ces installations est de 2,5 kWc (kilowatt-crête). Ces foyers acceptent d'échanger le surplus d'énergie produite avec leurs voisins au sein du groupement. En utilisant ce cas d'étude et notre modèle de groupement, nous évaluons l'impact des différentes organisations des échanges d'énergie sur la consommation d'énergie des foyers et du groupement, ainsi que sur le taux d'autoconsommation.

5.2 Impact de l'organisation des échanges sur l'autoconsommation d'énergie

Dans un premier temps nous allons analyser l'impact des différentes organisations des échanges à l'échelle du groupement. La Figure 3 montre la consommation d'électricité totale au sein du groupement ainsi que la production locale pour le mois de juillet 2019. Pour le mois simulé, le groupement a consommé un total de **1935 kWh** et produit **1638 kWh** grâce aux panneaux PV.

Deux variables sont observées afin d'évaluer l'impact des stratégies d'échanges d'énergie. Tout d'abord, **le taux d'autoconsommation** qui permet de connaître la proportion de la production locale directement consommée par les foyers. Ensuite, **le taux d'autoproduction** d'électricité calculé de la façon suivante :

$$taux_{autoproduction} = \frac{consommation_{locale}}{consommation_{totale}}$$

	Foyer A	Foyer B	Foyer C	Foyer D	Foyer E	Foyer F	Foyer G	Foyer H	Foyer I	Foyer J
Taux fixe : tantième	42%	17%	20%	21%	54%	18%	30%	40%	61%	34%
Proportionnel	40%	16%	20%	23%	51%	19%	30%	37%	60%	37%
Concurrentiel : au plus offrant	37%	19%	21%	21%	49%	17%	28%	35%	58%	35%

Tableau 2 : Taux d'autoproduction d'électricité des foyers avec trois panneaux PV pour le mois de juillet 2019 à Nice

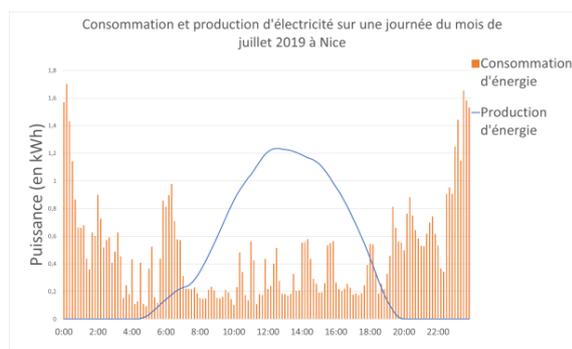


Figure 3 : Consommation et production d'électricité pour une journée de juillet 2019 à Nice

Les taux d'autoproduction du groupement (Tableau 3) sont relativement proches, quelle que soit l'organisation des échanges d'énergie au sein du groupement. Ce résultat provient du fait que les échanges répartissent le maximum d'énergie possible entre les foyers, quelle que soit l'organisation mise en place.

Le Tableau 2 donne le taux d'autoproduction des foyers en fonction de la façon dont les échanges sont réalisés par la PLE. Ces taux sont relativement proches, quelle que soit la façon dont nous organisons les échanges. Cela s'explique principalement par le fait que dans notre cas d'étude, la quantité d'énergie produite par les trois installations PV est largement suffisante pour couvrir les besoins en énergie du groupement en journée.

Cela se confirme en analysant la consommation en énergie du groupement pendant la journée. Durant cette période, on constate que les foyers ont consommé au total **937 kWh** d'énergie, ce qui est bien inférieur à la production d'électricité du groupement qui est de **1638 kWh**.

	Taux d'autoproduction	Taux d'autoconsommation
Taux fixe : tantième	31%	36%
Proportionnelle	31%	36%
Concurrentielle : au plus offrant	30%	35%

Tableau 3 : Impact des différentes organisations des échanges de l'énergie au sein du groupement

Cette simulation met en avant le fait que lorsque la production d'énergie dans un groupement est supérieure à la consommation de celui-ci, l'organisation choisie pour les échanges n'a qu'un faible impact sur la quantité d'énergie reçue par chaque foyer. La quantité d'énergie locale non consommée est alors relativement importante, dans ce cas **1033 kWh** sur la durée totale de la simulation. Cette énergie sera ensuite revendue à l'extérieur du groupement au fournisseur d'électricité.

Afin d'étudier la situation limitante où la quantité d'énergie produite dans le groupement est fortement inférieure à la consommation de ce dernier, nous avons simulé un second cas constitué de 10 foyers situés à Lille pour le mois de décembre 2019. Ici, seul le foyer A possède un moyen de production locale. La Figure 4 montre une journée de simulation dans cette nouvelle configuration, où la production d'énergie locale est très faible par rapport à la consommation totale du groupement.

Les taux d'autoproduction d'électricité des foyers donnés dans le Tableau 4 sont très faibles et on constate des différences entre les différentes stratégies d'échange de l'énergie.

	Foyer A	Foyer B	Foyer C	Foyer D	Foyer E	Foyer F	Foyer G	Foyer H	Foyer I	Foyer J
Taux fixe : tantième	5,64%	0,70%	1,50%	5,25%	7,13%	8,66%	5,61%	15,41%	5,24%	7,30%
Proportionnel	0,80%	0,45%	0,31%	0,82%	0,83%	0,76%	0,81%	1,07%	1,43%	0,77%
Concurrentiel : au plus offrant	15,00%	0,35%	0,25%	0,65%	0,66%	0,60%	0,66%	0,85%	1,21%	0,62%

Tableau 4 : Taux d'autoproduction d'électricité des foyers avec un seul panneau PV pour le mois de décembre 2019 à Lille

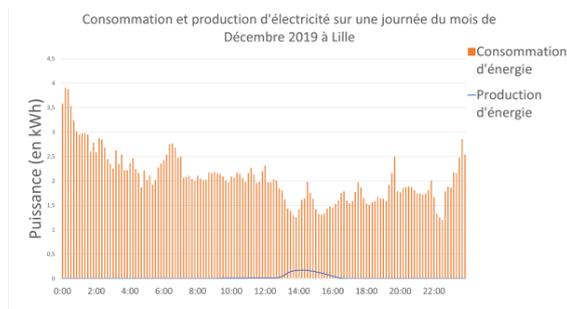


Figure 4 : Consommation et production d'électricité pour une journée de décembre 2019 à Lille

Par exemple, pour le foyer H une organisation à taux fixe basé sur les tantièmes i.e. la surface totale des foyers, permet de multiplier par 15 son taux d'autoproduction d'électricité par rapport à une organisation proportionnelle. La raison principale est que pour la première organisation (taux fixe), les foyers reçoivent l'énergie en fonction de leur surface alors que dans la seconde ils la reçoivent en fonction de leur consommation. Malgré sa surface importante, la consommation du foyer H reste faible car il ne possède pas de chauffage électrique. Ceci contribue fortement à la différence entre les deux politiques d'échange. Une autre différence importante entre les taux d'autoproduction se retrouve pour le foyer A. En effet, le taux d'autoproduction d'électricité pour une organisation concurrentielle augmente grandement par rapport aux deux autres organisations. Cela s'explique par le fait que le foyer A consomme l'énergie qu'il produit en priorité avant de mettre ce qu'il ne consomme pas en vente. D'autres critères pourraient être envisagés pour comparer les résultats entre les foyers en s'appuyant sur des métriques sociales du domaine[5].

Cette simulation a permis de montrer que, dans une situation où l'énergie produite localement

est faible par rapport à la consommation d'énergie du groupement, la façon dont les échanges sont organisés a un impact primordial sur l'énergie que chaque foyer recevra. Au-delà de la consommation d'énergie seule, ces différences impacteront également les flux financiers au sein du groupement, ce qui en fera un critère de choix de l'organisation optimale.

6 Conclusion

Dans cette étude, nous avons présenté une simulation à deux niveaux pour l'étude de configurations d'ACC collective de l'énergie : un ensemble de SMA basés sur la plateforme SMACH simulant l'activité humaine et la consommation électrique de chacun des foyers, et le groupement, un second système multi-agent dont le rôle est de permettre l'échange d'énergie. Nous présentons également différentes possibilités pour échanger de l'énergie. Les résultats semblent indiquer que ces différentes organisations d'échange de l'énergie ont un impact sur la consommation des foyers. L'impact est faible lorsque la production suffit à subvenir aux besoins. Au contraire, la modalité d'échange de l'énergie a un impact plus fort lorsque la production locale d'énergie est faible par rapport à la consommation totale d'énergie.

Les résultats de notre étude ont également mis en avant qu'une grande quantité d'énergie produite n'a pas pu être utilisée lorsque la production est importante. Pour augmenter le taux d'autoconsommation, une solution est d'introduire des systèmes de stockage collectifs ou individuels dans le groupement, afin de rendre disponible l'énergie de manière différée. L'ajout de tels systèmes

s'accompagnerait de la mise en œuvre d'une politique spécifique de stockage, à savoir un ensemble de règles au sein du groupement et des foyers pour définir le contrôle des séquences de stockage et de déstockage. L'implémentation de systèmes et politiques de stockage est la suite directe de notre travail.

L'autoconsommation collective peut impacter la politique énergétique des foyers individuels, c'est-à-dire leur relation à la consommation d'énergie et les processus de décision associés à leur utilisation des appareils électriques. La modélisation de politiques d'échange et de consommation spécifiques à cette situation est donc également à envisager. Il pourra s'agir de politiques impactant le comportement des habitants, notamment en leur faisant déplacer certaines activités consommatrices d'énergie, ou leur rôle dans les échanges en choisissant par exemple de stocker leur surplus pour une utilisation individuelle future.

Références

- [1] Albouys-Perrois, J. et al. 2019. A Co Simulation Of Photovoltaic Power Generation And Human Activity For Smart Building Energy Management And Energy Sharing. *Proceedings of the International Building Performance Simulation Association*. 16, (2019).
- [2] Albouys-Perrois, J. et al. 2019. Simulation multi-agent de l'autoconsommation collective en relation avec l'activité des foyers. *JFSMA 2019. Systèmes distribués, embarqués et diffus*. (2019).
- [3] Amouroux, E. et al. 2013. SMACH: Simuler l'activité humaine pour limiter les pics de consommation électrique. *Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. (2013).
- [4] Beynier, A. and Estivie, S. 2013. Optimizing distributed resource exchanges in multiagent systems under uncertainty. *Distributed and Multi-Agent Planning*. (2013), 8.
- [5] Brams, S.J. and King, D.L. 2005. Efficient Fair Division: Help the Worst off or Avoid Envy? *Rationality and Society*. 17, 4 (Nov. 2005), 387–421. DOI:<https://doi.org/10.1177/1043463105058317>.
- [6] Horling, B. and Lesser, V. 2005. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*. (2005), 281–316.
- [7] Luthander, R. et al. 2015. Photovoltaic self-consumption in buildings: A review. *Applied Energy*. 142, (2015), 80–94. DOI:<https://doi.org/10.1016/j.apenergy.2014.12.028>.
- [8] Matallanas, E. et al. 2011. Analysis of the Self-Consumption Possibilities in Small Grid-Connected Photovoltaic Systems in Spain. *26th European Photovoltaic Solar Energy Conference and Exhibition*. (2011), 6.
- [9] Plessis, G. et al. 2014. BuildSysPro: a Modelica library for modelling buildings and energy systems. (2014). DOI:<https://doi.org/10.3384/ECP140961161>.
- [10] Reynaud, Q. et al. 2017. Using Time-Use Surveys in Multi Agent Based Simulations of Human Activity. *Proceedings of the 9th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART*. Icaart (2017), 67–77. DOI:<https://doi.org/10.5220/0006189100670077>.
- [11] Smith, R.G. 1980. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*. 12 (1980), 1104–1113. DOI:<https://doi.org/10.1109/TC.1980.1675516>.
- [12] Vytelingum, P. et al. 2010. Trading agents for the smart electricity grid. *9th International Conference on AAMAS 2010*. (2010), 897–904.

Ce travail a été financé par le Fonds européen de développement régional (FEDER)

BEN : Une architecture pour des agents cognitifs, affectifs et sociaux dans la simulation

M. Bourgeois^a P. Taillandier^b L. Vercoouter^a
mathieu.bourgeois@insa-rouen.fr patrick.taillandier@gmail.com laurent.vercoouter@insa-rouen.fr

^aNormandie Univ, INSA Rouen, UNIHAVRE, UNIROUEN, LITIS
76000 Rouen, France

^bMIAT, INRA, 31000 Toulouse, France

Résumé

La simulation sociale en tant qu'outil scientifique nécessite le développement de comportements crédibles pour les agents simulant le comportement d'acteurs humains. Une façon d'améliorer la crédibilité des simulations obtenues est d'intégrer des dimensions cognitive, affective et sociales dans la prise de décision des agents. Pour autant, développer des agents avec ces dimensions peut s'avérer compliqué si l'on veut rester accessible à des chercheurs qui ne sont pas experts en informatique. L'architecture BEN (Behavior with Emotions and Norms) intègre des dimensions affectives et sociales à une prise de décisions basée sur une architecture BDI pour la réalisation de simulations sociales. Cette architecture modulaire s'appuie sur une formalisation des dimensions cognitives, affectives et sociales. BEN est implémenté dans la plateforme de modélisation et de simulation multi-agent GAMA et est ici illustrée sur un cas d'évacuation d'urgence.

Mots-clés : Simulation sociales, cognition, émotion, personnalité, normes

Abstract

Social simulation as a scientific tool requires the development of credible behaviours for agents simulating the behaviour of human actors. One way to improve the credibility of the simulations obtained is to integrate cognitive, emotional and social dimensions into agents' decision-making. However, developing agents with these dimensions can be complicated for social science researchers who do not have expert knowledge of computer programming. The BEN architecture (Behavior with Emotions and Norms) integrates emotional and social dimensions into decision making based on a BDI architecture for an application in the field of social simulations. This modular architecture is based on a formalization of the cognitive, affective and social dimensions. BEN is implemented in the GAMA multi-agent

modeling and simulation platform and used on an example case of nightclub evacuation.

Keywords: Social Simulations, cognition, emotion, personality, norms

1 Introduction

Les simulations à base d'agents sont utilisées pour étudier les systèmes complexes impliquant des acteurs humains; il est désormais question, dans ce cas, de simulations sociales [18]. Le but principal est alors de reproduire par simulation des situations réelles où un très grand nombre d'acteurs humains prennent des décisions, influençant ainsi l'évolution globale du système. De cette façon, la situation reproduite peut être étudiée dans un cadre contrôlé.

Le défi consiste alors à développer un comportement de prise de décision crédible pour les agents simulant des acteurs humains. Pour se faire, il est nécessaire d'intégrer des processus cognitifs, affectifs et sociaux dans la prise de décision des agents [39], un point de vue soutenu par le principe EROS (Enhancing Realism of Simulation) [21] qui favorise des simulations à la modélisation proche du cas réel, contrairement aux principes KISS (Keep It Simple, Stupid) [6] et KIDS (Keep It Descriptive, Stupid) [17] qui favorisent des modèles simples, facilement compréhensibles.

Dans cette optique, l'architecture BEN (Behavior with Emotions and Norms), proposée dans cet article et s'appuyant sur de précédents travaux [9], intègre la cognition, les émotions, la personnalité, la contagion émotionnelle, les relations sociales et la définition d'un système normatif pour la prise de décisions en simulation sociale. Ces dimensions cognitives, affectives et sociales sont formalisées dans un même cadre pour pouvoir interagir entre-elles, tout en étant suffisamment indépendantes pour ne pas obliger l'utilisateur à toutes les utiliser si ce n'est pas

nécessaire.

BEN a été implémenté dans la plateforme de modélisation et de simulation multi-agent GAMA [42], réputée pour être accessible à un public non expert en programmation informatique [34][26]. L'utilisation de BEN est illustrée par la simulation d'un cas réel d'évacuation de boîte de nuit pour montrer comment le comportement des agents peut être exprimé de manière crédible et explicable dans les termes de la psychologie naïve [29].

La section 2 de l'article propose un état de l'art sur la prise de décision en simulation sociale, la section 3 expose le formalisme utilisé pour définir les notions liées aux dimensions cognitives, affectives et sociales utilisée dans l'architecture BEN, qui est présentée dans la section 4. La section 5 présente une mise en application de BEN sur le cas d'étude de l'évacuation du Kiss Nightclub au Brésil et la section 6 conclut l'article.

2 État de l'art

Le développement d'agents crédibles est un élément clé pour la simulation sociale qui a été étudié précédemment. Plusieurs travaux ont donc été menés pour créer des architectures agent en s'appuyant sur différentes théories issues de travaux en psychologie ou en sociologie.

Dans le contexte de la simulation sociale, l'ajout de cognition à la prise de décision est une première étape permettant aux agents de prendre des décisions plus complexes que la simple réponse réflexive à une perception [1][7]. Ainsi, la mise en place d'un moteur cognitif doit être la base d'une architecture pour la simulation sociale.

2.1 Prise de décision cognitive

Parmi les différentes plates-formes multi-agents [24], certaines comme JACK [20] ou Jadex [33] implémentent le paradigme BDI (Belief Desire Intention) [12], donnant un comportement cognitif aux agents, en s'appuyant sur un formalisme basé sur la logique modale [15]. Pour autant, ces plates-formes ne sont pas adaptées à la simulation de milliers d'agents.

Pour contourner ce problème, Sing et Padgham [37] proposent de connecter une plate-forme de simulation à un framework BDI existant (comme JACK ou Jadex) et, sur cette même idée [32], la plate-forme Matsim [8] a été liée au framework BDI GORITE [35]. Ces travaux exigent un haut

niveau en informatique, ce qui rend difficile leur utilisation par des modélisateurs ayant de faibles compétences en programmation.

D'autre part, les plates-formes de simulation comme Netlogo [46] ou GAMA [42] sont des logiciels dédiés avec leur propre langage de programmation, leur propre interface et leur propre interpréteur et compilateur. Par défaut, ces plates-formes ne proposent pas d'architecture particulière pour le comportement de l'agent : les modélisateurs doivent définir ces comportements avec des règles "if-then-else". Cependant, il existe des plugins, pour NetLogo [36] et pour GAMA [41], pour utiliser des architectures agents basées sur le paradigme BDI afin de créer des simulations sociales avec un comportement plus complexe et plus crédible. Ils fournissent tous deux aux agents des concepts de haut niveau tels que les croyances et les intentions et le plugin de GAMA va au-delà, offrant un moteur de raisonnement, conduisant les agents à prendre des décisions basées sur la perception de son environnement.

2.2 Prise de décision affective et sociale

Une autre approche consiste à axer le raisonnement des agents autour de leurs émotions. Par exemple, EMA [19] est basé sur la théorie de l'évaluation cognitive des émotions [4] développée par Smith et Lazarus [38] tandis que DETT [45] est basé sur la théorie OCC [31] des émotions. Ces deux systèmes créent des émotions en évaluant les perceptions de l'environnement et en déduisant ensuite un comportement de l'état émotionnel de l'agent.

eBDI [22] s'appuie également sur la théorie OCC mais utilise une architecture BDI pour prendre des décisions. Cela signifie que les émotions sont créées par les perceptions et agissent ensuite sur les croyances, les désirs et les intentions. Enfin, ces états mentaux modifiés sont utilisés pour prendre une décision. Cette proposition n'a pas encore été intégrée à une plate-forme de simulation.

Enfin, certains chercheurs proposent de s'appuyer sur le contexte social de l'agent pour décrire son comportement : cela se fait avec des architectures normatives. EMIL-A [3] et NoA [23] décrivent le comportement de l'agent avec des normes, obligations et sanctions sociales. En d'autres termes, un agent prend une décision en fonction de l'état du système normatif au niveau d'une société d'agents.

BOID [13] et BRIDGE [16] proposent de combiner une architecture normative avec un paradigme BDI, conduisant l'agent à prendre en compte le système social dans sa décision. Cependant, contrairement à EMIL-A et NoA, l'agent a des croyances, des désirs et des intentions personnels, créant un comportement plus hétérogène et crédible. Mais, à notre connaissance, ces architectures n'ont pas été mises en œuvre en plate-forme de simulation pour traiter avec des milliers d'acteurs simulés.

2.3 Synthèse

Pour respecter le principe EROS, les modélisateurs ont besoin d'architectures proposant plusieurs dimensions psychologiques et sociales. Actuellement, comme le montre cette section, il n'existe pas d'architecture unique proposant à la fois des dimensions cognitives, affectives et sociales pour la simulation. Les seules tentatives, à notre connaissance, de combiner plus de deux traits ont utilisé la notion de personnalité, associant la cognition avec les émotions et la contagion émotionnelle [25] ou de combiner la cognition avec les émotions et les relations sociales [30].

Dans cet article, nous abordons cette question en proposant BEN (Behavior with Emotions and Norms), une architecture d'agent mettant en scène la cognition, les émotions, la personnalité, la contagion émotionnelle, les relations sociales et la gestion des normes. Pour le mettre en œuvre, nous avons basé notre travail sur l'architecture cognitive existante fournie par GAMA. Pour faciliter l'utilisation de BEN, nous l'avons implémenté en utilisant les principes de GAMA qui a prouvé sa facilité d'utilisation [26][34] grâce à son langage de modélisation GAML que nous avons étendu.

3 Formalisation des dimensions cognitives, affectives et sociales

L'architecture BEN se compose des notions telles que la cognition, la personnalité, les émotions, la contagion émotionnelle, les normes et les relations sociales pour décrire le comportement des agents dans le contexte d'une simulation sociale. Chacun de ces composants est représenté à l'aide d'un formalisme unifié pour assurer leur compatibilité. Ce formalisme est discuté plus en détails dans [10] et présenté dans des travaux précédents [11].

La partie principale de BEN est la cognition de l'agent. Un agent cognitif peut raisonner sur un ensemble de perceptions de son environnement et sur un ensemble de connaissances précédemment acquises. Dans BEN, cet environnement est représenté par le concept de prédicats. $\mathbf{P}_j(\mathbf{v})$ représente un prédicat de façon générale avec les éléments suivants :

- \mathbf{P} : l'identifiant du prédicat.
- \mathbf{j} : l'agent causant l'information.
- \mathbf{v} : l'ensemble de valeurs stockées dans le prédicat.

En fonction du contexte, cette représentation générale peut évoluer. \mathbf{P}_j représente une information sans valeur particulière attachée, $\mathbf{P}(\mathbf{v})$ représente une information qui n'est pas causée par un agent en particulier et \mathbf{P} représente une information sans valeur particulière et sans qu'un agent en soit la cause.

Par exemple, l'information qu'il y a un incendie dans l'environnement est représentée par le prédicat **fire**. Si ce feu est causé par l'agent *Bob*, le prédicat devient **fire_{Bob}**. Enfin, si cet incendie causé par Bob est à un endroit $(x:y)$, cette information est représentée par **fire_{Bob}(location : :(x,y))**.

3.1 Formalisme des notions cognitives

Avec BEN, un agent dispose d'états mentaux cognitifs sur lesquels il effectue son raisonnement. $\mathbf{M}_i(\mathbf{PMem}, \mathbf{Val}, \mathbf{Li})$ représente un état mental de façon générale de l'agent i avec les éléments suivants :

- \mathbf{M} : la modalité indiquant le type de l'état mental cognitif (par exemple une croyance).
- **PMem** : l'objet sur lequel porte l'état mental cognitif, pouvant être un prédicat, un autre état mental cognitif ou une émotion.
- **Val** : une valeur réelle dont la signification dépend de la modalité. Permet de comparer deux états mentaux cognitifs de même modalité.
- **Li** : une durée de vie indiquant le temps que mettra l'agent pour oublier cet état mental.

Dans BEN, la cognition s'appuie sur le modèle BDI [12] qui indique que l'agent possède des croyances, des désirs et des intentions. Pour lier la cognition avec les dimensions affectives et sociales, l'architecture

définit 6 modalités d'états mentaux cognitifs : **Belief_i(P)**, **Uncertainty_i(P)**, **Desire_i(P)**, **Intention_i(P)**, **Ideal_i(P)** et **Obligation_i(P)**.

Pour agir sur le monde selon ses intentions, un agent a besoin d'un plan d'actions, c'est-à-dire un ensemble de comportements exécutés dans un certain contexte en réponse à une intention.

3.2 Formalisme des dimensions affectives

La définition de traits de personnalité dans BEN suit le modèle OCEAN (Ouverture, Conscience, Extraversion, Agreabilité, Neurotisme) [27]. La personnalité d'un agent est représentée par un vecteur de 5 valeurs réelles comprises entre 0 et 1, avec la valeur neutre placée à 0.5.

Les émotions dans BEN sont basées sur la théorie OCC [31]; les émotions sont des réponses valuées à l'évaluation cognitive d'une situation. **Em_i(P,Ag,I,De)** représente une émotion possédée par l'agent *i* avec les éléments suivants :

- **Em** : le nom de l'émotion.
- **P** : le prédicat à propos duquel l'émotion est ressentie.
- **Ag** : l'agent responsable de l'émotion.
- **I** : l'intensité de l'émotion. Cette valeur est positive ou nulle.
- **De** : la valeur de décroissance de l'intensité émotionnelle.

Cette représentation permet à l'agent d'avoir plusieurs émotions en même temps, toutes sur des prédicats différents. Aussi, cette représentation peut être adaptée, avec **Em_i(P,Ag)** représentant une émotion sans intensité particulière ni valeur de déclin.

3.3 Formalisme des dimensions sociales

Chaque agent possède relations sociales avec d'autres agents simulant des acteurs humains en s'appuyant le travail de Svennevig [40] qui identifie quatre dimensions minimales pour décrire une relation sociale entre deux personnes. Dans BEN, la confiance est ajoutée comme cinquième dimension sociale. **R_{i,j}(L,D,S,F,T)** représente une relation sociale, de l'agent *i* vers agent *j*, avec les éléments suivants :

- **R** : l'identifiant de la relation sociale.
- **L** : une valeur réelle entre -1 et 1 représentant le degré d'appréciation avec l'agent concerné par le lien. Une valeur de -1 indique que l'agent *j* est détesté,

une valeur de 1 indique que l'agent *j* est adoré.

- **D** : une valeur réelle entre -1 et 1 représentant le degré de pouvoir exercé sur l'agent concerné par le lien. Une valeur de -1 indique que l'agent *j* domine la relation, une valeur de 1 indique que l'agent *j* est dominé dans la relation.
- **S** : une valeur réelle entre 0 et 1 représentant le degré de solidarité avec l'agent concerné par le lien. Une valeur de 0 indique qu'il n'y a aucune solidarité envers l'agent *j*, une valeur de 1 indique une solidarité totale avec l'agent *j*.
- **F** : une valeur réelle entre 0 et 1 représentant le degré de familiarité avec l'agent concerné par le lien. Une valeur de 0 indique qu'il n'y a aucune familiarité avec l'agent *j*, une valeur de 1 indique qu'il y a une familiarité totale avec l'agent *j*.
- **T** : une valeur réelle entre -1 et 1 représentant le degré de confiance avec l'agent concerné. Une valeur de -1 indique que l'agent *i* n'a aucune confiance envers l'agent *j*, une valeur de 1 indique une totale confiance envers l'agent *j*.

Pour agir sur l'environnement selon un système normatif, l'agent suit des normes, représentant dans BEN des séquences d'instructions auxquelles l'agent peut désobéir.

4 BEN : une architecture cognitive, affective et sociale

L'architecture BEN, représentée par la figure 1, est la principale contribution de cet article. Dans cette section, nous expliquons comment un agent utilisant BEN prend une décision avec la cognition, les émotions, la contagion émotionnelle, la personnalité, les relations sociales et les normes. Avec ces dimensions, un agent simulant un acteur peut réagir à un changement dans l'environnement et expliquer son comportement avec des concepts de haut niveau dans le cadre d'une simulation sociale. Par manque de place, le fonctionnement de l'architecture BEN est ici résumé ; une présentation détaillée est disponible dans [10].

4.1 Présentation globale de l'architecture

BEN est composé de quatre modules, chacun abritant plusieurs processus obligatoires (en lignes pleines) ou optionnels (en pointillés), gérés automatiquement (en bleu) ou devant être

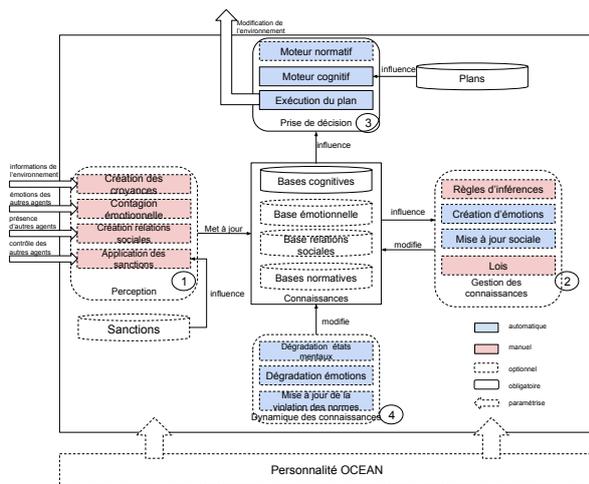


FIGURE 1 – L'architecture BEN

définis par l'utilisateur (en rose). Ces modules communiquent avec les bases de connaissances de l'agent et sont reliés à la personnalité de l'agent décrite par les cinq dimensions du modèle OCEAN [27].

Pour faciliter l'utilisation de BEN, ces traits de personnalité sont le seul paramètre auquel un modélisateur peut accéder ; ils sont utilisés pour calculer tous les autres paramètres nécessaires aux différents processus : probabilité de supprimer un plan ou une intention non réalisée dans la partie cognitive, charisme et réceptivité émotionnelle pour la contagion émotionnelle, intensité initiale et déclin des émotions créées par le moteur émotionnel, actualisation des valeurs des relations sociales obtenues avec le moteur social, et valeur d'obéissance utilisée par le moteur normatif.

La connaissance de l'agent est composée de bases cognitives, contenant des états mentaux cognitifs formalisés en section 3, une base émotionnelle, une base sociale et une base de normes. Cette connaissance peut évoluer à travers la simulation, ce qui n'est pas le cas des plans pour le moteur cognitif et des sanctions pour le moteur normatif, qui sont stockés dans des bases dédiées, hors de la connaissance de l'agent, comme le montre la figure 1.

4.2 Fonctionnement de l'architecture pour la prise de décision

Sur la figure 1, chaque module possède un numéro, indiquant son ordre d'exécution. Chaque

fois qu'un agent est activé, il perçoit l'environnement, il gère ses connaissances en fonction des nouvelles perceptions, il prend une décision et enfin il donne une dynamique temporelle à ses connaissances.

Perceptions. La première étape de BEN consiste à percevoir l'environnement. Cette étape permet de faire le lien entre l'environnement et les connaissances de l'agent, en créant des croyances et des incertitudes sur les informations de l'environnement, en définissant la contagion émotionnelle avec d'autres agents ou en créant de nouvelles relations sociales. Ces trois processus sont définis manuellement, ce qui signifie que le modélisateur doit indiquer quelle information est transformée en prédicat et quel état mental cognitif est construit sur ce prédicat, quelle émotion est soumise à une contagion émotionnelle et quelle est la valeur initiale pour chaque dimension d'une nouvelle relation sociale. Le dernier processus de ce module permet à un agent d'exécuter des sanctions sur les autres agents perçus.

Gestion des connaissances. La deuxième étape de BEN permet à l'agent de gérer ses connaissances après la perception et avant la prise de décision. Dans cette phase, les modélisateurs peuvent définir des règles d'inférence, qui permettent de créer ou de supprimer tout état mental cognitif en fonction de l'état réel des connaissances de l'agent. Sur le même modèle, des lois peuvent être définies pour créer des obligations si la valeur d'obéissance de l'agent, calculée à partir de sa personnalité, est suffisamment grande.

La gestion des connaissances passe aussi par l'utilisation d'un moteur émotionnel et d'un moteur de relations sociales. Ils permettent, respectivement, de créer automatiquement des émotions en fonction des connaissances de l'agent, et de mettre à jour les relations sociales avec les autres agents sans que l'utilisateur n'ait besoin d'intervenir dans ces processus.

Prise de décision. La troisième étape de BEN est la seule partie obligatoire. Ce module permet à l'agent de prendre des décisions puis d'exécuter une action, le tout au travers d'un moteur cognitif sur lequel un moteur normatif peut être ajouté. Il s'exécute automatiquement, sans intervention du modélisateur.

L'architecture cognitive est basée sur le modèle BDI [12] : l'agent a des intentions basées sur

ses désirs et une des intentions comme intention courante. Le modélisateur définit des plans d'action qui indiquent ce qu'un agent doit faire pour une intention courante particulière dans un contexte donné; le plan choisi est conservé comme le plan courant. Le moteur normatif fonctionne de la même manière que le moteur cognitif, avec des obligations comme désirs et des normes comme plans. La seule différence est une valeur d'obéissance qui peut être ajoutée aux normes et obligations.

Dynamisme temporel. La dernière partie de l'architecture donne une dynamique temporelle au comportement de l'agent. Ceci se fait automatiquement en dégradant les états mentaux cognitifs et les émotions et en mettant à jour l'état de chaque norme.

5 Application à l'évacuation d'une boîte de nuit

L'architecture définie dans la section 4 a été implémentée dans la plateforme de modélisation et de simulation GAMA [42], étendant le langage de programmation GAML pour aider les modélisateurs à définir des agents sociaux aux dimensions cognitives, affectives et sociales. Cette implémentation est utilisée dans le cas d'exemple d'une évacuation d'une boîte de nuit en feu détaillé dans cette section.

5.1 Présentation du cas d'exemple

Le 27 janvier 2013, le Kiss Nightclub de Santa Maria, Rio Grande do Sul (Brésil), a pris feu à la fin d'un spectacle dirigé par un groupe de musique local. Le plafond s'est enflammé à cause des feux d'artifice, émettant des fumées toxiques qui ont causé la mort de 242 personnes. L'enquête officielle a mis en lumière différents facteurs qui ont aggravé la tragédie : il y avait entre 1200 et 1400 personnes dans le bâtiment qui pouvaient normalement accueillir 641 personnes, il n'y avait qu'une seule porte d'entrée/sortie, il n'y avait ni détecteur de fumée ni alarme et enfin, les panneaux de sortie montraient la direction des toilettes. L'enquête montre également que la plupart des décès sont dus à l'asphyxie, près des toilettes [5].

Notre but ici est de reproduire le comportement des personnes prises dans cette tragédie de la manière la plus crédible possible. En d'autres termes, nous utilisons l'architecture BEN pour

créer les comportements des agents afin d'obtenir un résultat aussi proche que possible de ce qui s'est passé dans cette boîte de nuit pendant l'incendie.

5.2 Modélisation du cas avec BEN

Les connaissances initiales de l'agent, au lancement de la simulation, peuvent être divisées en trois types : les croyances sur le monde, les désirs initiaux et les relations sociales. De plus, chaque agent dispose d'une personnalité décrite par une valeur dans chacune des dimensions du modèle OCEAN. Quelques exemples de ces connaissances initiales ainsi que leur formalisation avec BEN sont données ci-dessous :

- Une croyance sur la localisation exacte de la porte de sortie avec une durée de vie fixée -> $Belief_i(exitDoor, lifetime1)$
- Le désir qu'il n'y ait pas de feu -> $Desire_i(notFire, 1.0)$
- Une relation d'amitié avec un agent j -> $R_{i,j}(L, D, S, F, T)$

Le premier processus de BEN est la perception de l'environnement, qui sera réduite en fonction de la fumée entourant l'agent. Il faut donc indiquer ce que l'agent perçoit et comment ces perceptions influencent ses connaissances, ce qui est montré dans un cas pratique par le bloc de code 1 :

- Percevoir la porte de sortie met à jour les croyances liées à celle-ci.
- Percevoir le feu ajoute la croyance qu'il y a un feu.
- Percevoir la fumée ajoute une croyance liée au niveau de fumée détectée.
- Percevoir les autres agents permet la création de relations sociales avec eux. Une contagion émotionnelle portant sur la peur de l'incendie est aussi définie.

```
perceive target:exit
in:perceived_area{
  focus id:"exitLocation"
  lifetime: 20;
}

perceive target:fire
in:perceived_area{
  focus id:"fire";
}
```

Code 1: Exemple de définition de perceptions dans le cas du Kiss Nightclub

Une fois que les connaissances de l'agent sont à jour avec son environnement, elles vont réagir à ce qui a été perçu. Cela se fait avec la définition des règles d'inférence et des lois, dont le bloc de code 2 montre une implémentation pratique :

- Une loi crée l'obligation de suivre les panneaux de sortie s'il y a un doute raisonnable (modélisé par la valeur d'obéissance attachée à la loi et la quantité de fumée perçue) de catastrophe.
- Une règle d'inférence ajoute le désir de fuir si l'agent a la croyance qu'il y a un incendie.
- Une règle d'inférence ajoute une incertitude qu'il y a un incendie si l'agent a la croyance qu'il y a de la fumée.
- Une règle d'inférence ajoute le désir de fuir si l'agent a une émotion de peur portant sur l'incendie si son intensité est supérieure à un seuil donné.

```
rule belief: fireSaw
  new_desire:fleeing
  remove_intention:wandering
  remove_desire:wandering ;
rule belief: smokeSaw
  new_uncertainty:fireSaw
  strength: smokeQuantity/100
  when: not has_belief(fireSaw);
```

Code 2: Exemple de définition de lois et de règles d'inférence dans le cas du Kiss Nightclub

Avec l'exécution des règles d'inférence et des lois, chaque agent crée des émotions par le biais du moteur émotionnel. Dans ce cas d'exemple, la présence d'une incertitude sur l'incendie (ajoutée par la règle d'inférence concernant la croyance sur la fumée) avec le désir initial qu'il n'y a pas de feu produit une émotion de peur, dont l'intensité est calculée en fonction de la quantité de fumée perçue.

Une fois que l'agent a le désir de fuir (parce qu'il a perçu l'incendie ou sa peur d'un incendie est d'une intensité assez grande), il a besoin de plans d'action et de normes qui lui indiquent comment agir. La définition de quelques plans d'action et normes pour répondre à l'intention de fuir est donnée ci-dessous, avec le contexte de déclenchement et le résultat de l'action :

- l'agent a une bonne visibilité et possède une croyance sur la position exacte de la sortie -> un plan indique à l'agent de courir vers la porte de sortie en suivant le plus court chemin.

- l'agent a une bonne visibilité mais aucune croyance sur l'emplacement de la sortie -> une norme amène l'agent à suivre les agents autour de lui en qui il a confiance.
- l'agent a une mauvaise visibilité et possède l'obligation de suivre les panneaux de sortie -> une norme indique à l'agent de suivre les panneaux de sortie, ce qui l'amène dans ce cas vers les toilettes.
- l'agent a une mauvaise visibilité et la croyance que les panneaux de sortie sont erronés -> un plan amène l'agent à se déplacer au hasard.
- l'agent perçoit un ami perdu dans la fumée -> un plan indique à l'agent d'aller trouver son ami pour l'amener vers la sortie.

Au fur et à mesure que la situation évolue au cours de la simulation, un agent peut modifier son plan en cours. Par exemple, si un agent quitte la zone enfumée alors qu'il fuit vers les toilettes, il peut percevoir la sortie et s'y rendre au lieu de continuer à suivre les panneaux.

5.3 Résultats et discussion

Au début de la simulation, les agents sont placés au hasard dans le Kiss Nightclub recréé avec une personnalité initialisée par une distribution gaussienne centrée sur 0,5 et avec un écart type de 0,12 pour chaque dimension. La propagation de la fumée est modélisée selon un rapport officiel du gouvernement français [14]; un agent est considéré mort après 50 secondes dans la fumée dense.

La figure 2 montre un résultat visuel de la simulation où les lignes noires représentent les murs de la discothèque, les carrés gris représentent la fumée et les triangles représentent les acteurs simulés, la couleur de chaque triangle indique le plan suivi. Une vidéo de la simulation ainsi que le modèle complet se trouvent à l'adresse suivante :

<https://github.com/mathieuBourgais/ExempleThese>

Comme le nombre exact de personnes dans la boîte de nuit n'est pas connu, nous avons testé trois cas : 1200 personnes au début, 1300 personnes au début et 1400 personnes au début. Les résultats statistiques obtenus dans le tableau 1 sont calculés à partir de 10 simulations pour chaque scénario.

Les résultats statistiques indiquent que notre modèle est bien calibré pour reproduire le cas réel

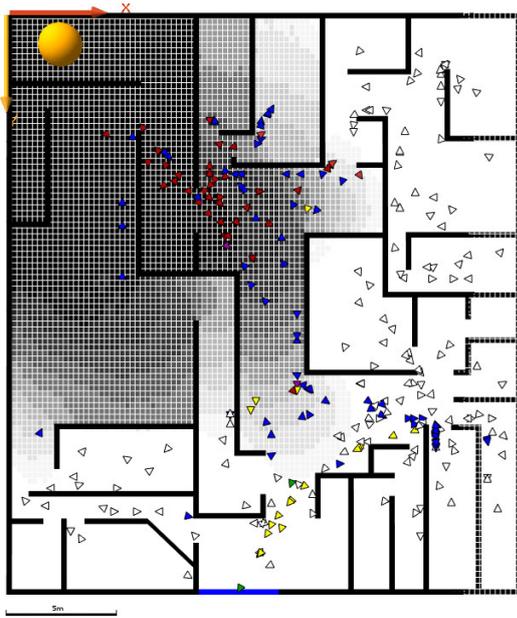


FIGURE 2 – Simulation de l'évacuation du Kiss Nightclub

TABLE 1 – Nombre d'agents décédés dans la simulation de l'incendie du Kiss Nightclub

Nombre d'agents	1200	1300	1400
moyenne	230.2	237.7	249.4
écart type	20.1	15.6	32.6

où 242 personnes sont décédées. Cependant, le principal résultat concerne l'explicabilité et l'expressivité du modèle. La vidéo de la simulation montre différents types de comportement qui peuvent être exprimés avec des concepts de haut niveau grâce à BEN.

Par exemple, beaucoup d'agents quittent le club au début de la simulation parce qu'ils ont directement perçu l'incendie. Ce comportement semble correspondre à un cas réel où les gens qui voient un incendie dans une boîte de nuit s'enfuient. Par contre, les agents qui ne perçoivent pas la fumée ou l'incendie fuient plus tard. Pendant ce temps, ils ont oublié l'emplacement de la sortie et ont dû suivre les panneaux de sortie officiels, ce qui les a conduit, dans ce cas, aux toilettes.

Grâce à l'architecture BEN, nous avons pu traduire un comportement exprimé en langage commun en un comportement réel pour des acteurs simulés. A tout moment, il est possible de mettre en pause la simulation pour inspecter le comportement d'un agent; ce comportement sera exprimé en termes d'états mentaux cognitifs,

d'émotions, de relations sociales, de normes et de plans qui sont, de notre point de vue, plus faciles à lire et à comprendre que des équations abstraites. Ce point est soutenu par le fait que le BEN et sa partie cognitive reposent sur la psychologie populaire [29].

6 Conclusion

Cet article présente BEN, une architecture comportementale et modulaire pour la prise de décision d'agents sociaux dans la simulation. Cette architecture s'appuie sur une formalisation des notions de cognition, d'émotions, de personnalité, de contagion émotionnelle, de relations sociales et de systèmes normatifs leur permettant d'interagir entre elles dans un processus de prise de décision amenant à des résultats crédibles en simulation de situations impliquant des acteurs humains. Cette architecture a été implémentée dans la plateforme de modélisation et de simulation multi-agent GAMA [42] et mise en oeuvre sur le cas d'étude de l'évacuation d'une discothèque en feu au Brésil, permettant de reproduire de façon crédible une situation réelle, tout en utilisant les mêmes composants de haut niveau utilisés pour décrire le comportement d'acteurs humains.

BEN a déjà été utilisé, tout ou en partie, sur d'autres projets par d'autres chercheurs : l'étude de l'évacuation d'une discothèque aux États-Unis [44], l'étude d'évacuation de zones ouvertes lors d'incendies de brousse en Australie [2], l'étude de comportements de foyers sur leur consommation d'énergie [28] ou encore l'étude de fermiers vietnamiens dans le delta du Mékong [43]. Ces différents travaux montrent que la modularité de BEN lui permet de s'adapter à différents cas d'étude et d'être utilisé par des chercheurs spécialistes de leur domaine n'ayant pas pour autant des compétences expertes en programmation.

Références

- [1] Carole Adam and Benoit Gaudou. Bdi agents in social simulations : a survey. *The Knowledge Engineering Review*, 2016.
- [2] Carole Adam, Patrick Taillandier, Julie Dugdale, and Benoit Gaudou. Bdi vs fsm agents in social simulations for raising awareness in disasters : A case study in Melbourne bushfires. *International Journal of Information Systems for Crisis Response*

- and Management (IJISCRAM)*, 9(1) :27–44, 2017.
- [3] Giulia Andrighetto, Rosaria Conte, Paolo Turrini, and Mario Paolucci. Emergence in the loop : Simulating the two way dynamics of norm innovation. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2007.
- [4] Magda B Arnold. *Emotion and personality*. Columbia University Press, 1960.
- [5] B Atiyeh. Brazilian kiss nightclub disaster. *Annals of burns and fire disasters*, 26(1) :3, 2013.
- [6] Robert Axelrod. Advancing the art of simulation in the social sciences. In *Simulating social phenomena*, pages 21–40. Springer, 1997.
- [7] Tina Balke and Nigel Gilbert. How do agents make decisions ? a survey. *Journal of Artificial Societies and Social Simulation*, 17(4) :13, 2014.
- [8] Michael Balmer, Marcel Rieser, Konrad Meister, David Charypar, Nicolas Lefebvre, Kai Nagel, and K Axhausen. Matsim-t : Architecture and simulation times. *Multi-agent systems for traffic and transportation engineering*, 2009.
- [9] M Bourgais, P Taillandier, and Laurent Vercoüter. Cognition, émotions et relations sociales pour la simulation multi-agent. In *JFSMA 2017*, Caen, France, July 2017.
- [10] Mathieu Bourgais. *Vers des agents cognitifs, affectifs et sociaux pour la simulation*. Theses, Normandie Université, 2018.
- [11] Mathieu Bourgais, Patrick Taillandier, and Laurent Vercoüter. Ben : An agent architecture for explainable and expressive behavior in social simulation. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 147–163. Springer, 2019.
- [12] M Bratman. *Intentions, plans, and practical reason*. Harvard Univ. Press, 1987.
- [13] Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, and Leendert van der Torre. The boid architecture : conflicts between beliefs, obligations, intentions and desires. In *Proceedings of the fifth international conference on Autonomous agents*, pages 9–16. ACM, 2001.
- [14] C. Chivas and J. Cescon. Formalisation du savoir et des outils dans le domaine des risques majeurs (dra-35) - toxicité et dispersion des fumées d’incendie phénoménologie et modélisation des effets. Technical report, INERIS, 2005.
- [15] Philip R Cohen and Hector J Levesque. Intention is choice with commitment. *Artificial intelligence*, 42(2-3) :213–261, 1990.
- [16] Frank Dignum, Virginia Dignum, and Catholijn M Jonker. Towards agents for policy making. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 141–153. Springer, 2008.
- [17] Bruce Edmonds and Scott Moss. From kiss to kids—an ‘anti-simplistic’ modelling approach. In *International workshop on multi-agent systems and agent-based simulation*, pages 130–144. Springer, 2004.
- [18] Nigel Gilbert and Klaus Troitzsch. *Simulation for the social scientist*. McGraw-Hill Education (UK), 2005.
- [19] Jonathan Gratch and Stacy Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 2004.
- [20] Nick Howden, Ralph Rönquist, Andrew Hodgson, and Andrew Lucas. Jack intelligent agents-summary of an agent infrastructure. In *5th International conference on autonomous agents*, 2001.
- [21] Wander Jager. Enhancing the realism of simulation (eros) : On implementing and developing psychological theory in social simulation. *Journal of Artificial Societies and Social Simulation*, 20(3) :14, 2017.
- [22] Hong Jiang, Jose M Vidal, and Michael N Huhns. Ebdi : an architecture for emotional agents. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 2007.
- [23] Martin Josef Kollingbaum. *Norm-governed practical reasoning agents*. PhD thesis, University of Aberdeen, 2005.
- [24] Kalliopi Kravari and Nick Bassiliades. A survey of agent platforms. *Journal of Artificial Societies and Social Simulation*, 18(1) :11, 2015.
- [25] Margaux Lhommet, Domitile Lourdeaux, and Jean-Paul Barthès. Never alone in the crowd : A microscopic crowd model based on emotional contagion. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 2, pages 89–92. IEEE, 2011.

- [26] EG Macatulad and AC Blanco. 3dgis-based multi-agent geosimulation and visualization of building evacuation using gama platform. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2014.
- [27] Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. *Journal of personality*, 1992.
- [28] Alice Micolier, Franck Taillandier, Patrick Taillandier, and Frédéric Bos. Libim, an agent-based approach to simulate occupant-building interaction from the building-information modelling. *Engineering Applications of Artificial Intelligence*, 82 :44–59, 2019.
- [29] Emma Norling. Folk psychology for human modelling : Extending the bdi paradigm. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 202–209, 2004.
- [30] Magalie Ochs, Nicolas Sabouret, and Vincent Corruble. Simulation of the dynamics of nonplayer characters' emotions and social relations in games. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(4) :281–297, 2009.
- [31] Andrew Ortony, Gerald L Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.
- [32] Lin Padgham, Kai Nagel, Dharendra Singh, and Qingyu Chen. Integrating bdi agents into a matsim simulation. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*, 2014.
- [33] Alexander Pokahr, Lars Braubach, and Winfried Lamersdorf. Jadex : A bdi reasoning engine. In *Multi-agent programming*. Springer, 2005.
- [34] Nur Raihan Ramli, Sazalinsyah Razali, and Mashanum Osman. An overview of simulation software for non-experts to perform multi-robot experiments. In *ISAMSR*. IEEE, 2015.
- [35] Ralph Rönquist. The goal oriented teams (gorite) framework. In *International Workshop on Programming Multi-Agent Systems*. Springer, 2007.
- [36] Ilias Sakellariou, Petros Kefalas, and Ioanna Stamatopoulou. Enhancing netlogo to simulate bdi communicating agents. In *Hellenic Conference on Artificial Intelligence*. Springer, 2008.
- [37] Dharendra Singh and Lin Padgham. Opensim : A framework for integrating agent-based models and simulation components. In *Frontiers in Artificial Intelligence and Applications-Volume 263 : ECAI*, 2014.
- [38] Craig A Smith, Richard S Lazarus, et al. Emotion and adaptation. *Handbook of personality : Theory and research*, pages 609–637, 1990.
- [39] Ron Sun. *Cognition and multi-agent interaction : From cognitive modeling to social simulation*. Cambridge Univ. Press, 2006.
- [40] Jan Svennevig. *Getting acquainted in conversation : a study of initial interactions*. John Benjamins Publishing, 2000.
- [41] Patrick Taillandier, Mathieu Bourgais, Philippe Caillou, Carole Adam, and Benoit Gaudou. A bdi agent architecture for the gama modeling and simulation platform. In *MABS 2016*, 2016.
- [42] Patrick Taillandier, Benoit Gaudou, Arnaud Grignard, Quang-Nghi Huynh, Nicolas Marilleau, Philippe Caillou, Damien Philippon, and Alexis Drogoul. Building, composing and experimenting complex spatial models with the gama platform. *GeoInformatica*, Dec 2018.
- [43] Quang Chi Truong, Patrick Taillandier, Benoit Gaudou, Minh Quang Vo, Trung Hieu Nguyen, and Alexis Drogoul. Exploring agent architectures for farmer behavior in land-use change. a case study in coastal area of the vietnamese mekong delta. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 146–158. Springer, 2015.
- [44] Marion Valette, Benoit Gaudou, Dominique Longin, and Patrick Taillandier. Modeling a real-case situation of egress using bdi agents with emotions and social skills. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 3–18. Springer, 2018.
- [45] H. Van Dyke Parunak, Robert Bisson, Sven Brueckner, Robert Matthews, and John Sauter. A model of emotions for situated agents. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006.
- [46] Uri Wilensky and I Evanston. Netlogo : Center for connected learning and computer-based modeling. *Northwestern Univ., Evanston, IL*, 1999.

Apprentissage adaptatif de comportements éthiques

R. Chaput¹, O. Boissier², M. Guillermin³, S. Hassas¹

¹ Univ. Lyon, Université Lyon 1, LIRIS, UMR5205, F-69622, LYON, France

² Univ. Lyon, IMT Mines Saint-Étienne, CNRS, Laboratoire Hubert Curien UMR 5516

³ GEEST, UR Confluence, Sciences et Humanités, Université Catholique de Lyon

{remy.chaput,salima.hassas}@univ-lyon1, Olivier.Boissier@emse.fr, mguillermin@univ-catholyon.fr

Résumé

L'utilisation croissante d'algorithmes d'Intelligence Artificielle (IA) dans des applications impactant des humains requiert de doter ces systèmes d'un comportement pouvant être jugé éthique selon des valeurs humaines. Bien que plusieurs approches existent, la question de l'adaptation au contexte, aux préférences et principes éthiques des utilisateurs, reste posée. Nous proposons de traiter cette question par l'Apprentissage par Renforcement Multi-Agent de tels comportements dans des situations différentes. Nous utilisons des tables de Q-Valeurs et des Cartes Auto-Organisatrices Dynamiques pour permettre l'apprentissage adaptatif de la représentation de l'état de l'environnement, ainsi que des fonctions de récompense pour guider l'éthique du comportement. Cette proposition est évaluée sur un simulateur de répartition d'énergie dans des Smart Grids que nous avons développé. Plusieurs fonctions de récompense visant à déclencher des comportements éthiques sont évaluées. Les résultats montrent la capacité de s'adapter à différentes conditions. En sus des contributions sur le plan de l'adaptation éthique, nous comparons notre modèle à d'autres approches d'apprentissage et montrons de meilleures performances par rapport à une approche d'Apprentissage Profond basée sur le modèle Actor-Critic.

Mots-clés

Éthique, Apprentissage par renforcement, Systèmes Multi-Agent, Répartition de l'énergie.

Abstract

The increase in the use of Artificial Intelligence (AI) algorithms in applications impacting human users and actors has, as a direct consequence, the need for endowing these AI systems with ethical behaviors. While several approaches already exist, the question of adaptability to changes in contexts, users behaviors or preferences still remains open. We propose to tackle this question using Multi-Agent Reinforcement Learning of ethical behavior in different situations using Q-Tables and Dynamic Self-Organizing Maps to allow dynamic learning of the representation of the environment's state and reward functions to prescribe ethical behaviors. To evaluate this proposal, we

developed a simulator of intelligent management of energy distribution in Smart Grids, evaluating different rewards functions to trigger ethical behaviors. Results show the ability to adapt to different conditions. Besides contributions on ethical adaptation, we compare our model to other learning approaches and show it performs better than a Deep Learning one (based on Actor-Critic).

Keywords

Ethics, Reinforcement Learning, Multi-Agent Systems, Energy management.

1 Introduction

Les récents progrès dans le domaine de l'Intelligence Artificielle (IA) ont mené à une augmentation rapide de l'utilisation d'algorithmes d'IA dans des applications ayant un impact potentiel sur des acteurs et utilisateurs humains. De tels systèmes incluent, par exemple, le *trading* automatique, la conduite autonome ou assistée, l'allocation de ressources, etc.

Moor [14] affirme que la plupart des applications de nos jours ont des conséquences éthiques, étant donné que leurs actions peuvent causer du tort ou des bénéfices aux humains. Il pointe également le besoin pour ces applications ou agents artificiels d'être conçus avec, ou d'intégrer, des considérations éthiques plus approfondies. Plusieurs approches ont ainsi été proposées (voir section 2). Bien qu'elles soient suffisamment génériques pour gérer une variété de situations, la question de leur capacité à s'adapter à des changements de situations en cours d'exécution reste encore ouverte.

Dans cet article, nous proposons un nouveau système qui apprend dynamiquement des comportements perçus comme éthiques, reflétant des valeurs humaines. Cette approche utilise des Cartes Auto-Organisatrices Dynamiques (*Dynamic Self-Organizing Maps* - DSOMs) pour apprendre la représentation de l'espace d'entrée et discrétiser l'espace d'action, ainsi que des tables de Q-Valeurs (*Q-Tables*) pour mémoriser l'intérêt des actions, avec des fonctions de récompense spécifiquement construites pour guider l'agent vers un comportement éthique ; ces algorithmes permettent l'adaptation quand l'environnement change dynamiquement.

Dans cet article, nous ne ferons pas l'hypothèse que les agents artificiels en eux-mêmes peuvent être considérés comme des "preneurs de décision éthique". Nous utiliserons néanmoins le raccourci "comportement éthique" pour décrire le comportement d'un agent qu'un humain jugerait éthique selon ses valeurs ou principes. Ainsi, notre but est de faire apprendre à ces agents artificiels de tels "comportements éthiques".

Nous présentons dans un premier temps un rapide horizon des approches existantes dans la section 2 et pointons leurs limites quant à l'adaptabilité de leurs comportements éthiques. Dans un second temps, nous présentons en section 3 le modèle éthique et l'algorithme d'apprentissage sur lesquels est basée notre approche. L'évaluation de ce modèle dans le contexte des *Smart Grids* est décrit en section 4. En section 5, nous présentons nos résultats et les comparons à ceux d'autres modèles d'apprentissage. Finalement, nous discutons des avantages et limitations de notre modèle, et présentons des perspectives dans la section 6.

2 État de l'art

Depuis les années 2000, plusieurs groupes de travaux discutent de la capacité d'agents artificiels à montrer un comportement éthique, i.e. qui soit compatible avec des valeurs humaines (e.g. Machine Ethics [3], Machine Morality [17], Moral AI [8]). Dans [10, 9], les relations entre IA et éthique sont présentées autour des niveaux suivants : éthique pour la conception (*ethics for design*), éthique dans la conception (*ethics in design*), éthique par conception (*ethics by design*). Par la suite, nous nous concentrons sur les approches relatives à l'éthique par conception, les deux autres niveaux se rapportant à l'aspect éthique du développement et des conséquences des systèmes artificiels, plutôt qu'au contenu explicitement éthique intégré dans ces systèmes. L'éthique par conception rassemble les travaux visant à intégrer des capacités de raisonnement éthique comme partie intégrante de la production du comportement de l'agent [9]. Ces travaux consistent à résoudre une tâche en prenant en compte des considérations éthiques, e.g. prendre une décision ayant des conséquences sur les humains en considérant un sous-ensemble de valeurs, ou déterminer si une action est acceptable ou non selon un principe éthique. Pour ce faire, ils utilisent différentes techniques d'IA, que nous analysons par rapport à leur capacité d'adaptation face aux environnements dynamiques, en considérant les approches d'éthique implicite d'une part et d'éthique explicite d'autre part [14].

Les agents à éthique implicite s'appuient sur des comportements éthiques pré-codés par les concepteurs, associés à des situations particulières. Ainsi, l'agent ne raisonne pas sur l'éthique et ne peut s'adapter aux changements de situation non prévus par les concepteurs. Nous pouvons citer comme exemple d'un tel système [6], qui décrit un *Ethical Governor* pour des drones létaux autonomes.

Les agents à éthique explicite possèdent des règles générales, peuvent raisonner sur l'éthique et dans la plupart des cas justifier leurs décisions. Ils introduisent l'éthique

par raisonnement, qui vise à décider du comportement des agents à partir d'un ensemble de règles explicites, qui peuvent être distinguées entre les règles et valeurs morales (générales, e.g. l'Impératif Catégorique de Kant), et les principes éthiques (plus spécifiques et appliqués, e.g. la transparence des algorithmes), qui évaluent la dimension morale (ou éthique) d'actions dans un contexte donné. Ces approches sont également nommées *Top-Down* par [2]. Elles incluent par exemple le travail de [7] sur le jugement et la confiance dans les Systèmes Multi-Agents, en utilisant le *trading* comme cas d'application, ou le travail de [18] sur l'architecture cognitive LIDA dans le cadre de la délibération morale. De telles approches permettent de s'adapter et raisonner sur la dimension éthique du comportement selon la situation. Toutefois, les règles morales, les valeurs morales et les principes éthiques sont fixés et définis par les concepteurs du système à l'instanciation, ce qui empêche l'adaptation des agents *in situ*.

L'éthique par apprentissage franchit une étape de plus dans l'adaptation en visant à apprendre la mise en relation entre les situations et les règles morales ou principes éthiques. Ces approches sont également nommées *Bottom-Up* par [2] : les règles sont extraites à partir des cas. Cela inclut le travail de [4] sur la conduite autonome, en utilisant des jugements d'experts pour composer un jeu de données de situations éthiques, ou encore le travail de [20], utilisant l'apprentissage par renforcement avec une divergence entre l'agent et un comportement humain normalisé comme récompense éthique. De telles approches peuvent s'adapter si les données de départ sont modifiées, mais l'adaptation durant le fonctionnement n'a pas été explorée. L'apprentissage se fait hors-ligne. Elles souffrent de ce fait des mêmes limitations que l'éthique par raisonnement, par rapport à l'adaptabilité.

Le lecteur intéressé est invité à lire [22] pour une étude plus complète sur ce sujet. Tandis que la plupart de ces travaux considèrent un seul agent, peu adoptent une perspective multi-agent, introduisant par exemple la capacité de traiter des conflits entre les normes sociales et les valeurs de plusieurs acteurs, e.g. le travail de [1] sur les SI-PAs (*Socially Intelligent Personal Agents*), ou l'influence de plusieurs agents interagissant dans un environnement partagé. Ce sont d'importantes dimensions à considérer dans les applications à venir, particulièrement du fait que chaque agent doit s'adapter aux autres, à la fois en termes de normes sociales mais aussi d'interactions.

Il pourrait être intéressant d'augmenter l'adaptabilité des agents artificiels aux changements environnementaux en ligne, ou même aux changements au sein de l'environnement physique et culturel, impactant les différentes sortes de considérations éthiques.

3 Modèle

Afin de surmonter les limitations précédemment mentionnées, nous proposons un nouveau modèle basé sur de l'Apprentissage par Renforcement Multi-Agent.

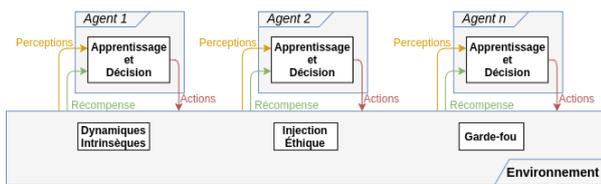


FIGURE 1 – Modèle d'apprentissage de comportements éthiques dans un contexte multi-agent : plusieurs agents interagissent et perçoivent un environnement partagé. Le processus d'apprentissage et de décision de chaque agent s'appuie sur des réceptacles éthiques, qui les rendent capables de traiter les injections éthiques incluses dans la récompense donnée par l'environnement.

3.1 Modèle éthique

Notre modèle éthique est intégré dans un Système Multi-Agent (SMA), dont le but principal est de réaliser une tâche tout en assurant un comportement éthique. Les agents sont composés d'un ensemble de structures de données et algorithmes (groupés dans le cadre "Apprentissage et Décision" dans la figure 1) qui traitent les perceptions pour produire des actions. Ces actions forment un comportement que l'on souhaite éthique.

Pour ce faire, il faut donner une orientation éthique à l'apprentissage. De manière abstraite, on nommera cet apport injection éthique¹. Cette injection éthique regroupe les considérations éthiques explicites que les concepteurs du système voudraient injecter dans le système, et plus particulièrement dans les agents. Elle peut être située dans l'agent directement (par des connaissances explicites), ou dans l'environnement.

Dans notre cas, notre but étant que les agents apprennent ces considérations et puissent s'adapter aux changements, nous choisissons de la placer dans l'environnement. Elle est rendue accessible aux agents par le biais d'une fonction de récompense, reflétant les valeurs éthiques qui sont considérées importantes par les concepteurs. La récompense est donc calculée par l'environnement puis fournie aux agents; les agents ignorent donc la façon dont la récompense est calculée, ce qui permet de la modifier si besoin (sans avoir besoin de modifier l'agent).

Le modèle, composé de l'environnement et des agents, et enrichi de l'injection éthique, est également enrichi d'un garde-fou. Il correspond aux limites que les concepteurs ne veulent pas que le système franchisse (e.g. parce qu'elles représentent une impossibilité physique, ou une situation non désirable). De même que pour l'injection éthique, le garde-fou peut être situé dans différents modules : il peut s'agir de contraintes imposées par l'environnement, ou par des agents à travers des interactions avec les autres

1. Il est important de distinguer l'injection éthique des perceptions et actions de l'agent, même si certaines d'entre elles (comme dans notre cas l'équité, voir section 4) possèdent une signification éthique pour le concepteur ou l'humain. Les perceptions et actions ne permettent pas d'orienter le comportement de l'agent vers un comportement éthique. Cette direction est donnée par l'injection éthique, qui représente les considérations éthiques que l'agent doit avoir ou apprendre.

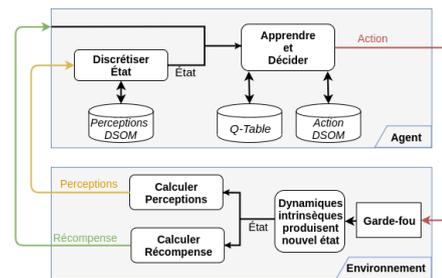


FIGURE 2 – Le cycle d'apprentissage, considérant un unique agent interagissant avec l'environnement.

agents. Selon cette définition, le garde-fou n'influence pas l'apprentissage, et nous le distinguons donc de l'injection éthique, dans le sens où il impose des contraintes, tandis que l'injection éthique guide la décision. Idéalement, les agents devraient apprendre à ne pas franchir ces limites. Dans notre approche actuelle, nous choisissons d'intégrer le garde-fou à l'environnement.

L'utilisation de plusieurs agents permet des cas d'utilisation et des tâches avec aspect éthique plus riches, du fait que les agents peuvent avoir des intérêts divergents, soulevant ainsi des conflits éthiques et éventuellement des dilemmes. De plus, les agents pourraient interagir entre eux, et notamment servir d'injection éthique ou de garde-fou : un agent pourrait émettre un jugement à l'encontre d'un autre agent, fournissant alors une récompense ou bien une contrainte (e.g. en interdisant une action proposée). Pour simplifier dans un premier temps, de telles interactions entre agents ne sont pas étudiées ici.

3.2 Processus d'apprentissage

En suivant le modèle introduit ci-dessus, nous nous intéressons maintenant aux capacités d'apprentissage et de décision des agents, étant donné les perceptions de l'environnement dans lequel ils sont situés. Ce processus de décision et apprentissage est basé sur l'Apprentissage par Renforcement (*Reinforcement Learning* - RL). L'injection éthique de notre modèle est mise en place dans la génération de récompenses pour le renforcement de la production de comportements éthiques. Les données en entrée de chaque agent sont des propriétés calculées de l'environnement (telle qu'une métrique "sur-consommation"). Comme nous le verrons dans la section suivante, ces métriques sont spécifiquement conçues selon le cas d'application. Les structures et algorithmes inclus dans les agents (voir figure 2) et qui supportent ce processus d'apprentissage sont une table de Q-Valeurs (*Q-Table*) et deux Cartes Auto-Organisatrices Dynamiques (*Dynamic Self-Organizing Maps* - DSOM) : la DSOM de perceptions (P-DSOM) et la DSOM d'actions (A-DSOM). Ils représentent les réceptacles éthiques de par leur capacité à générer les comportements éthiques, telles que la représentation des états, des actions et la politique de choix d'une action dans un état donné.

Formellement, ce système peut être modélisé comme un Processus de Décision Markovien (MDP), dans lequel l'en-

semble des états S consiste en un ensemble d'états multi-dimensionnels et continus représentés par des perceptions de l'environnement.

L'ensemble des actions A est un ensemble où chaque action $a \in A$ est un vecteur de paramètres continus. Nous imposons cette contrainte afin de permettre des environnements et cas applicatifs plus riches, en permettant des actions flexibles (par opposition aux actions discrètes).

Afin de prendre en compte le *Multi-Agent Credit Assignment Problem*², la fonction de récompense R est calculée sous la forme de *Difference Rewards* [21] : nous comparons l'état actuel de l'environnement avec un environnement hypothétique dans lequel l'agent n'aurait pas agi. Formellement, considérant une fonction $V : A^n \rightarrow [0, 1]$ qui évalue un ensemble d'actions, la récompense d'un agent i est : $R_i = V(\text{Actions}) - V(\text{Actions} \setminus \{\text{Actions}_i\})$. Intuitivement, l'agent obtient une récompense positive si son action améliore l'état de l'environnement, et négative sinon. Nous rappelons que cette fonction de récompense R est la même pour tous les agents (elle est spécifiée dans l'environnement), mais la récompense est calculée de manière individuelle par l'environnement pour chaque agent. L'environnement hypothétique est donc calculé dans l'environnement ; les agents n'ont pas accès aux actions des autres agents, mais seulement à la récompense calculée R_i .

Afin de manipuler les états et actions, nous utilisons l'algorithme *Q-Learning* [19] augmenté de deux DSOMs [15] (S est appris par la P-DSOM, tandis que A est appris par l'A-DSOM). Les DSOMs sont des grilles 2D de neurones, ayant un vecteur associé (de perceptions ou paramètres d'actions dans notre cas), et sont liées aux états et actions discrets dans la *Q-Table*. Elles sont particulièrement appropriées à notre problème d'adaptation grâce à leur hyper-paramètre d'élasticité η , qui remplace le mécanisme habituel de décroissance du voisinage, afin de permettre à la carte de rester stable quand les données ne changent pas, et de s'adapter quand un changement est détecté. Le seuil de détection de changement est directement déterminé par le coefficient d'élasticité introduit : la fonction de voisinage, paramétrée par η , détermine si le neurone le plus proche de la donnée en entrée est suffisamment proche. Si ce n'est pas le cas, tous les neurones sont mis à jour (la carte s'adapte donc au changement). Cette dynamique de stabilité/adaptation est obtenue en imposant un couplage plus ou moins resserré aux neurones. Cette approche est une extension de [16], où nous remplaçons les Cartes Auto-Organisatrices de Kohonen [11] par les DSOMs, ainsi que la politique d'exploration ϵ -cupide par une politique Boltzmann (ligne 2 de l'algorithme 1 ; le paramètre de température τ contrôle le dilemme exploration-exploitation).

Le processus de décision (voir algorithme 1 et figure 3) utilise la P-DSOM pour discrétiser les perceptions p en une hypothèse d'état i (ligne 1). Cet état est utilisé pour consulter la *Q-Table* et sélectionner un index d'action j selon une distribution de probabilité Boltzmann (ligne 2). Les

2. Chaque agent doit recevoir une récompense proportionnelle à sa contribution dans la solution globale.

Algorithm 1 Partie décision du processus d'apprentissage

Variables partagées : U l'ensemble des neurones de la P-DSOM, W l'ensemble des neurones de l'A-DSOM, et Q la Q-Table

Hyper-paramètres : τ température de Boltzmann

Entrée : p les perceptions

Sortie : a les paramètres d'actions

- 1: Déterminer le plus proche neurone $i := \arg \min \|p - U_i\|$
 - 2: Choisir index d'action j avec probabilité $P(j) := \frac{e^{\frac{Q_{i,j}}{\tau}}}{\sum_k e^{\frac{Q_{i,k}}{\tau}}}$
 - 3: Soient les paramètres d'actions $:= W_j$
 - 4: **for all** dimension k de W_j **do**
 - 5: Ajouter bruit uniforme : $W'_{j,k} := W_{j,k} + \text{rand}(-\epsilon, +\epsilon)$
 - 6: **end for**
 - 7: **return** les paramètres d'action perturbés $a := W'_j$
-

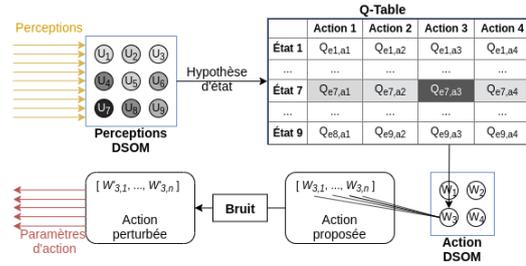


FIGURE 3 – Processus de décision, déterminant l'action à effectuer à partir des perceptions.

paramètres d'action correspondants sont le vecteur associé au neurone j dans l'A-DSOM (ligne 3). Nous ajoutons un bruit uniforme ϵ pour explorer l'espace des actions (lignes 4-6), et le vecteur perturbé est retourné comme action définitive $a \in A$ dans l'environnement.

Dans le processus d'apprentissage (voir algorithme 2), à la suite du processus de décision, les conteneurs éthiques (Q-Table et DSOMs) sont mis à jour selon la récompense r . Si r est meilleure que l'intérêt mémorisé, l'Action DSOM est mise à jour (l'action perturbée a était intéressante ; lignes 4-8). Ensuite, toutes les Q-Valeurs sont mises à jour (lignes 9-11), en intégrant le voisinage ψ des Perceptions et Actions DSOMs (lignes 1-2) à l'équation traditionnelle de Bellman. Finalement, la P-DSOM est mise à jour selon les perceptions reçues p (lignes 12-14). Ce cycle de décision-apprentissage-décision peut être vu dans la figure 2.

4 Cas d'application

En association avec notre partenaire industriel, nous appliquons notre modèle au cadre de la distribution d'énergie parmi les utilisateurs d'une micro-grille. Dans ces grilles, la production électrique est décentralisée, au lieu de s'appuyer sur le réseau national. Les *prosumers* (i.e. utilis-

Algorithm 2 Processus d'apprentissage

Variabes partagées : U la P-DSOM, W la A-DSOM, et Q la Q-Table

Fonctions : $c_U(k), c_W(k)$ position du neurone k dans les P-DSOM et A-DSOM

Hyper-paramètres : $\alpha_Q, \alpha_I, \alpha_A$: taux d'apprentissage, η_I, η_A : élasticité, γ : facteur d'actualisation

Entrées : p' nouvelles perceptions, r récompense, p perceptions précédentes, a action retournée par le processus de décision

```

1:  $\forall m \in U \quad \psi_I(m) := e^{-\frac{1}{\eta_I^2} \frac{\|c_U(m) - c_U(i)\|}{\|p - U_i\|}}$ 
2:  $\forall n \in W \quad \psi_A(n) := e^{-\frac{1}{\eta_A^2} \frac{\|c_W(n) - c_W(j)\|}{\|a - W_j\|}}$ 
3: Déterminer  $i := \arg \min \|p - U_i\|$ .
4: Déterminer  $i' := \arg \min \|p' - U_{i'}\|$ .
5: if  $(r + \gamma \max_{j'} Q_{i', j'}) > Q_{i, j}$  then
6:   for all neurone  $k \in W$  do
7:      $W_k := W_k + \alpha_A \|a - W_k\| \psi_A(k)(a - W_k)$ 
8:   end for
9: end if
10: for all état  $m \in U$ , action  $n \in W$  do
11:    $Q_{m, n} := Q_{m, n} + \alpha_Q \psi_I(m) \psi_A(n) [r + \gamma \max_{j'} (Q_{i', j'}) - Q_{m, n}]$ 
12: end for
13: for all neurone  $k \in U$  do
14:    $U_k := U_k + \alpha_I \|p - U_k\| \psi_I(k)(p - U_k)$ .
15: end for

```

teurs), en plus de consommer, produisent de petites quantités d'énergie (e.g. avec des panneaux photo-voltaïques) dans leur batterie personnelle. Considérant que la production et la demande peuvent fluctuer sur de courtes périodes, ces *prosumers* s'organisent en grilles pour échanger de l'énergie. Ces échanges locaux supposent une forme de coopération pour éviter les situations inégales. De manière similaire, quand la grille est trop sollicitée, les utilisateurs doivent réduire leur consommation (au moins temporairement), et de ce fait réduire leur confort, pour éviter les situations de coupure d'électricité.

Dans ces situations, l'intérêt de chaque *prosumer* peut générer des conflits avec toute la société : bien qu'il veuille maximiser son confort en consommant, il peut engendrer une dégradation des conditions de la grille (e.g. coupures d'électricité) ou un sacrifice de l'intérêt de plusieurs agents pour le bien des autres. Ainsi, ils doivent apprendre individuellement à prendre en compte l'intérêt de la société et adapter leur consommation de manière éthique. Ce cas d'application offre un cadre intéressant avec des considérations éthiques mobilisant des tensions entre des valeurs (e.g. l'équité, le respect des autres, l'écologie).

Suivant le modèle éthique que nous avons décrit en section 3.1, nous avons implémenté un simulateur multi-agent, avec un pas de temps discret, dans lequel l'environnement représente une micro-grille produisant de l'énergie. Chaque agent est un *prosumer* qui implémente le processus de décision effectué par un bâtiment pour consommer

et échanger de l'énergie. Plusieurs profils de bâtiment sont considérés : École primaire, Bureaux, Habitations ; chacun ayant un besoin spécifique chaque heure (voir figure 4) et consommant de l'énergie pour satisfaire ce besoin. Leur confort est calculé à partir du besoin et de leur consommation, en utilisant une fonction différente pour chaque profil, ce qui permet par exemple que les écoles aient une priorité supérieure aux habitations (car leur confort sera inférieur pour le même ratio). Ils possèdent également une batterie personnelle.

4.1 Actions

Les paramètres d'actions sont représentés par un vecteur $a = [q, p, s, u, r, z] \in R^6$, tel que q est la quantité d'énergie consommée depuis la micro-grille, p est la quantité d'énergie consommée depuis la batterie personnelle de l'agent, s est la quantité d'énergie que l'agent donne depuis sa batterie à la grille, u est la quantité d'énergie que l'agent stocke depuis la grille dans sa batterie, r est la quantité d'énergie que l'agent achète depuis le réseau national pour le stocker dans sa batterie, et z est la quantité d'énergie que l'agent vend au réseau national depuis sa batterie personnelle.

4.2 Perceptions

Les agents perçoivent l'état de l'environnement à travers plusieurs métriques. Bien que certaines puissent être considérées en référence avec des considérations éthiques (e.g. équité), les perceptions n'orientent pas l'agent vers un comportement éthique et ne font pas nécessairement partie de la récompense, i.e. l'injection éthique influant sur l'apprentissage. Afin de décrire ces métriques, nous introduisons quelques notations : *Conforts* est l'ensemble des confort des agents ; $Pris_i = q_i + u_i$ est la quantité d'énergie prise par l'agent i depuis la micro-grille ; $Donne_i = s_i$ est la quantité donnée par l'agent i à la micro-grille ; $Transactions_i = r_i + z_i$ est la quantité d'énergie échangée par l'agent i avec le réseau national.

- *Heure* : L'heure actuelle de l'environnement. Calculée comme $(Temps \bmod 24)/24$, ainsi chaque pas de temps représente une heure.
- *Énergie disponible* : La quantité d'énergie initialement disponible dans la micro-grille, notée E .
- *Équité* : Mesure statistique de dispersion de l'ensemble *Conforts*, en utilisant l'index de Hoover. Calculée comme $1 - Hoover(Conforts)$.
- *Énergie perdue* : Énergie disponible qui n'a pas été utilisée (consommée, stockée ou vendue) par les agents. Calculée comme $\min\left(\frac{E + \sum_i Donne_i - \sum_i Pris_i}{\sum_i Pris_i}, 0\right)$.
- *Autonomie* : Absence de transactions avec la grille nationale. Calculée comme $1 - \frac{\sum_i Transactions_i}{\sum_i (q_i + p_i + s_i + u_i + Transactions_i)}$.
- *Bien-être* : La médiane des confort des agents.
- *Exclusion* : La proportion d'agents dont le confort est inférieur à 50% de la médiane. Calculée comme $\frac{|Conforts_i < 0.5 \times Bien\text{-}Etre|}{|Agents|}$.
- *Sur-Consommation* : La proportion d'éner-

gie utilisée par les agents mais qui n'était pas disponible initialement. Calculée comme $\min\left(\frac{\sum_i Pris_i - \sum_i Donne_i - E}{\sum_i Pris_i}, 0\right)$.

Ces métriques sont les mêmes pour chaque agent puisqu'elles décrivent l'environnement; toutefois, les agents perçoivent également des données additionnelles sur eux-mêmes :

- *Batterie personnelle* : Le ratio entre la quantité d'énergie disponible dans la batterie personnelle de l'agent et la capacité de cette batterie.
- *Confort* : Le confort de l'agent au pas de temps précédent.
- *Bénéfice* : La quantité d'argent que l'agent a gagné (ou perdu) à travers ses transactions avec le réseau national (en vendant ou achetant de l'énergie).

Ces 3 données sont différentes pour chaque agent; la combinaison des métriques de l'environnement et des données de l'agent produit un vecteur de perceptions $p \in R^{11}$.

4.3 Dynamiques intrinsèques et garde-fou

À chaque pas de temps, une certaine quantité d'énergie est disponible pour les agents, à la fois dans la grille (production globale) et dans leur batterie (production par agent). Les agents décident d'échanger de l'énergie entre la grille, leur batterie, et le réseau national, ce qui produit de nouveaux états de l'environnement. Si les agents n'agissent pas explicitement sur une quantité d'énergie (elle n'est pas consommée, stockée ou vendue), elle est considérée comme perdue. À l'inverse, si un agent consomme plus d'énergie qu'il n'y en a de disponible dans la grille (après les dons des autres agents), sans l'acheter explicitement, cette différence est appelée la sur-consommation. Nous implémentons un simple mécanisme de garde-fou pour prévenir les coupures d'électricité en simulant une transaction dans ce cas, entre la grille locale et le réseau national.

4.4 Injection éthique - Récompenses

Afin d'explorer la capacité de notre modèle à apprendre des comportements éthiques, nous avons conçu plusieurs fonctions de récompense :

- *Équité* : Cette récompense utilise un indicateur de dispersion statistique, nommé index de Hoover, sur les confort des agents. Elle vise l'équité dans la distribution des confort, c'est-à-dire qu'aucun agent ne se sacrifie tandis que les autres vivent dans le luxe. $R_{equite,a} = (1 - Hoover(Conforts)) - (1 - Hoover(Conforts \setminus \{Conforts_a\}))$

- *Sur-Consommation (SC)* : Cette récompense vise à réduire la sur-consommation en impactant négativement l'énergie consommée par l'agent mais qui n'était pas disponible dans la grille.

$$R_{SC,a} = 1 - \frac{\max(\sum_i Pris_i - \sum_i Donne_i - E, 0)}{\sum_i Pris_i} - \frac{\max(\sum_i Pris_i - \sum_i Donne_i - E, 0) - Pris_a}{\sum_i Pris_i}$$

- *Multi-Objectif-Somme (MOS)* : Cette récompense complexe combine deux sous-récompenses : la sur-consommation et le confort. Cela vise à

maximiser le confort tout en minimisant la sur-consommation et évite les solutions triviales que les agents peuvent trouver pour optimiser ces sous-récompenses, e.g. consommer le maximum pour maximiser le confort, sans prendre en compte la sur-consommation, ou à l'inverse ne rien consommer pour minimiser la sur-consommation. Ces solutions échouent à capturer l'attention éthique qui est de satisfaire son confort tout en faisant attention à l'intérêt de la société. $R_{MOS,a} = \frac{1}{2} \times R_{SC,a} + \frac{1}{2} \times Conforts_a$

- *Multi-Objectif-Produit (MOP)* : De manière similaire à Multi-Objectif-Somme, cette récompense combine sur-consommation et confort. $R_{MOP,a} = R_{SC,a} \times Conforts_a$

- *Adaptabilité1 (Ada1)* : Cette récompense change au cours du temps, afin de vérifier l'adaptabilité de notre processus d'apprentissage. Nous apprenons d'abord à minimiser la sur-consommation, puis après un nombre de pas de temps fixe (2000), nous ajoutons l'équité. $R_{Ada1,a} =$

$$\begin{cases} R_{SC,a}, & \text{si } etape \leq 2000 \\ \frac{R_{SC,a} + R_{equite,a}}{2}, & \text{sinon} \end{cases}$$

- *Adaptabilité2 (Ada2)* : Suivant le même principe, cette récompense change au cours du temps, considérant 3 phases. $R_{Ada2,a} =$

$$\begin{cases} R_{SC,a}, & \text{si } etape \leq 2000 \\ \frac{R_{SC,a} + R_{equite,a}}{2}, & \text{si } etape \leq 6000 \\ \frac{R_{SC,a} + R_{equite,a} + Confort_a}{3}, & \text{sinon} \end{cases}$$

5 Expérimentations et résultats

5.1 Expérimentations

Nous comparons les performances du modèle originel de Smith avec des SOMs (ci-après "Q-SOM") à notre extension avec des *Dynamic* SOMs (ci-après "Q-DSOM"). Nous considérons également une implémentation par Apprentissage Profond d'une architecture *Actor-Critic*, nommée DDPG [12], ainsi qu'une stratégie aléatoire comme base de référence.

Dans chacune des expérimentations, nous avons considéré un petit nombre de bâtiments : 20 habitations, 5 bureaux et 1 école. Le besoin en énergie de chaque profil a été déterminé en utilisant un jeu de données public de consommation³. Trois types de bâtiments ont été sélectionnés dans la même ville (Anchorage) afin de minimiser le risque de biais entre les profils (e.g. pas le même besoin en chaleur) : *Residential*, *Primary School*, *Small Office*. Le profil de consommation horaire a été moyenné sur toute l'année pour chacun (voir figure 4), ce qui donne 24 valeurs de besoin (1 par heure) pour chaque profil d'agent.

Nous avons choisi de rendre disponible dans la micro-grille, à chaque pas de temps, une quantité d'énergie calculée comme une valeur aléatoire entre 80% et 110% de

3. <https://openei.org/datasets/dataset/commercial-and-residential-hourly-load-profiles-for-all-tmy3-locations-in-the-united-states>

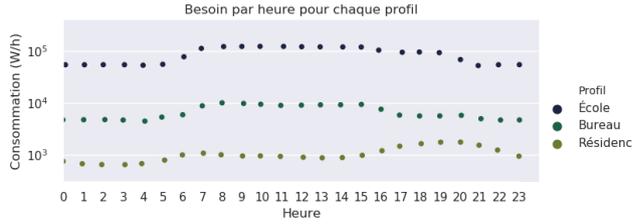


FIGURE 4 – Besoin en énergie pour chaque heure et chaque profil.

Modèle	Paramètre	Valeur
DDPG	Taille de lot	128
	Taux d'apprentissage	0.001
	γ	0.99
	τ	0.001
Q-DSOM et Q-SOM	Dimensions Perceptions SOM	7x7
	Perceptions SOM α	0.8
	Dimensions Action SOM	5x5
	Action SOM α	0.5
	Q-Learning α	0.5
	Q-Learning γ	0.9
	Boltzmann τ	0.5
	Bruit des actions ϵ	0.08
Q-DSOM	Perceptions DSOM η	2.0
	Action DSOM η	2.0
Q-SOM	Perceptions SOM σ	1.0
	Action SOM σ	1.0

TABLE 1 – Valeurs des différents hyper-paramètres pour chacun des 3 modèles. Ces valeurs contrôlent par exemple le dilemme exploitation-exploration, le bruit ajouté aux paramètres d'action, etc. La politique aléatoire ne dispose pas d'hyper-paramètre.

l'énergie totale dont ont besoin les bâtiments. Cet intervalle a été choisi afin de faire apparaître plus de situations de conflits (où il n'y a pas assez d'énergie pour tous) que de situations où les agents peuvent consommer plus que nécessaire.

Les trois modèles (DDPG, Q-SOM, Q-DSOM) utilisent différents hyper-paramètres dont nous relevons les valeurs dans la table 1. Toutes les fonctions de récompense ont été évaluées avec les mêmes valeurs d'hyper-paramètres.

Nous pouvons notamment noter que la carte des perceptions (SOM ou DSOM) a une taille 7x7, ce qui correspond à 49 neurones (états discrets). Ce petit nombre d'états (à comparer au vecteur de perceptions à 11 dimensions, composé de valeurs continues) permet à la table de Q-Valeurs d'apprendre une politique d'action optimale pour chaque état.

5.2 Résultats

Chacun des scénarios a été exécuté 10 fois afin de produire des résultats robustes. La figure 5 montre ces résultats

comme diagramme de quartiles.

Comme la figure le montre, Q-SOM et Q-DSOM ont la plupart du temps de meilleures performances que DDPG et aléatoire. La seule exception où DDPG obtient un score presque parfait est la récompense de Sur-Consommation : nous supposons qu'il est plus facile pour l'algorithme DDPG d'atteindre une solution triviale (e.g. consommer 0) grâce à son mécanisme de rétro-propagation de l'erreur, tandis que Q-SOM et Q-DSOM appliquent des bruits aléatoires pour explorer l'espace des actions. Cette hypothèse est soutenue par le fait que DDPG obtient un score inférieur à Q-SOM et Q-DSOM sur les récompenses Multi-Objectif (Somme et Produit). La politique aléatoire obtient toujours un score inférieur à Q-SOM et Q-DSOM, mais meilleur que DDPG sur les récompenses d'équité, Multi-Objectif Somme et Multi-Objectif Produit.

Les modèles Q-SOM et Q-DSOM obtiennent des résultats similaires; Q-DSOM a un score médian supérieur à celui de Q-SOM sur les fonctions de récompense "Multi-Objectif Somme" et "Adaptabilité 2" (MOS et Ada2), mais Q-SOM a un score médian supérieur sur "Multi-Objectif Produit" et "Adaptabilité 1" (MOP et Ada1). Ces résultats suggèrent un fort impact du choix de la fonction de récompense sur les performances d'apprentissage.

La figure 6 montre l'évolution de la récompense globale (i.e. pour l'ensemble des agents), en utilisant la fonction de récompense *Adaptabilité2* et le modèle Q-DSOM. Pour rappel, cette fonction contient 3 phases, considérant dans chacune des phases un sous-ensemble de valeurs à apprendre (d'abord la sur-consommation, puis sur-consommation et équité, et enfin sur-consommation, équité et confort). Autrement dit, la fonction de récompense qui est utilisée pour guider les agents change selon le temps; les agents doivent s'adapter à ce changement et apprendre les nouvelles valeurs. Ces phases sont visibles sur la figure (brusques changements dans la récompense reçue). La croissance des récompenses sur cette figure montre l'adaptabilité effective des agents face à de tels changements.

6 Discussion

Premièrement, nous pouvons noter que, alors que nous choisissons les DSOMs, d'autres travaux proposent de remplacer les SOMs par des *Growing Self-Organization Maps* (GSOMs) à la place [13, 5], ce qui peut améliorer la convergence dans certains environnements. En effet, les DSOMs ne garantissent pas la convergence; considérant 4 états en forme de carré avec seulement 3 neurones, le dernier "rebondira" indéfiniment entre 2 coins. Les GSOMs créent plus de neurones, tandis que les SOMs utilisent le mécanisme de décroissance pour forcer la convergence vers une solution approchée. La création de plus de neurones dans les GSOMs implique une plus grande difficulté à apprendre l'intérêt de chaque état (chaque neurone représentant un état discret, un plus grand nombre d'états requiert au minimum plus de temps), tandis que le mécanisme de décroissance dans les SOMs requiert un ajustement des hyper-paramètres (la décroissance ne doit pas

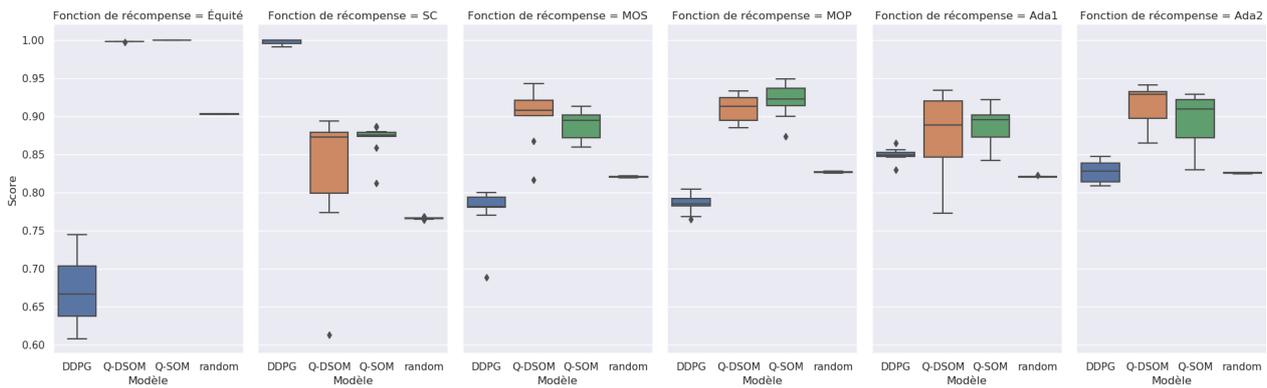


FIGURE 5 – Résultats comparatifs entre les modèles pour chaque fonction de récompense, sur plusieurs simulations. Le score d’une simulation correspond à la moyenne de la récompense globale (calculée pour l’ensemble des agents) sur l’ensemble des pas de temps (10 000). Chacun des 6 sous-graphiques correspond à une fonction de récompense (avec SC pour la Sur-Consommation, MOS pour Multi-Objectif Somme, MOP pour Multi-Objectif Produit, Ada1 et Ada2 pour Adaptabilité 1 et 2).

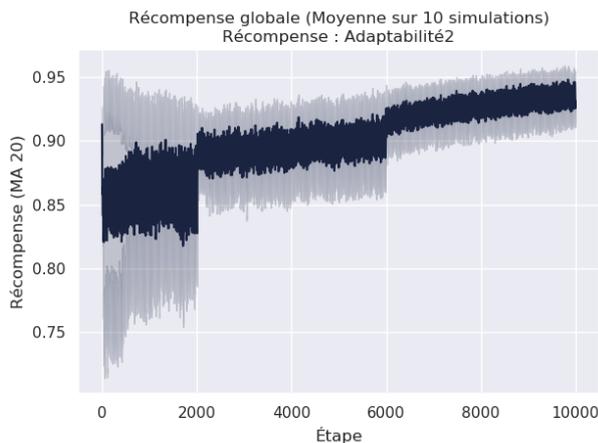


FIGURE 6 – Récompense globale à chaque pas de temps. Les récompenses affichées sont moyennées sur 20 pas de temps (*Moving Average - MA*) et sur les 10 simulations exécutées : la courbe en gris foncé indique la moyenne sur ces 10 simulations, les courbes en gris clair indiquent l’écart-type.

être trop rapide sans quoi l’apprentissage ne sera pas efficace, ni trop lent sans quoi la convergence ne sera pas efficace). Nous pensons que les DSOMs sont de ce fait un choix intéressant, en particulier parce qu’elles permettent d’apprendre la distribution des données au lieu de la densité (permettant ainsi d’apprendre même les états rares).

En sus des vertus relevant de l’éthique par conception, notre choix d’utiliser de multiples agents nous permet d’intégrer plus facilement des vertus relevant de l’éthique dans la conception (*ethics in design*). En effet, il est ainsi plus facile de satisfaire des valeurs éthiques telles que le respect de la vie privée (chaque agent pouvant manipuler directement des données sensibles de l’utilisateur sans les partager avec une instance centralisée), ou encore le respect de l’environnement (avec l’informatique distribuée et le *Green Computing*). Ces deux propriétés sont liées aux implications éthiques de l’utilisation du système plutôt qu’au comportement éthique du système.

Notre approche démontre la capacité d’apprendre l’espace des perceptions et de déterminer l’action optimale pour un comportement éthique. Toutefois, quelques limitations peuvent être soulignées :

- La représentation des différentes actions à travers une correspondance entre les Q-Valeurs et les neurones de l’Action DSOM peut mener à des questionnements (éthiques). Si un certain neurone est déplacé trop loin de sa précédente position, l’action correspondante aurait une signification complètement différente, mais serait toujours en correspondance avec la même Q-Valeur (i.e. le même intérêt), et aurait toujours le même voisinage. En pratique, les mécanismes de mise à jour des DSOMs et de la *Q-Table* modifient également l’intérêt et le voisinage, mais nous n’excluons pas ceci des questions (éthiques) pouvant survenir.
- De plus, l’exploration d’actions multi-dimensionnelles en utilisant un bruit ne semble

pas optimale. En particulier, cela signifie que nous pouvons améliorer l'action proposée sur une certaine dimension, mais complètement dégrader la même action sur une seconde dimension (ce qui annulerait la première amélioration).

- Contrairement à la DSOM d'actions et la *Q-Table*, la DSOM de perceptions n'utilise pas la récompense pour se mettre à jour ; seules les exemples de données en entrée sont utilisés. Toutefois, on pourrait arguer qu'une représentation correcte est nécessaire pour déterminer l'action correcte ; ainsi, une récompense basse pourrait être le signe que la représentation n'est pas utile à l'agent et devrait être apprise différemment.

De plus, les résultats présentés dépendent des expérimentations réalisées ; nous pouvons citer quelques scénarios qui seraient intéressants à expérimenter pour vérifier la robustesse du modèle :

- Le profil de consommation étant moyenné par jour, il est possible que certaines variations (e.g. les saisons) soient effacées. Il serait intéressant de comparer avec une simulation où le profil ne serait pas moyenné, en particulier par rapport au problème bien connu de l'oubli catastrophique.
- L'énergie disponible est toujours entre 80 et 110% de l'énergie totale dont les agents ont besoin. On pourrait imaginer qu'une pénurie d'énergie survienne (e.g. absence soudaine de vent ou de soleil), sur une période plus ou moins longue. Cela permettrait d'analyser la capacité d'adaptation des agents face à un changement dans la dynamique de l'environnement elle-même.
- Nous avons considéré une petite grille d'agents, il serait possible de faire des expérimentations avec un ensemble plus grand afin de déterminer si le nombre d'agents a un impact sur les résultats.

Pour conclure, l'approche proposée ici suggère l'exploration de nouvelles questions (notamment éthiques) :

- Il serait intéressant de se demander si la fonction de récompense, associée aux structures et algorithmes que l'agent implémente, lui permet effectivement d'apprendre le comportement éthique désiré. En d'autres termes, l'agent est-il susceptible d'effectuer une forme de *reward hacking* (i.e. de converger vers des solutions triviales qui maximisent la fonction de récompense mais auxquelles les concepteurs ne s'attendaient pas, e.g. consommer 0 Watts afin que l'équité soit maximale et la sur-consommation nulle).
- Nous pouvons également nous demander si les choix reflétés dans les fonctions de récompense et les comportements éthiques obtenus sont effectivement éthiques et désirés. Cette question nécessiterait une discussion plus large avec des philosophes, des décideurs politiques, et des utilisateurs.
- Nous avons choisi de placer un garde-fou dans l'environnement, ce qui implique que le comportement des agents dépend de cet élément qui leur est ex-

terne. Il pourrait être intéressant d'ajouter un garde-fou sous forme d'agent ou bien de l'intégrer directement aux agents.

- La fonction de récompense est actuellement la même pour tous les agents, or les utilisateurs (humains) ont des valeurs différentes (en d'autres termes, n'ont pas les mêmes préférences). Il pourrait être intéressant de permettre de paramétrer chaque agent, d'une part pour l'acceptabilité d'une telle solution par les utilisateurs finaux, et d'autre part pour étudier la capacité d'apprentissage et d'adaptation des agents face à d'autres agents ayant des "valeurs" différentes (à travers la fonction de récompense), et de ce fait un comportement différent.

Remerciements

Ces travaux sont financés par la région Auvergne-Rhône-Alpes (Pack Ambition Recherche) dans le cadre du projet Ethics.AI. Les auteurs remercient leurs partenaires académiques et industriels au sein du projet.

Références

- [1] Nirav Ajmeri, Hui Guo, Pradeep K Murukannaiah, and Munindar P Singh. Designing ethical personal agents. *IEEE Internet Computing*, 22(2) :16–22, 2018.
- [2] Colin Allen, Iva Smit, and Wendell Wallach. Artificial morality : Top-down, bottom-up, and hybrid approaches. *Ethics and information technology*, 7(3) :149–155, 2005.
- [3] Michael Anderson and Susan Leigh Anderson. Guest editors' introduction : Machine ethics. *IEEE Intelligent Systems*, 21(4) :10–11, 2006.
- [4] Michael Anderson and Susan Leigh Anderson. Toward ensuring ethical behavior from autonomous systems : a case-supported principle-based paradigm. In *2014 AAAI Fall Symposium Series*, 2014.
- [5] Tomoyuki Arai, Yuichiro Toda, Iwasa Mutsumi, Shuai Shao, Ryuta Tonomura, and Naoyuki Kubota. Reinforcement learning based on state space model using growing neural gas for a mobile robot. In *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*, pages 1410–1413. IEEE, 2018.
- [6] Ronald Arkin. *Governing lethal behavior in autonomous robots*. Chapman and Hall/CRC, 2009.
- [7] Nicolas Cointe, Grégory Bonnet, and Olivier Boisier. Ethics-based cooperation in multi-agent systems. In *14th Social Simulation Conference (SSC18), August 2018.*, 2018.
- [8] Vincent Conitzer, Walter Sinnott-Armstrong, Jana Schaich Borg, Yuan Deng, and Max Kramer. Moral decision making frameworks for artificial intelligence. In *The Workshops of the The Thirty-First*

AAAI Conference on Artificial Intelligence, Saturday, February 4-9, 2017, San Francisco, California, USA, 2017.

- [9] Virginia Dignum. Ethics in artificial intelligence : introduction to the special issue, 2018.
- [10] Virginia Dignum. *Responsible Artificial Intelligence : How to Develop and Use AI in a Responsible Way*. Springer International Publishing, 2019.
- [11] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9) :1464–1480, Sep. 1990.
- [12] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv :1509.02971*, 2015.
- [13] Hesam Montazeri, Sajjad Moradi, and Reza Safabakhsh. Continuous state/action reinforcement learning : A growing self-organizing map approach. *Neurocomputing*, 74(7) :1069–1082, 2011.
- [14] James Moor. Four kinds of ethical robots. *Philosophy Now*, 72 :12–14, 2009.
- [15] Nicolas Rougier and Yann Boniface. Dynamic self-organising map. *Neurocomputing*, 74(11) :1840–1847, 2011.
- [16] Andrew James Smith. Applications of the self-organising map to reinforcement learning. *Neural networks*, 15(8-9) :1107–1124, 2002.
- [17] Wendell Wallach, Colin Allen, and Iva Smit. Machine morality : bottom-up and top-down approaches for modelling human moral faculties. *Ai & Society*, 22(4) :565–582, 2008.
- [18] Wendell Wallach, Stan Franklin, and Colin Allen. A conceptual and computational model of moral decision making in human and artificial agents. *Topics in cognitive science*, 2(3) :454–485, 2010.
- [19] Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.
- [20] Yueh-Hua Wu and Shou-De Lin. A low-cost ethics shaping approach for designing reinforcement learning agents. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [21] Logan Yliniemi and Kagan Tumer. Multi-objective multiagent credit assignment through difference rewards in reinforcement learning. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 407–418. Springer, 2014.
- [22] Han Yu, Zhiqi Shen, Chunyan Miao, Cyril Leung, Victor R. Lesser, and Qiang Yang. Building ethics into artificial intelligence. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 5527–5533, 2018.

Un processus pour la sélection de stratégie de gestion de l'information dans les systèmes de confiance

N. Cointe, A. Ghorbani, C. Chorus

Department of Engineering Systems and Services,
Faculty of Technology, Policy and Management,
Delft University of Technology, Pays-Bas

nicolas.cointe@tudelft.nl

a.ghorbani@tudelft.nl

c.g.chorus@tudelft.nl

Résumé

Dans certains systèmes multi-agents, en particulier des systèmes ouverts et distribués, la coopération peut nécessiter au préalable d'observer et évaluer le comportement des autres. Pour ce faire, plusieurs techniques de reconnaissance d'objectifs et de buts ont déjà été présentées dans la littérature. Nous introduisons ici le concept de gestion de l'information, permettant à l'agent autonome artificiel de mesurer et connaître la quantité d'information donnée à un observateur pour en tenir compte dans sa prise de décisions, et montrons comment cela peut impacter ses interactions sociales. Un ensemble d'expérimentations employant des agents BDI illustre enfin la sensibilité de ces interactions sociales aux stratégies de gestion de l'information utilisées.

Mots-clés

Confiance, auto-organisation, confidentialité.

Abstract

In some multiagent systems, more particularly open and distributed systems, coalitions formation requires to observe and evaluate the others' behavior before initiating any cooperation. In that way, the literature already proposes some goals and plans recognition techniques. We introduce here the concept of Information regulation, allowing artificial autonomous agents to measure – and be aware of – the quantity of information given to an observer and consider it to make decisions. A set of experiments illustrates how the various information regulation strategies may impact social interactions of BDI agents.

Keywords

Trust, self-organization, privacy.

1 Introduction

Dans des jeux coopératifs, trouver des partenaires susceptibles d'être intéressés par une coopération et la construction d'une coalition ou d'une organisation est une tâche complexe et sensible, lorsqu'une telle interaction peut influencer les bénéfices et pertes des agents ou (non exclusif) les exposer à des sanctions sociales telles qu'une exclusion.

De tels cas de figures peuvent se présenter par exemple dans des situations présentant une dimension morale [6] et où des actions, bien qu'exécutables et permises, peuvent rendre leurs auteurs désirables ou non au sein d'un collectif. Notons que nous traitons ici de cas où les agents ne disposent d'aucune preuve sur les intentions des autres agents mais emploient des techniques proposées dans la littérature afin d'évaluer avec une certitude suffisante, les objectifs d'un agent.

Sélectionner des partenaires semblant partager des objectifs similaires peut se faire, de manière classique, à l'aide de modèles de confiance [21] afin de laisser le système s'auto-organiser. L'avantage d'un tel modèle est qu'il est construit de manière décentralisée, par les agents, en se fondant sur leur expérience individuelle – tirant profit d'observations, connaissances et raisonnement propres à l'agent – et éventuellement sur des expériences transmises par un système de réputation [26].

Afin de pouvoir raisonner sur les objectifs des autres agents, la littérature propose déjà un ensemble de techniques diverses, que nous regroupons dans cet article au sein du concept plus général de *reconnaissance d'objectifs*, permettant d'inférer, à partir d'observations, le but le plus vraisemblable compte tenu du comportement d'un agent observé. Les méthodes de reconnaissance d'objectif ont démontré ces deux dernières décennies leur intérêt dans divers domaines applicatifs nécessitant une coopération entre agents autonomes et humains, tels que l'apprentissage de routines comportementales pour la surveillance de personnes âgées à domicile [14], ou bien pour la coopération d'agents autonomes entre eux, par exemple dans le cas d'un groupe de robots devant coordonner leurs déplacements en minimisant les communications [4]. La reconnaissance d'objectifs est également utile dans un cadre compétitif, afin d'anticiper et éventuellement contrer les buts d'un adversaire. Par exemple, leur usage a montré sa pertinence pour la détection de comportements humains dangereux dans une foule [22], ou la déduction de la stratégie employée par un adversaire dans des jeux vidéos [15] ou des compétitions de football robotiques [16].

Cependant, tenir compte de l'observation de leur comportement, et de la présence d'un système de confiance

et de réputation peut permettre à des agents –ou groupes d’agents– d’influencer l’opinion des autres agents du système afin de les tromper sur leurs propres intentions [2] ou seulement maintenir un certain niveau d’incertitude chez l’observateur [7]. Dans le cadre de cet article, nous exposons le cas d’agents cherchant à maximiser ou à minimiser l’information donnée au travers de leur comportement à des observateurs afin de clarifier ou dissimuler leurs objectifs dans le but d’influencer leurs chances de rejoindre un collectif d’agents, ou de minimiser les risques de sanctions.

Le concept de *gestion d’information* propose d’évaluer la quantité d’information pouvant être révélée par un comportement (exécuté ou anticipé) à un potentiel observateur. Cette mesure de quantité d’information transmise permet alors d’employer une stratégie de minimisation de l’information, nous parlerons alors de *dissimulation*, ou au contraire de sa maximisation, que nous désignons par le terme de *transparence*.

Cet article propose d’explorer divers aspects des stratégies de gestion d’information au travers d’un modèle dans lequel chacun cherche à identifier les buts des autres et gagner la confiance de ceux qu’il estime poursuivre un objectif similaire ou compatible afin de construire un collectif.

En Section 2, nous présentons des travaux de la littérature présentant diverses contributions en matière de reconnaissance d’objectif, de gestion d’information et de mécanismes de construction de confiance. Puis en Section 3 nous présentons notre proposition afin de décrire le processus selon lequel les agents adoptent une stratégie de régulation de l’information en fonction du contexte. La Section 4 illustre des résultats expérimentaux obtenus à la suite de l’implémentation de ce modèle, et son usage dans un scénario montrant des agents dotés de points de vue et objectifs divers. Enfin, en Section 5, nous présentons nos conclusions sur l’intérêt de cette contribution et présentons nos intentions pour la poursuite de nos travaux.

2 État de l’art

Dans cette section, nous exposons des travaux illustrant plus en détail les concepts fondamentaux sur lesquels repose notre proposition.

En premier lieu, nous définissons le comportement b_{ag_i} d’un agent $ag_i \in Ag$ comme un ensemble ordonné d’actions :

$$b_{ag_i} = \{a_{ag_i, t_0}, \dots, a'_{ag_i, t'}\}$$

avec $a_{ag_i, t_0}, \dots, a'_{ag_i, t'}$ une séquence d’actions exécutée par l’agent ag_i entre les instants t_0 et t' . Nous désignons dans le cadre de cet article sous l’intitulé de *comportement observé*, un ensemble d’actions observées ayant déjà été exécutées par l’agent ag_i tel qu’à l’instant courant t , $t' \leq t$. À l’inverse, un *comportement hypothétique* inclut au moins une action hypothétique, c’est-à-dire dont l’exécution n’a pas été observée. Cette absence d’observation peu s’expliquer par la nature anticipée d’un comportement, dans le cas où $t' > t$, soit qu’un agent cherche à évaluer un plan qu’il a l’intention d’exécuter, soit qu’il en

visage un comportement futur possible d’un autre agent. Cela peut aussi permettre, dans des systèmes partiellement observables, d’envisager des actions qui n’ont pas été observées bien qu’elles puissent avoir été effectuées.

La sous-section 2.1 présente premièrement plusieurs aspects de la modélisation de la confiance dans les sociétés artificielles. En sous-section 2.2, nous présentons un ensemble non-exhaustif de techniques de reconnaissance d’objectifs, afin d’illustrer leur diversité et proposer en conséquence une définition générique de *fonction de reconnaissance d’objectif* (GRF) englobant l’ensemble de ces propositions. Enfin, la sous-section 2.3 montre comment une fonction de reconnaissance d’objectif peut être utilisée dans un processus décisionnel afin de mettre en œuvre une stratégie de reconnaissance d’objectif.

2.1 Modèles de confiance

Dans un système multi-agent ouvert, décentralisé et hétérogène, les agents peuvent chercher à atteindre divers objectifs, avoir des perceptions et actions diverses, voire des architectures entièrement différentes. Afin de permettre aux agents d’évaluer dynamiquement les autres agents selon leurs propres critères et architecture, afin de sélectionner de potentiels collaborateurs, une approche classique consiste à implémenter un processus de confiance [21]. Dans un tel modèle chaque agent évalue les autres en fonction de l’observation de leur comportement et des évaluations successives. L’état mental, construit par ces évaluations est appelé *image* et prend généralement une valeur, discrète ou continue qui, lorsqu’elle est considérée suffisante, peut entraîner l’établissement d’une confiance [26]. L’une des définitions les plus consensuelles [18, 21] de la confiance dans le domaine des systèmes multi-agents, semble être celle proposée par Gambetta [9] la définissant comme : “un niveau particulier de la probabilité subjective qu’un agent –ou groupe d’agent– puisse effectuer une action”. Ici, plus formellement, nous considérerons que la confiance est une croyance d’un agent en l’intention d’un autre agent d’accomplir une action ou un but spécifique.

La confiance est aussi considérée dans la littérature comme un élément de *soft security* [24] qui, au niveau collectif, prévient la propagation de fausses croyances par des agents malicieux en les excluant d’interactions sociales.

Dans le cas particulier de la reconnaissance de buts, la probabilité d’un objectif donné, évaluée au regard d’une fonction de reconnaissance de but et d’un ensemble de connaissances, constitue une description d’une attente de l’agent observateur envers le comportement de l’agent observé.

De plus, par souci de concision nous ne traitons pas dans cet article de l’usage de la confiance dans le mécanisme de construction d’une coalition, mais proposons de considérer l’acquisition de la confiance des autres comme un préalable nécessaire à l’intégration d’un agent dans un collectif d’agents (pouvant prendre par exemple la forme d’une organisation) ou comme un événement déclencheur de la construction d’une telle structure. Nous nous plaçons dans le cadre de systèmes pouvant être assimilés à des jeux coopératifs (c’est-à-dire dans lesquels les agents ont

des gains ou performances accrues en cas de coopération). Ainsi, nous décrivons ici des agents qui vont poursuivre leur propre but, tout en prenant en compte l'impact social de leur comportement afin de pouvoir augmenter si possible leurs gains en gagnant la confiance d'autres agents.

En plus de cet usage de l'évaluation mutuelle entre agents en vue de la coopération, nous considérons également dans le cadre de cet article l'usage opposé, c'est-à-dire consistant à employer ces mécanismes de construction de la confiance pour identifier les agents susceptibles de poursuivre des objectifs indésirables pour l'observateur. Une fois acquise la confiance dans la poursuite d'objectifs indésirables (du point de vue de l'observateur), il est possible d'appliquer des sanctions, soit unilatéralement en refusant de collaborer avec l'agent observé, soit la forme de peines décidées par un groupe d'agents pouvant réduire les gains de l'agent jugé ou l'exclure d'un collectif [17, 28]. Les sanctions associées à ces infractions peuvent être représentées explicitement sous forme d'institutions (voir par exemple le cadre conceptuel ADICO [13]) et mettre ainsi à disposition des agents une représentation compréhensible et accessible pouvant être prise en compte dans leur décision.

2.2 Reconnaissance d'objectifs

Comme mentionné en section 2.1, afin de permettre aux agents de se construire une représentation mentale de l'intention d'un autre agent, nous allons employer une fonction de reconnaissance d'objectif, que nous définirons formellement par une fonction abstraite. Être capable d'interpréter le comportement d'un autre agent fait partie intégrante de la capacité à comprendre le contexte dans lequel l'agent observateur se trouve, et plus précisément le contexte social. Diverses approches ont fait l'objet de propositions dans la littérature en Intelligence Artificielle et nous nous attachons ici, sans prétendre être exhaustifs, à montrer leur diversité et mettre en lumière les points communs.

Nous présentons en premier lieu une première catégorie, que nous nommerons *reconnaissance de plans* (PR) [20]. L'usage de plans étant classique pour décrire la manière dont les agents peuvent atteindre leurs buts au travers d'une séquence d'actions et de buts intermédiaires, cette approche regroupe les propositions d'algorithmes permettant d'évaluer les similitudes entre un comportement observé et les plans d'une bibliothèque de plans. Pour que cette approche soit efficace, la bibliothèque de plans doit être similaire sinon partagée entre l'agent observé et l'observateur. Cette approche a montré de bons résultats, même face à des agents humains ou plus généralement des cas d'observabilité partielle [11]. Au cours de l'observation, l'observateur évalue et révisé une ensemble d'*explications possibles*, dans lesquelles chaque action est associée à un plan. L'explication la plus probable, au regard du rasoir d'Ockham, est la plus simple, c'est-à-dire celle qui fait appel aux moins de plans différents possibles. L'évaluation de l'ensemble des explications souffre toutefois d'une explosion combinatoire, puisqu'il faut effectivement pour chaque action du comportement observé, envisager toutes les oc-

currences de cette action dans la bibliothèque de plans. De nombreux algorithmes proposés, tels qu'ELEXIR [10], PHATT [11], YAPPR [12], DOPLAR [15] ou encore SLIM [19] cherchent à résoudre ce problème en réduisant continuellement la taille de l'ensemble des explications possibles en procédant à l'exclusion des plus invraisemblables.

La reconnaissance de plans donne des résultats significatifs lorsque l'observateur dispose d'une bibliothèque de plans pertinente et appropriée, et si les plans la constituant sont assez détaillé et différents les uns des autres. Toutefois dans certaines applications les observateurs ne peuvent pas disposer de telles connaissances données à l'initialisation. C'est par exemple le cas d'agents employant des techniques d'apprentissage pour connaître l'effet de leurs effecteurs sur le monde perçu au travers de leur fonction de perception. C'est aussi le cas lorsque le concepteur d'un système, ou *a minima* de l'agent observateur, ne sait pas encore ce que peuvent être les comportements pertinents dans le système. Pour résoudre ce problème, certains auteurs proposent alors d'employer des techniques d'*apprentissage automatique de comportements* pour permettre à l'observateur de construire des modèles de comportement à partir d'observations. Ces modèles sont ensuite utilisés pour évaluer les comportements observés durant leur exécution. Par exemple, le domaine applicatif choisi pour présenter l'algorithme ILSA [14] montre un agent observant une personne âgée dans son domicile afin de proposer en cas de nécessité un contact avec du personnel soignant approprié. Cette approche montre tout d'abord son intérêt par sa capacité à extraire un ensemble de motifs émergeant progressivement des comportements humains observés. Sa flexibilité lui permet de composer avec les capteurs à sa disposition, les spécificités du domicile dans lequel le système est déployé et les habitudes spécifiques à la personne observée. D'autres proposent d'utiliser des techniques d'apprentissage non supervisé hors du temps d'exécution [16] pour effectuer de la recherche de motifs fréquents dans des comportements observés et révéler des descriptions abstraites de types de comportement qui seront ensuite utilisés en temps réel pour inférer les objectifs de l'agent observé durant l'exécution. Dans ce cas, le motif associé aux comportements observés une liste d'objectifs associés à une certaine probabilité extraite de la base d'apprentissage.

Les techniques présentées et regroupées dans ces deux approches –par *reconnaissance de plans* ou par *apprentissage de comportements*– proposent toutes d'informer l'observateur de l'objectif le plus probable associé à un comportement et à un ensemble de connaissances (donné à l'agent ou appris durant ou hors de l'exécution). La première catégorie est généralement préférée lorsque le concepteur dispose de connaissances sur les comportements attendus. La seconde approche, offre en revanche la possibilité de construire cette connaissance en apprenant à associer des séquences d'actions à des buts à partir d'un grand nombre d'observations. Afin de proposer une abstraction de cet ensemble de techniques, nous définissons un concept plus gé-

néral et abstrait de *fonction de reconnaissance d'objectif* ou GRF (pour *Goal Recognition Function*) telle que :

$$GRF : b_{agi} \times K \times G \rightarrow [0; 1]$$

où cette fonction prend en paramètres le comportement b_{agi} , l'ensemble de connaissances K (représenté comme une bibliothèque de plans, un ensemble de motifs ou toute autre forme de connaissances), et l'ensemble des buts G possibles associés à la probabilité $P(g_i)$ que l'agent observé ait l'intention d'atteindre le but g_i . Afin de normaliser le résultat obtenu, nous considérons par la suite la somme de toutes les probabilités associées aux objectifs comme égales à 1. En l'absence d'information (typiquement, à l'initialisation du système si l'observateur ne dispose d'aucune croyance préalable sur les intentions des autres agents, ou dans le cas où un agent est dans l'incapacité d'interpréter un comportement), cette probabilité est donc $P(g) = \frac{1}{|G|}, \forall g \in G$.

2.3 Gestion d'information

Puisque nous envisageons des agents capables d'employer une fonction de reconnaissance d'objectifs sur des comportements observés attribués à d'autres agents (voir Section 2.2), et employer l'information obtenue pour établir des relations de confiance (voir Section 2.1), il apparaît que le comportement d'un agent aura une influence sur ses relations avec les autres et pourrait influencer ses capacités à atteindre ses propres objectifs. L'information ainsi donnée aux autres agents au travers du comportement doit donc être prise en compte. Il est possible alors d'imaginer des agents cherchant à garder leurs intentions secrètes, pour éviter par exemple des sanctions ou un isolement, ou au contraire des agents tentant d'exposer le plus possible leurs objectifs au travers de leur comportement, afin de faciliter les coopérations et occuper un rôle central dans la société des agents autonomes artificiels.

Une stratégie de gestion de l'information ou IRS (pour *Information Regulation Strategy*) permet alors à un agent de prendre des décisions qui, en plus de lui permettre d'atteindre ses objectifs, vont permettre de minimiser ou maximiser l'information donnée. Afin de mesurer cette quantité d'information donnée à un observateur via un comportement hypothétique utilisant une fonction de reconnaissance d'objectif, l'agent calcule l'entropie de Shannon [27] de l'ensemble des probabilités que cette fonction attribue aux objectifs possibles. L'entropie $H_{b,K}$ correspondant au comportement b au regard d'un ensemble de connaissances K est obtenue grâce à la formule suivante :

$$H_{b,K} = - \sum_{g \in G} P(g | b, K) \cdot \log(P(g | b, K))$$

Deux stratégies de gestion de l'information sont alors envisageables : l'agent peut adopter une stratégie de *transparence*, c'est-à-dire minimiser l'entropie sur la distribution des probabilités afin de rendre son objectif réel le plus évident possible, ou au contraire adopter une stratégie de *dissimulation*, en maximisant l'entropie afin de maintenir

l'observateur dans un état de confusion le plus longtemps possible. Une stratégie de gestion de l'information est donc mise en œuvre au niveau du processus de prise de décision d'un agent, en planifiant ses actions non seulement pour atteindre les objectifs, mais également pour atteindre et maintenir une certaine certitude ou incertitude chez un observateur. Appliquer une telle stratégie peut nécessiter d'effectuer des actions qui seraient, en l'absence d'un tel critère de choix, considérées comme sous-optimales (plus longues, nécessitant plus de ressources, etc.). Nous parlons alors de coûts de mise en œuvre de ces stratégies.

Notons à ce stade que la stratégie de dissimulation est à distinguer de la définition la plus intuitive de mensonge ou de tromperie, qui consisterait à maximiser la probabilité associée à un autre but que celui réellement poursuivi par l'agent. La dissimulation ne cherche pas à créer de fausse croyance chez l'observateur, en l'amenant à croire que le comportement de l'agent vise à réaliser un autre but, mais seulement à maintenir une situation ambiguë et une absence d'information [5].

Après avoir montré comment des agents peuvent bâtir des relations de confiance à partir de diverses méthodes de reconnaissance d'objectifs, et introduit le concept de gestion d'information, il est nécessaire d'expliquer comment un agent peut choisir la stratégie appropriée compte tenu du contexte social et de ses propres objectifs. Nous allons maintenant nous attacher à montrer dans la suite de cet article en Section 3 comment employer ces stratégies de gestion de l'information en fonction du contexte social perçu par l'agent afin de maximiser ses espérances de gain (et minimiser les risques de sanctions).

3 Sélectionner une stratégie de gestion de l'information dans un système de confiance

Comme expliqué précédemment, nous proposons cette contribution comme un moyen de sélectionner la stratégie de gestion de l'information la plus appropriée compte tenu du contexte. Par contexte, nous entendons principalement les états mentaux produits par l'évaluation du comportement des autres (images et confiances, donc).

3.1 Évaluation de l'utilité des stratégies

Tout d'abord, le processus de sélection va évaluer l'utilité espérée de l'une ou l'autre stratégie dans le contexte présent selon les formules suivantes :

$$\begin{aligned} EU_T &= P_{C|T} \cdot \text{Benefit}_C - \text{Cost}_T \\ EU_O &= P_{C|O} \cdot \text{Benefit}_C - \text{Cost}_O \end{aligned}$$

Où l'utilité espérée d'une stratégie de transparence, EU_T (pour *Expected Utility of transparency*), dépend de la probabilité $P_{C|T}$ d'être inclus dans un collectif d'agents C en suivant cette stratégie, les bénéfices Benefit_C d'une telle inclusion et le coût Cost_T associés à la révélation de ses objectifs (ou perte des bénéfices liés à la confidentialité) et

incluant les coûts de mise en œuvre de cette stratégie, ainsi que d'éventuelles sanctions.

De manière analogue, l'utilité espérée d'une stratégie de dissimulation, EU_O (pour *Expected Utility of obfuscation*), dépend de la probabilité $P_{C|O}$ d'être inclus dans ce même collectif C tout en suivant cette stratégie et le coût $Cost_O$ de la mise en œuvre de la stratégie.

Trois cas de figures peuvent alors se présenter :

- si $EU_T > EU_O$, l'agent a alors intérêt à adopter une IRS de transparence
- si $EU_T < EU_O$, l'agent a alors intérêt à adopter une IRS de dissimulation
- si $EU_T = EU_O$, l'agent sait alors explicitement que les deux stratégies sont d'intérêt équivalent (il aura alors besoin d'une règle de bris d'égalité pour sélectionner l'une ou l'autre).

Les définitions précises des probabilités, coûts et bénéfices dépendent du contexte applicatif et ces éléments du modèle sont présentés ici de manière volontairement abstraite. Un exemple d'implémentation triviale mais concrète de ces fonctions est proposé en Section 4.1. Nous considérons à ce stade par simplicité que l'agent connaît leur valeur au travers d'une croyance. Il est par exemple envisageable de laisser l'agent réévaluer la probabilité $P_{C|T}$ d'être admis au sein d'un collectif en observant la proportion d'agents effectivement admis parmi ceux identifiés comme ayant eux aussi adopté cette IRS.

3.2 Sélection de la stratégie de régulation

Après avoir défini en Section 3.1 les modalités d'évaluation de l'utilité des stratégies de dissimulation et de transparence, nous illustrons ici comment l'agent sélectionne la stratégie la plus adaptée en fonction de sa perception du contexte, et plus particulièrement du contexte social.

Nous considérons ici les stratégies de gestion d'information évoquées en Section 2.3. Nous ajoutons à cela une stratégie appelée *indifférent*, consistant à ne pas tenir compte de la quantité d'information transmise à l'observateur lors de la prise de décision. Cette troisième stratégie sera employée lorsque l'agent considère que la gestion d'information n'est pas un critère pertinent dans ses interactions sociales pour influencer la décision des autres dans la formation de collectifs d'agents. Nous introduisons cette croyance $BsocImpact$ signifiant, lorsqu'elle est présente dans l'ensemble des croyances B de l'agent, que l'agent est convaincu que l'information transmise à un observateur *via* son comportement aura un impact social (c.-à-d. que la connaissance issue de l'observation de son comportement influence la décision des autres agents de l'inclure dans des coalitions ou de l'en exclure). La production et révision de cette croyance ne sera pas traitée en détail dans cet article mais par exemple, dans un système normatif, il est envisageable que l'agent vérifie au préalable si la poursuite de son objectif est acceptable ou non au regard de normes et d'institutions [8]. De même, il est envisageable que l'agent, s'il venait à être doté d'une fonction de reconnaissance de stratégie (voir Section 5), puisse évaluer si l'emploi d'une stratégie particulière est corrélée à une meilleure inclusion

ou exclusion.

Lors de la mise à jour de sa base de croyances, l'agent évalue la stratégie la plus appropriée en fonction de ses croyances et utilités espérées pour les différentes stratégies.

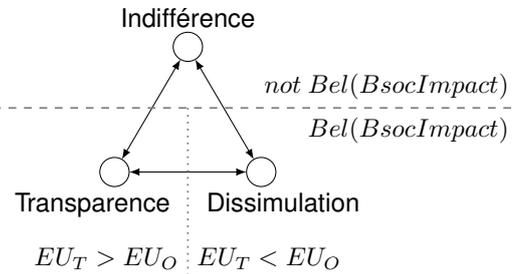


FIGURE 1 – Transitions entre stratégies

La figure 1, illustre la manière dont un agent peut évoluer d'une IRS à l'autre en fonction des utilités espérées et de ses croyances. Premièrement, seul un agent ayant la croyance $BsocImpact$ a intérêt à adopter une autre stratégie que la stratégie d'*indifférence*, les notions d'utilité espérée n'ayant aucune pertinence dans un tel contexte. Ensuite, la comparaison des utilités permet à un agent de basculer d'une stratégie de *transparence* à une stratégie de *dissimulation* et réciproquement. Le schéma ne fait volontairement pas mention de la règle de bris d'égalité discutée en section 3.1

4 Expérimentations

Nous avons fait le choix dans cet article de fournir un modèle qui propose qu'une définition volontairement abstraite de certains éléments (fonctions d'utilité, règle de bris d'égalité, etc.) en laissant au concepteur d'un réel système la tâche de les définir. Nous justifions ce choix par la nécessité de s'adapter à la grande diversité des domaines applicatifs et mécanismes de reconnaissance de but présentés en Section 2. Cette section présente un exemple arbitraire mais simple de mise en œuvre du modèle afin d'illustrer les effets de l'emploi de la contribution proposée dans cet article et explorer les limites de notre proposition. Nous avons choisi de présenter ici les résultats obtenus à la suite d'une démarche d'expérimentation. Pour cela, nous présenterons brièvement l'implémentation de ce modèle à l'aide du cadriciel JaCaMo [1] en Section 4.1. Ensuite nous illustrons les mécanismes présentés dans cet article au travers des résultats d'exécution d'un scénario en Section 4.2.

4.1 Mise en œuvre

Description du système

Nous avons choisi d'implémenter cette preuve de concept à l'aide du cadriciel multi-agent JaCaMo [1], connu dans la communauté pour offrir une architecture inspirée du modèle BDI [23] dont les croyances, plans et raisonnements peuvent être décrits à l'aide du langage AgentSpeak [3] dont la syntaxe est proche du paradigme de programmation logique. Jason, l'implémentation d'AgentSpeak proposée dans JaCaMo, propose de définir les plans

de l'agent en associant à un évènement déclencheur (ou *triggering event*), un contexte d'activation (une expression logique évaluée au moment de la sélection de plans), et *corps* formé d'une séquence d'actions internes (e.g. ajout/suppression de croyances, déclenchement d'évènements, ajout/suppression de buts) et externes (e.g. communication, interactions avec l'environnement). La description de l'environnement se fait à l'aide de la bibliothèque CArAgO [25] en langage Java et consiste principalement en la description d'*artefacts*, proposant aux agents des actions et pouvant informer ces agents via un système d'évènements. Dans l'expérimentation présentée ici, les agents sont tous connectés à un même artefact, proposant douze actions différentes $\{a_1, \dots, a_{12}\}$ permettant chacune de progresser vers un ou plusieurs objectifs. Tous les agents sont informés par l'artefact de l'exécution d'une action et de l'identité de son auteur lorsqu'elle est exécutée. Afin de représenter d'éventuelles imperfections dans la fonction de perception ou la fonction de reconnaissance d'objectifs, l'artefact possède un attribut $pfail \in [0; 1]$ modélisant la probabilité d'une erreur de perception et/ou d'interprétation par la GRF (voir Section 2.2), et aboutissant à un bruit plus ou moins significatif dans la construction des images, permettant de modéliser un environnement partiellement observable. Une valeur non-nulle de $pfail$ permet par exemple de modéliser la difficulté des agents à distinguer correctement des actions effectuées dans l'environnement. Cela peut venir par exemple d'erreurs de capteur, de problèmes de perception locale ou encore d'un manque d'occurrences de certains comportements dans une base d'apprentissage.

Les agents vont, à partir des images construites au cours de leurs observations du comportement des autres agents, accorder et retirer leurs confiances dans les objectifs des autres. Nous avons ici adopté un modèle simple dans lequel les probabilités fournies par la GRF sont utilisées comme images et à partir desquels la confiance est établie lors de dépassement d'une valeur seuil fixée à $P(G) = 0.6$. Notons que, la somme des probabilités étant normalisée à 1, il est impossible d'avoir ici confiance en l'intention d'un agent d'accomplir plusieurs objectifs simultanément.

Implantation des IRS

Les IRS, dans le cadre de cette implémentation, ont été introduites dans ce modèle sous forme de prédicats utilisés dans le contexte de plans. Ainsi, ces IRS sont employées lors de la sélection de plan afin de choisir celui qui (exception faite de l'*indifférence*) minimise ou maximise l'entropie dans la distribution des probabilités des objectifs.

Les objectifs des agents, l'IRS courante, les comportements observés des autres agents et les images et confiances construites à partir de ces observations sont représentés sous forme d'états mentaux dans la base de croyance. Ces états mentaux sont révisés au fur et à mesure de la perception successive d'actions exécutées dans l'environnement.

Pour observer le comportement du système, nous observerons les états mentaux d'un agent du système, et plus parti-

culièrement l'évolution des probabilités associées aux objectifs possibles des agents, l'entropie de cette distribution de probabilités (représentant donc une mesure de la quantité d'information détenue par l'observateur choisi), et les utilités associées aux différentes stratégies.

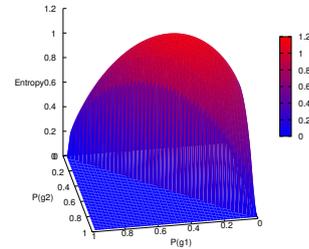


FIGURE 2 – Représentation de l'entropie dans un espace tridimensionnel décrit par les probabilités associées à trois buts différents.

La Figure 2, propose une représentation de l'entropie en fonction de la probabilité des buts. Ici, l'espace de toutes les distributions possibles des probabilités est projeté sur un plan, et la dimension verticale représente l'entropie $H_{b,L}$ correspondant à ces distributions. L'entropie est maximale pour $P(g_1) = P(g_2) = P(g_3) = \frac{1}{3}$, c'est-à-dire lorsque l'information détenue par l'observateur est minimale. À l'inverse, lorsque $\exists g \in G : P(g) = 1$, l'information est maximale et l'entropie associée $H_{b,L} = 0$. Comme on peut ainsi le voir sur cette figure, adopter une stratégie de *dissimulation*, telle que décrite formellement en Section 2.3 revient à rester autant que possible proche du sommet de cette fonction, où la confusion est maximale pour l'observateur. La stratégie de *transparence* en revanche consiste à adopter le comportement qui fera migrer cette entropie le plus vite possible vers le "pôle" correspondant à la probabilité maximale pour le but réel de l'agent, et en rester le plus proche possible. à ces pôles, l'entropie atteint une valeur nulle.

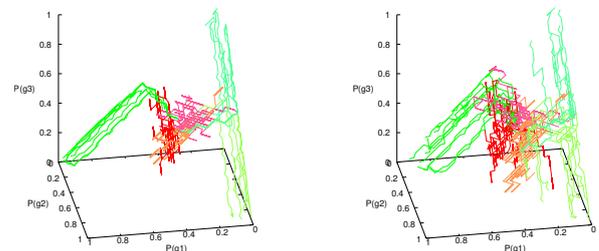


FIGURE 3 – Évolutions des probabilités des objectifs pour $pfail=0.1$ (à gauche) et $pfail=0.2$ (à droite)

La Figure 3 illustre du point de vue de l'un des agents du système, l'évolution des images de 60 agents dans un espace où chaque dimension correspond à la probabilité es-

timée de l'un des buts. Les probabilités associées à leurs objectifs sont initialisées à $P(g)_{t_0} = \frac{1}{|G|}$, point correspondant en Figure 2 au sommet de la courbe, où l'entropie est maximale.

En couleurs chaudes, des agents ayant adopté une IRS de dissimulation tendent à rester le plus loin possible des pôles, c'est-à-dire là où l'entropie sur les probabilités est maximale. En revanche, les agents ayant adopté une IRS de transparence migrent vers les trois pôles de l'intersection formée entre l'espace et le plan défini par l'équation $P(g_1) + P(g_2) + P(g_3) = 1$, comme défini en Section 2.2. Plus la probabilité *pfail* d'erreur de perception et d'interprétation est importante, plus les agents ayant un même objectif et une même IRS sont dispersés sous l'effet des erreurs de perception. Cette dispersion peut aller jusqu'à conduire un observateur à accorder par erreur sa confiance à un agent.

Selection d'une IRS

Le processus d'évaluation des IRS est appelé à chaque perception de l'agent, vérifiant l'existence ou l'absence de la croyance *BsocImpact* et révisant, s'il y a lieu, les utilités espérées pour les stratégies de transparence et de dissimulation. Nous proposons dans le cadre de cette implémentation, d'évaluer les utilités espérées à l'aide des valeurs suivantes :

$$Benefit_C = \sum_{ag \in Ag} benefit(ag)$$

avec

$$benefit(ag) = \begin{cases} \frac{1}{|Ag|} ssi \ trust(ag, g) \wedge myGoal(g) \\ 0 \ sinon \end{cases}$$

ainsi que

$$Cost_T = \sum_{ag \in Ag} cost_T(ag)$$

avec

$$cost_T(ag) = \begin{cases} \frac{1}{|Ag|} ssi \ trust(ag, g) \wedge not \ myGoal(g) \\ \frac{1}{2 \cdot |Ag|} ssi \ \nexists g \in G \ s.t. \ trust(ag, g) \\ 0 \ sinon \end{cases}$$

et

$$Cost_O = 0$$

$$P_{C|T} = 0.9$$

$$P_{C|O} = 0.1$$

Avec de telles définitions des fonctions de coûts et bénéfices, les agents qui n'adoptent pas une IRS d'indifférence seront incités à dissimuler leurs objectifs tant qu'ils n'ont pas confiance en une proportion suffisante d'agents pour poursuivre des objectifs similaires au siens.

4.2 Résultats

Nous proposons d'étudier le comportement du système au travers d'un premier scénario. Dans ce scénario, deux agents *ir1* et *ir2* sont initialisés avec pour objectifs respectifs *g1* et *g2*, et la croyance *BsocImpact* (voir Section 3.2). Deux autres agents *ir3* et *ir4*, sont initialisés quand à eux avec un même objectif *g1*, mais sans la croyance *BsocImpact*. Les agents sont initialisés sans *a priori* sur les buts des autres agents du système. Le même scénario est joué à de multiples reprises afin d'observer les propriétés du système, peu importe les situations particulières créées par les aléas venant de l'absence d'ordonnancement dans la plateforme et les erreurs de perceptions des agents.

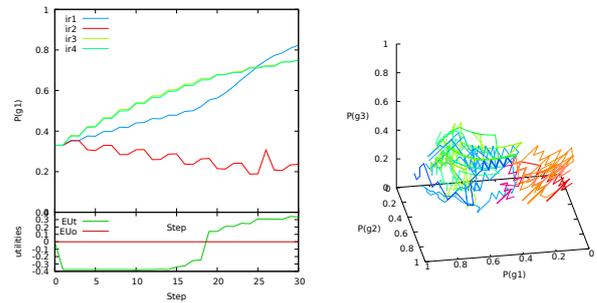


FIGURE 4 – Observation de l'évolution des images des agents associés au but *g1* et des utilités espérées (figure de gauche), et évolution des probabilités associés aux divers buts possibles (figure de droite), du point de vue de l'agent *ir1*. Résultats expérimentaux agrégés sur 100 runs, avec *pfail*=0.05.

La Figure 4 montre le point de vue de l'agent *ir1* et l'évolution de sa vision des autres agents au cours de la simulation. Comme l'agent ne dispose pas d'informations au départ sur les objectifs des autres agents, l'utilité espérée d'une IRS de transparence tombe dès les premières observations à un niveau inférieur à celle de l'IRS de dissimulation.

Les agents *ir3* et *ir4*, qui ne disposent pas de la croyance *BsocImpact*, adoptent un comportement qui va accroître de manière régulière la probabilité associée à leur objectif, tandis que pour *ir1* et *ir2*, l'usage d'une IRS de dissimulation leur permet de ne donner que peu d'informations aux observateurs (ce qui se traduit par un accroissement moindre de la probabilité P(*g1*) durant les premières actions de la simulation).

Lorsque les images des agents *ir3* et *ir4* sont assez significatives pour permettre à l'agent *ir1* d'accorder sa confiance, l'utilité de l'IRS de transparence remonte brutalement en raison de l'intérêt perçu par *ir1* de dévoiler son objectif à des observateurs qui semblent majoritairement le partager. L'agent change alors sa stratégie (le changement est explicitement mentionné dans les logs). L'image que l'agent *ir1* construit à partir de l'observation de son propre comportement entame alors une brusque ascension.

Pour l'agent *ir2* en revanche, les agents *ir3* et *ir4*, puis *ir1* étant identifiés comme poursuivant un autre objectif que le sien, l'IRS de dissimulation reste celle ayant l'uti-

lité la plus élevée. La figure de droite montre comment, au cours des multiples itérations de l'exécution de ce scénario, l'agent ir_2 –représenté en couleurs chaudes– tend à rester au centre de l'espace des probabilités, où l'entropie est maximale (comme illustré en Figure 2, Section 4.1).

Ce scénario illustre comment la sélection d'IRS, via les définitions d'utilités exposées ici, permet de décrire des agents prudents et opportunistes, c'est-à-dire attendant d'avoir obtenu suffisamment d'information sur le comportement des autres agents du système pour décider ou non de dévoiler leurs propres objectifs au travers de leur comportement.

5 Conclusion

Au travers de cet article, nous avons tout d'abord rappelé les enjeux de la prise en compte de l'information transmise aux observateurs dans des systèmes où la confiance des agents envers les autres peut impacter leur performance. Nous avons montré comment la littérature propose de doter les agents de fonctions de reconnaissance d'objectifs à partir de comportements observés et inférer ainsi l'objectif le plus vraisemblable d'un agent au regard d'un ensemble de connaissances. Nous avons ensuite montré comment la gestion d'information permet à un agent de prendre connaissance de la quantité d'information transmise à un observateur via son comportement, et comment anticiper cela dans sa prise de décision à l'aide d'une stratégie de gestion de l'information. Cet article a ensuite présenté une proposition de mécanisme de sélection de la stratégie de gestion de l'information en fonction du contexte, et plus particulièrement des relations de confiance établies avec les autres agents à partir de ses propres observations. Ce mécanisme de sélection repose sur une évaluation des utilités des stratégies à disposition en fonction des bénéfices espérés en dévoilant ses objectifs, comparés aux coûts associés à d'éventuelles sanctions en réaction à une telle révélation. Enfin, une preuve de concept illustre la mise en œuvre de ce modèle à travers un système implémenté et testé au travers d'un scénario dans lequel les agents vont employer ce processus pour changer de stratégie en fonction de l'évolution de la situation.

Un tel modèle présente tout son intérêt dans les domaines applicatifs où les agents peuvent poursuivre plusieurs buts différents et ont des raisons valables de ne pas désirer voir certains autres agents accomplir certains buts (par exemple pour des motivations stratégiques ou éthiques) et où les agents peuvent avoir des interactions sociales conditionnées par une certaine confiance (formation de coalitions, construction d'organisation, partage de ressources, coordination, partage d'information, etc.). Par exemple, un tel modèle pourrait être pertinent dans le cadre de gestion éthique d'actifs financiers [6], où des agents dotés de théories morales différentes permettant de définir ce qu'est un investissement éthique cherchent à déterminer les partenaires appropriés pour des échanges de gré-à-gré.

Les résultats expérimentaux montrent comment, dans un cadre abstrait et simple, les agents peuvent adapter leur

stratégie à l'évolution de leur image des autres agents, et ainsi faire preuve d'une forme de prudence dans la révélation de leurs objectifs. Toutefois, nous aimerions voir ces résultats confirmés dans le cadre d'un déploiement dans un domaine applicatif réaliste, et pourquoi pas avec des agents humains (dans un cadre vidéoludique par exemple).

Ce modèle repose également sur de nombreuses hypothèses : l'hypothèse de l'efficacité des perceptions et de la fonction de reconnaissance d'objectif par exemple. Il serait intéressant à ce titre d'explorer les effets d'une hétérogénéité de points de vue ou de fonctions implémentées, pour modéliser par exemple le cas d'agents dotés de capteurs, d'architectures ou de raisonnements différents.

De plus, nous avons proposé ici un modèle de sélection de la stratégie de gestion de l'information en nous reposant sur un système de confiance, et il serait intéressant de l'étendre à la prise en compte de systèmes de réputation [26] en laissant les agents membres d'un collectif transmettre leurs états mentaux. Une telle extension serait d'autant plus intéressante pour les contextes applicatifs dans lesquels les agents ne disposent que d'une perception locale et donc avoir des points de vues divergeant sur un même agent en raison de leurs expériences individuelles.

Enfin, de manière analogue à la *fonction de reconnaissance de buts* utilisée par les agents, nous aimerions proposer une *fonction de reconnaissance de stratégies*, permettant à un agent de reconnaître avec une certaine probabilité la stratégie de gestion de l'information employée par un autre agent. Une telle fonctionnalité ouvrirait la voie à la construction de confiance sur la gestion de l'information et permettrait par exemple de faire émerger des groupes d'agents qui, en plus de se considérer mutuellement comme cherchant à atteindre des buts acceptables, font preuve d'une transparence qui serait également appréciée. De tels modèles pourraient également présenter un intérêt pour accorder des privilèges dans des systèmes ayant de forts enjeux de sécurité en réservant l'accès à certaines ressources aux agents dont l'objectif est identifié et accepté, et semblant adopter une stratégie de transparence. À l'inverse il est possible d'imaginer aussi que l'emploi efficace d'une stratégie de dissimulation soit valorisée dans certains groupes, par exemple lorsque la réussite des objectifs du groupe est conditionnée par l'ignorance des agents exclus du collectif (par exemple dans le cas de jeux de stratégie ou pour modéliser des problèmes de *privacy*).

Informations complémentaires et reproductibilité des résultats

L'implémentation de la preuve de concept, des informations complémentaires sur les mécanismes implémentés et les données employées pour produire les figures sont mis à disposition sur un dépôt en ligne. Il est également possible de créer de nouveaux scénarios en modifiant les agents et leurs croyances initiales. L'ensemble peut être téléchargé en clonant le dépôt git à l'adresse suivante :

<https://gitlab.com/NCointe/poc-jfsma20>

Remerciements

Ce travail a été réalisé grâce au support financier du Conseil Européen de la Recherche (ERC) au sein du programme recherche et innovation Horizon 2020 de l'Union Européenne via le projet BEHAVE (numéro d'agrément 724431).

Références

- [1] Olivier Boissier, Rafael H Bordini, Jomi F Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6) :747–761, 2013.
- [2] Grégory Bonnet. A protocol based on a game-theoretic dilemma to prevent malicious coalitions in reputation systems. In *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31, 2012*, pages 187–192, 2012.
- [3] Rafael H Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons, 2007.
- [4] Zack Butler, Robert Fitch, and Daniela Rus. Distributed control for unit-compressible robots : goal-recognition, locomotion, and splitting. *IEEE/ASME transactions on mechatronics*, 7(4) :418–430, 2002.
- [5] CG Chorus. How to keep your av on the right track ? an obfuscation-based model of decision-making by autonomous agents. *hEART 2018*, 2018.
- [6] Nicolas Cointe, Grégory Bonnet, and Olivier Boissier. Ethics-based cooperation in multi-agent systems. In *14th Social Simulation Conference (SSC18)*, August 2018.
- [7] Nicolas Cointe, Amineh Ghorbani, and Caspar Chorus. Management of obfuscation-based decision making in a coalition. In *15th Social Simulation Conference (SSC19)*, September 2019.
- [8] Virginia Dignum and Julian Padget. *Multiagent organizations*, chapter 2, pages 51–98. MIT press, second edition, 2013.
- [9] Diego Gambetta et al. Can we trust trust. *Trust : Making and breaking cooperative relations*, 13 :213–237, 2000.
- [10] Christopher W Geib. Delaying commitment in plan recognition using combinatory categorial grammars. In *IJCAI*, pages 1702–1707, 2009.
- [11] Christopher W Geib and Robert P Goldman. Partial observability and probabilistic plan/goal recognition. In *Proceedings of the International workshop on modeling other agents from observations (MOO-05)*, volume 8, pages 1–6, 2005.
- [12] Christopher W Geib, John Maraist, and Robert P Goldman. A new probabilistic plan recognition algorithm based on string rewriting. In *ICAPS*, pages 91–98, 2008.
- [13] Amineh Ghorbani and Giangiacomo Bravo. Managing the commons : a simple model of the emergence of institutions through collective action. *International Journal of the Commons*, 10(1), 2016.
- [14] Valerie Guralnik and Karen Zita Haigh. Learning models of human behaviour with sequential patterns. In *Proceedings of the AAAI-02 workshop "Automation as Caregiver"*, pages 24–30, 2002.
- [15] Froduald Kabanza, Julien Filion, Abder Rezak Benaskeur, and Hengameh Irandoust. Controlling the hypothesis space in probabilistic plan recognition. In *IJCAI*, pages 2306–2312, 2013.
- [16] Andreas D Lattner, Andrea Miene, Ubbo Visser, and Otthein Herzog. Sequential pattern mining for situation and behavior prediction in simulated robotic soccer. In *Robot Soccer World Cup*, pages 118–129. Springer, 2005.
- [17] Samhar Mahmoud, Simon Miles, and Michael Luck. Cooperation emergence under resource-constrained peer punishment. In *AAMAS*, pages 900–908, 2016.
- [18] D Harrison McKnight and Norman L Chervany. What trust means in e-commerce customer relationships : An interdisciplinary conceptual typology. *International journal of electronic commerce*, 6(2) :35–59, 2001.
- [19] Reuth Mirsky et al. Slim : Semi-lazy inference mechanism for plan recognition. *arXiv preprint arXiv :1703.00838*, 2017.
- [20] Reuth Mirsky, Roni Stern, Kobi Gal, and Meir Kalech. Sequential plan recognition : An iterative approach to disambiguating between hypotheses. *Artificial Intelligence*, 260 :51–73, 2018.
- [21] G. Muller, L. Vercoouter, and O. Boissier. Towards a general definition of trust and its application to openness in MAS. In *6th Workshop on Deception, Fraud and Trust in Agent Societies*, pages 49–56, 2003.
- [22] Jérémy Patrix, Abdel-Allah Mouaddib, Simon Le Gloannec, Dafni Stampouli, and Marc Contat. Discrete relative states to learn and recognize goals-based behaviors of groups. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 933–940. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [23] Anand S Rao, Michael P Georgeff, et al. BDI agents : from theory to practice. In *ICMAS*, volume 95, pages 312–319, 1995.
- [24] Lars Rasmusson and Sverker Jansson. Simulated social control for secure internet commerce (position paper). In *Proceedings, New Security Paradigms Workshop, Lake Arrowhead*, 1996.

- [25] Alessandro Ricci, Michele Piunti, Mirko Viroli, and Andrea Omicini. Environment programming in CARtAgO. In *Multi-agent programming*, pages 259–288. Springer, 2009.
- [26] Jordi Sabater-Mir and Laurent Vercoeur. Trust and reputation in multiagent systems. *Multiagent systems*, page 381, 2013.
- [27] CE Shannon and W Weaver. *The mathematical theory of communications*. University of Illinois press. Urbana. 1963.
- [28] Subhasis Thakur, Guido Governatori, and Abdul Sattar. On modeling punishment in multi-agent systems. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1375–1376. Citeseer, 2012.

Micro-Motivations du Second Ordre : application à la ségrégation spatiale *

P. Collard

Université Côte d'Azur, CNRS, laboratoire I3S

philippe.collard@univ-cotedazur.fr

Résumé

Cet article porte sur l'écart entre les préférences individuels des individus et leurs conséquences au niveau collectif. Le modèle de ségrégation de Sakoda-Schelling est représentatif de cette problématique ; un critère de discrimination est utilisé pour décider si une personne "insatisfaite" quittera ou non sa place ; sur la base du ratio de personnes "acceptables" autour d'un individu, des simulations simples montrent que même les agents tolérants finissent par être agrégés spatialement bien au-delà de l'exigence locale de leur niveau de tolérance. Nous utilisons ici un critère du second ordre pour inciter les individus à quitter leur place. Cet article fournit des éléments de réponse à la question : un tel critère contribue-t-il à réduire l'écart entre les micro-motivations et leur macro-conséquence ?

Mots-clés

Simulation multi-agent, Modèles de comportement agent, Lien micro-macro.

Abstract

This paper deals with the gap between the preferences of individual agents and collective outcomes. The typical example of this topic is the Schelling-Sakoda segregation model where a selfish discrimination criterion is used to decide whether or not a person will leave his place ; based on the ratio of "acceptable" people around an individual, simple simulations show that even tolerant agents end up being spatially aggregated far beyond the local requirement of their tolerance level. Here we use a second order discrimination criterion to compel people to leave their places. This article provides some answers to the question : does such a criterion contribute or not to reduce the gap between micromotives and macrobehaviour ?

Keywords

Multi-agent simulation, Models of agent behaviour, Micro-Macro link.

Introduction

La sociologie computationnelle et la modélisation à base d'agents constituent le cadre général de cet article [?].

*Une version étendue de cet article a été publiée dans le *Journal of Computational Social Science* [?]

Nous nous intéressons au lien entre le *microscopique*, qui est centré sur les *individus* et ses interactions avec les autres, et le *macroscopique*, qui est centré sur la *société* [?]. La grande majorité des personnes pensent que leur propre comportement, s'il était appliqué par tous, aurait des effets congruents au niveau de la collectivité ; le mérite du modèle de Sakoda-Schelling a été de montrer que cela n'est pas nécessairement le cas [?][?][?][?]; en effet, dans le contexte spécifique de la ségrégation résidentielle, Schelling a montré *via* des simulations que même les agents plutôt tolérants finissent par être agrégés spatialement bien au-delà de l'exigence locale de leur niveau de tolérance. Ce modèle a été étendu à des seuils de tolérance individuels, à des réseaux d'influence autres qu'une simple grille, à des critères de discrimination à plusieurs niveaux, etc. [?][?][?][?].

Le terme "second-ordre" est utilisée dans de nombreux domaines comme les Mathématiques, la Technologie, la Psychologie [?], la Philosophie [?], etc. et s'applique à une variété de concepts. En sociologie computationnelle, Nigel Gilbert a été le premier à proposer une version du modèle de Schelling où les "propriétés du second ordre" font référence à une rétroaction du niveau macro vers le niveau micro ; plus précisément, il écrit "*l'émergence du second ordre se produit lorsque les agents reconnaissent des phénomènes émergents, tels que des sociétés, des clubs, des organisations formelles, des institutions, des localités, et ainsi de suite, où le fait que vous soyez membre, ou non membre, change les règles d'interaction entre vous et les autres agents*" [?]. Le présent article n'adopte pas ce point de vue où des phénomènes macro émergents influencent le niveau micro ; nous supposons plutôt qu'une décision du second ordre est motivée par l'effet immédiat qu'un micro comportement primaire pourrait avoir.

Dans la première section de cet article, nous présentons le modèle de ségrégation de Sakoda-Schelling et les outils qui permettent de mesurer le degré de ségrégation et l'écart entre les niveaux micro et macro. La principale contribution de l'article est présentée dans la deuxième section : le concept de *micro-otivation du second ordre* est introduit et discuté. Dans la troisième section, nous simulons les deux approches et exploitons les résultats expérimentaux. Enfin, nous concluons et proposons des lignes d'action possibles pour de futurs travaux.

1 Modèle de ségrégation de Schelling

On suppose que (i) le monde est une *grille* bi-dimensionnel constituée de cellules peuplées d'agents qui représente un espace urbain idéalisé où vivent les individus, (ii) chaque cellule représente un site d'habitation qui peut être ou non occupé par un agent ; il y a donc au plus un agent par cellule. Soit P la population et δ la densité d'agents sur la grille.

1.1 Type, location et voisinage

Un agent a_i dans P est décrit par son *type* t_i qui reste constant dans le temps. Il y a deux valeurs possibles pour le type $\{\alpha, \beta\}$ qui représentent par exemple le sexe : homme ou femme, la langue maternelle : anglophone ou non, le groupe ethnique, la position sociale, etc. On suppose qu'il y a parité : $\#\{a_i \in P | t_i = \alpha\} = \#\{a_i \in P | t_i = \beta\}$. Le temps est discrétisé ; à chaque pas de temps un agent peut se déplacer de cellules en cellules et donc sa position peut changer au cours du temps. A un instant donné, le voisinage d'un agent est composé des agents localisés dans les huit cellules qui l'entourent. Pour chaque agent $a_i \in P$, soit $V_i(t)$ l'ensemble de ses voisins immédiat au temps t ; le cardinal $\#V_i(t)$ est alors une mesure d'influence et bien évidemment $0 \leq \#V_i \leq 8$.

1.2 Micro-motivations

A chaque pas de temps, un agent insatisfait au regard de la composition de son voisinage est motivé pour se déplacer vers une autre cellule, alors qu'un agent satisfait n'a aucune incitation à bouger. Si tous les agents sont satisfaits à un pas de temps donné alors il n'y a plus aucun mouvement et cela correspond à un *point fixe* de la dynamique.

La satisfaction d'un individu dépend de ses préférences quant à la composition de son voisinage ; plus précisément, en fonction de son propre type et du type de ses voisins, sa satisfaction dépend du ratio de voisins différents.

La mesure de dissemblance permet de mesurer ce ratio : soit $\tilde{P}(t) = \{a_k \in P | V_k(t) \neq \emptyset\}$ l'ensemble des agents non-isolés, on définit la *dissemblance* d'un agent $a_i \in \tilde{P}(t)$ par :

$$d_i(t) = \frac{\#\Delta_i(t)}{\#V_i(t)} \quad (1)$$

où $\Delta_i(t) = \{a_k \in V_i(t) | t_k \neq t_i\}$ est l'ensemble des voisins avec un trait opposé.

L'indicateur booléen de satisfaction est alors défini comme suit :

$$\Sigma_i(t) = \begin{cases} d_i(t) \leq \tau & \text{si } (V_i(t) \neq \emptyset) \\ true & \text{sinon} \end{cases} \quad (2)$$

où τ est un nombre réel¹ constant dans l'intervalle $[0..1]$ qui indique le seuil en dessous duquel un agent est satisfait ; on l'appela la *tolérance* et on supposera que sa valeur est commune à tous les agents.

1. En fait il n'existe qu'un nombre limité de valeurs rationnelles significatives pour la tolérance

En ce qui concerne le problème de trouver une nouvelle place, on suppose qu'il y a toujours quelques places vacantes sur la grille (i.e. $\delta < 1$) ; par conséquent, chaque agent insatisfait pourra choisir au hasard une place sur la grille et s'y rendre si et seulement si elle est vacante.

1.3 Macro-comportement

Le macro comportement est caractérisé par la configuration spatiale de la population sur la grille lorsque chacun est satisfait et donc que personne ne change de place ; on note t^* le pas de temps où un tel événement se produit pour la première fois.

La satisfaction au pas de temps t est la proportion d'agents satisfaits dans la population :

$$sat(t) = \frac{\#\{a_i \in P | \Sigma_i(t) = true\}}{\#P} \quad (3)$$

Bien évidemment, $sat(t^*) = 1$.

L'index de ségrégation permet de mesurer le niveau global de ségrégation dans la population à la convergence. On reformule les mesures proposées par Carrington [?][?] et Goffette-Nagot [?]. En premier lieu, on définit une mesure globale de *similarité* :

$$sim(t) = \frac{1}{\#\tilde{P}(t)} \sum_{i=1}^{\#\tilde{P}(t)} (1 - d_i(t))$$

L'index de *ségrégation* est alors défini comme suit :

$$seg = \begin{cases} \frac{sim(t^*) - sim(0)}{1 - sim(0)} & \text{si } sim(t^*) \geq sim(0) \\ \frac{sim(t^*) - sim(0)}{sim(0)} & \text{sinon} \end{cases} \quad (4)$$

où $sim(0)$ est la valeur attendue de la mesure de similarité résultant d'une répartition aléatoire des agents sur la grille ; initialement, avec des positionnements aléatoires, on obtient une configuration proche de cela. La valeur maximale de 1 correspond à une configuration avec deux gros clusters homogènes, tandis que les valeurs négatives indiquent des populations très mélangées.

L'écart micro-macro entre les micro-motivations et le macro comportement souligné par Schelling est dû au chevauchement des voisinages : un agent qui se déplace en fonction de sa propre motivation, affecte non seulement le voisinage qu'il quitte et celui où il arrive, mais aussi, sur le long terme, tous les agents [?].

Le micro niveau est caractérisé par la tolérance τ et le macro niveau par la dissemblance moyennée sur la population. La différence entre ces deux mesures fournit donc une indication pertinente de l'écart micro-macro.

2 Micro-motivations du second ordre

Cette section décrit la contribution principale de l'article. Nous présentons les questionnements auxquels sont confrontés les individus dès lors qu'ils veulent créer, rejeter ou maintenir des liens sociaux. C'est dans ce contexte que nous définissons précisément le concept de motivation du second ordre.

2.1 Contexte

Pour établir des liens sociaux, les individus utilisent des méta-perceptions, ou des croyances sur la façon dont les autres les perçoivent ; comme l'a écrit Erika Carlson : " *les gens ont tendance à aimer les personnes qui ont une perception exacte d'eux-mêmes, mais les gens ont tendance à apprécier davantage leurs propres relations avec les personnes qui, selon eux, les voient de façon positive* " [?]. E. Carlson et al. ont mené des études qui suggèrent que " *les gens peuvent faire des distinctions valables entre la façon dont ils se voient et celle dont les autres les voient. Ainsi, les personnes semblent avoir une véritable idée de leur réputation ...* " [?] [?].

Face à un choix difficile à faire d'un point de vue social, les individus ont souvent des comportements de contournement qui peuvent être dictés par *ce qu'on va dire* ou *ce qu'on pense de soi*. Pour discriminer les autres personnes, les individus ont développé des stratégies élaborées, souvent inconscientes, voire involontaires ; par exemple, afin d'éviter une situation où il y a trop de personnes dissemblables dans son voisinage, plutôt que de paraître "négatif", les individus (i) préfèrent invoquer une raison complètement différente, comme le fait que leur lieu de résidence actuel est trop petit ou pas assez confortable, (ii) peuvent faire porter la décision par un tiers, par exemple en demandant à une personne similaire qui n'est pas son voisin de le rejoindre, (iii) utilisent, comme nous le faisons dans cet article, un critère de second ordre qui aura finalement un macro-effet similaire à celui du premier ordre.

Penser que les individus ont tendance à optimiser leur propre intérêt est souvent inexact ; parfois, on adopte une stratégie de second ordre pour simplifier la décision et on "cherche" ainsi à être ignorant ; la raison peut être que la connaissance, comme l'appartenance à une communauté, cause un préjudice à soi-même ou aux autres [?].

2.2 Définition

Dorénavant, nous ferons référence au critère utilisé dans le modèle de Schelling comme le critère de discrimination primaire ou du *premier ordre*. Dans cet article, nous proposons une nouvelle approche basée sur un critère du *second ordre*.

Le concept de *micro-motivation du second ordre* repose sur l'idée que la satisfaction primaire devient une sorte de *méta-état* ; en d'autres termes, pour une décision du second ordre, la satisfaction d'un agent ne dépend que de la satisfaction de ses voisins selon le critère primaire.

Soit $a_i \in P$, au pas de temps t , on définit l'indicateur booléen de *satisfaction au second ordre* comme :

$$\hat{\Sigma}_i(t) = \begin{cases} \frac{\#\hat{\Delta}_i(t)}{\#V_i(t)} \leq \hat{\tau} & \text{si } (V_i(t) \neq \emptyset) \\ true & \text{sinon} \end{cases} \quad (5)$$

où $\hat{\Delta}_i(t) = \{a_k \in V_i(t) | \Sigma_k(t) = false\}$ est l'ensemble des voisins insatisfaits selon le critère primaire et $\hat{\tau}$ est un nombre réel dans l'intervalle $[0..1]$ qui indique le seuil sous lequel un agent est satisfait au second ordre ; on l'appelle le

seuil de second-ordre et on suppose que sa valeur constante est commune à tous les agents.

A chaque pas de temps, cela exige que chaque agent connaisse la satisfaction primaire de ses voisins. On note t^* le premier pas de temps à partir duquel tout le monde est satisfait au second ordre ; évidemment, à ce pas de temps, il n'y a aucune garantie que les agents soient également satisfaits selon le critère primaire. Sans préjuger de la nature de la convergence, notons qu'une population où tous les agents sont satisfaits selon le critère primaire est également un point fixe pour la dynamique de second ordre.

3 Simulations expérimentales

Les simulations sont réalisées avec une implémentation des modèles dans l'environnement programmable multi-agents *NetLogo* [?].

Le monde est une grille carrée de 2500 cellules. Pour éviter les effets de bord, on impose que le monde "wrap" horizontalement et verticalement ; chaque cellule a donc exactement huit cellules voisines. Pour permettre à un agent insatisfait de se déplacer, certaines cellules sont vacantes alors que d'autres sont occupées ; on fixe la densité δ des agents à 0,9 qui est une valeur de densité couramment utilisée dans les différentes variantes du modèle de Schelling [?]. Relativement aux équations ?? et ??, on considère les sept valeurs de seuils de la forme $k/8$ avec $k \in [1; 8[$ correspondant à des comportements locaux allant d'intolérants à tolérants. En raison de la configuration initiale aléatoire et de la stratégie de recherche d'une place vacante, les modèles sont probabilistes ; aussi, sauf indication contraire, tous les résultats quantitatifs seront moyennés sur 50 exécutions indépendants.

3.1 Mesures agrégées

Dans un premier temps, on utilisera des mesures agrégées comme la *satisfaction* globale (eq. ??), l'index de *ségrégation* (eq. ??) et la *dissemblance* globale (eq. ??) pour mettre en évidence les différences entre les deux approches.

3.1.1 Premier ordre

Ici l'unique paramètre est la tolérance τ . On observera la population à la convergence au temps t^* lorsque tous les agents sont satisfaits selon le critère primaire.

– Si les agents sont très intolérants ($\tau = 0,125$), il n'y a pas de convergence ; par contre, dès que $\tau \geq 0,25$, la dynamique atteint un point fixe (Tab. ?? colonne 2).

– Si les agents sont plutôt intolérants ($\tau = 0,250$ ou $\tau = 0,375$), la dynamique conduit à une forte ségrégation et la frontière entre les deux communautés est une bande étroite composée de places vacantes [?] (fig. ?? : courbe supérieure).

– Le phénomène le plus inattendu est lorsque les agents sont plutôt tolérants ($\tau = 0,50$ ou $\tau = 0,625$) car la dynamique conduit toujours à la ségrégation (Tab. ?? colonne 3) (fig. ?? : courbe supérieure) ; rappelons qu'un index de ségrégation nul correspond à une configuration initiale aléatoire et que la valeur maximale de τ correspond à une configuration de deux gros clusters homogènes.

La dissemblance moyenne augmente régulièrement avec τ ; et, de plus, elle est bien inférieure à ce qui est requis sur la base de la tolérance (fig. ?? : courbe inférieure) (Tab. ?? colonne 3).

Ces premières expériences confirment les résultats obtenus par Schelling [?][?], notamment sur la mise en évidence d'un hiatus entre les niveaux micro et macro, et serviront de base de comparaison dans ce qui suit.

3.1.2 Second ordre

Les paramètres sont la tolérance τ et le seuil $\hat{\tau}$. On observera la population à la convergence au temps t^* lorsque tous les agents sont satisfaits au second ordre.

Pour chaque valeur de τ il existe plusieurs valeurs de $\hat{\tau}$ pour lesquelles la population converge ; soit $\hat{\tau}_{min}(\tau)$ la plus petite de ces valeurs. La table ?? montre la correspondance entre τ (colonne 1) et $\hat{\tau}_{min}$ (en gras) : un seuil de tolérance bas exige un seuil de second-ordre élevé et inversement si τ est élevé alors $\hat{\tau}$ doit être faible. Par exemple, pour obtenir une convergence avec $\tau = 0,125$, $\hat{\tau}$ devra au minimum être fixé à 0,625 ; en effet, si le seuil de second-ordre était également bas, il serait impossible de constituer ne serait-ce que l'embryon d'une zone mono-trait. Notons que le seuil $\hat{\tau}_{min}(\tau)$ donne le ratio de satisfaction de premier ordre le plus élevé parmi toutes les valeurs de seuil de second-ordre possibles pour la tolérance τ .

Afin de déterminer les mesures de ségrégation et de dissemblance au pas de temps t^* , pour chaque valeur de tolérance τ , on considère uniquement le seuil de second-ordre $\hat{\tau}_{min}(\tau)$.

– Quel que soit la tolérance τ , la dynamique conduit à un point fixe avec un ratio de satisfaction au premier ordre allant de 0,898 à 1 (Tab. ?? valeurs en gras).

– La dissemblance augmente avec la tolérance τ avec une rupture à $\tau = 0,5$ (Tab. ??) ; sa valeur est toujours inférieure à ce qui est requis sur la base de la tolérance (fig. ?? courbe du milieu) mais est bien supérieure à celle obtenue avec le modèle du premier ordre (fig.?? : courbe du bas). Cette rupture est confirmée par la valeur élevée de l'index de ségrégation pour les agents intolérants ($\tau \leq 0,5$) et la valeur basse pour les agents tolérants (Tab. ??) (fig. ?? : courbe du bas).

3.2 Espace de ségrégation

L'espace de ségrégation fait référence à la représentation graphique de la grille à la convergence ; il fournit une vue qualitative du phénomène de ségrégation. Bien au-delà des indicateurs agrégés, la comparaison des espaces de ségrégation permet de mieux comprendre les différences entre les deux approches. On visualise les espaces de ségrégation pour quatre valeurs de tolérance allant de 0,250 à 0,625 (fig. ?? et fig. ??).

3.2.1 Premier ordre

Visualiser l'état de la grille au pas de temps t^* permet de mettre en évidence l'émergence de la ségrégation engendrée par des décisions de premier ordre ; on observe sur les figures ?? que le phénomène d'agrégation est d'autant plus important que la tolérance est faible.

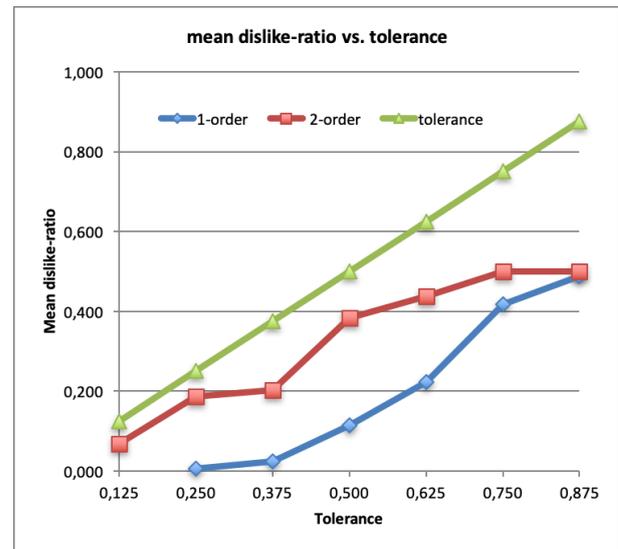


FIGURE 1 – Dissemblance *versus* Tolérance. De bas en haut : premier ordre, second ordre et tolérance. (Résultats moyennés sur 50 exécutions)



FIGURE 2 – Ségrégation *versus* Tolérance : second ordre (bas) et premier ordre (haut). (Résultats moyennés sur 50 exécutions)

TABLE 1 – Satisfaction du second ordre à \hat{t}^* (moyenne sur 50 exécutions avec (écart type))

τ \ $\hat{\tau}$	Satisfaction du premier ordre à \hat{t}^*	.125	.250	.375	.500	.625	.750
.125	-	-	-	-	-	.898 (.07)	.67
.250	1	-	-	-	.848 (.06)	.720	.57
.375	1	-	-	.890 (.01)	.740	.66	.55
.500	1	-	-	.864 (.07)	.740	.68	.620
.625	-	1	.959 (.05)	.870	.820	.800	.78
.750	1	1 (0)	.960	.950	.940	.930	.94

TABLE 2 – Dissemblance (moyenne sur 50 exécutions)

τ	$\hat{\tau}_{min}$	1-ordre à \hat{t}^*	2-ordre à \hat{t}^*
.125	.625	-	.067
.250	.500	.006	.189
.375	.375	.025	.185
.500	.375	.114	.383
.625	.250	.222	.438
.750	.125	.417	.500

TABLE 3 – Ségrégation (moyenne sur 50 exécutions)

τ	$\hat{\tau}_{min}$	1-ordre à \hat{t}^*	2-ordre à \hat{t}^*
.125	.625	-	.868
.250	.500	.990	.622
.375	.375	.950	.630
.500	.375	.770	.234
.625	.250	.560	.125
.750	.125	.170	.000

– Si les agents sont intolérants, les places vacantes (carrés noirs) sont utilisées pour "construire" la frontière entre les deux communautés; ceci explique pourquoi le modèle de Schelling ne converge pas si la densité est trop élevée (i.e. si le nombre de places vacantes est insuffisant) (fig. ?? et fig. ??).

– Le point clé ici est que la ségrégation se produit même si les agents sont plutôt tolérants; dans ce cas, les places vacantes ne jouent pas un rôle particulier et sont dispersées de façon aléatoire (fig. ?? et fig. ??).

3.2.2 Second ordre

L'espace de ségrégation au pas de temps \hat{t}^* apparaît sur les figures ??.

– Pour les agents intolérants, la ségrégation est importante et les places vacantes sont soit utilisées pour construire la frontière, soit dispersées aléatoirement dans l'espace

(fig. ?? et fig. ??).

– Pour les agents tolérants, la ségrégation est faible et les places vacantes sont regroupées et ne jouent aucun rôle particulier (fig. ?? et fig. ??).

Quelle que soit la tolérance, on trouve quelques regroupements de places vacantes et aussi quelques agents "marginiaux" insatisfaits au premier ordre (formes triangulaires).

3.3 Distribution de la dissemblance

Ici on examine la distribution sur l'ensemble des agents de la dissemblance (eq. ??) pour les valeurs de τ dans l'ensemble $\{0, 250; 0, 375; 0, 50; 0, 625\}$; afin de faciliter l'interprétation des histogrammes, sur chaque figure, la ligne verticale représente la tolérance (fig. ?? et fig. ??). Rappelons qu'initialement cette distribution est binomiale centrée.

3.3.1 Premier ordre

Les distributions sont tracées au temps t^* lorsque les agents sont satisfaits au premier ordre (fig. ??).

– La grande majorité des agents intolérants ont une dissemblance nul, c'est-à-dire qu'ils ne sont en contact qu'avec des agents de même trait (fig. ??, fig. ??).

– Pour les agents tolérants, bien que moins prononcés, ce phénomène persiste et la distribution *décroit* jusqu'à atteindre la valeur de tolérance (fig. ?? et fig. ??).

3.3.2 Second ordre

Les distributions sont tracées au pas de temps \hat{t}^* lorsque tous les agents sont satisfaits au second ordre (fig. ??).

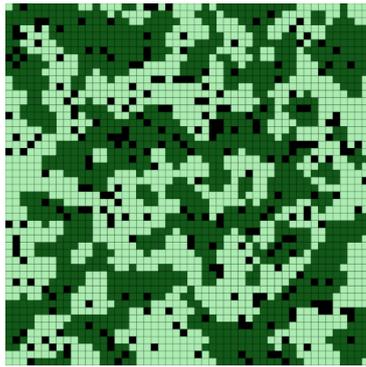
– La grande majorité des agents intolérants ont une dissemblance inférieure à la tolérance et la distribution décroît jusqu'à ce que la valeur de tolérance soit atteinte; cependant, il existe une minorité de "marginiaux" avec une dissemblance supérieure à la tolérance, ce qui signifie que ces agents sont satisfaits au second ordre mais insatisfaits au premier ordre et donc, dans une certaine mesure, isolés de leur propre communauté (figures ?? et ??).

– Les figures ?? et ?? montrent que la grande majorité des agents tolérants ont une dissemblance inférieure à la tolérance et que la distribution est *croît* régulièrement jusqu'à atteindre la valeur de tolérance. Là encore, une minorité d'agents a une dissemblance supérieure à la tolérance.

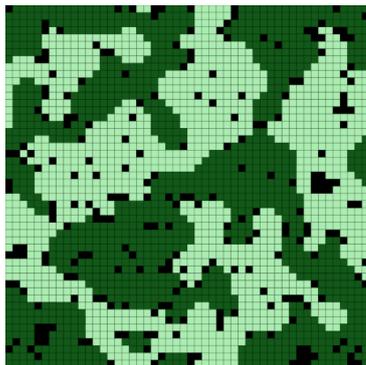
Les deux points principaux concernant les décisions du second ordre sont : (i) une inversion de la pente de la distribution en passant d'une population intolérante à une population tolérante; et, quel que soit le niveau de tolérance, (ii) l'existence d'un petit nombre de marginaux insatisfaits du premier ordre et satisfaits du second ordre.

3.3.3 Discussion

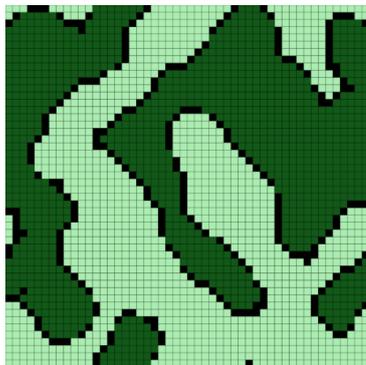
Les décisions du premier ordre conduisent à un écart important entre les niveaux micro et macro et cela même si les agents sont plutôt tolérants. Deux caractéristiques de la distribution de la dissemblance ont une signification au regard de cet écart : premièrement, une majorité substantielle d'agents ont une dissemblance nulle, deuxièmement, la pente de la distribution diminue rapidement; la dyna-



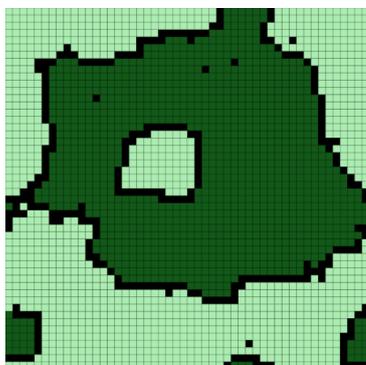
(a) $\tau=0.625$



(b) $\tau=0.500$

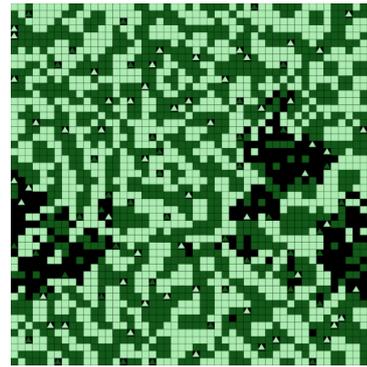


(c) $\tau=0.375$

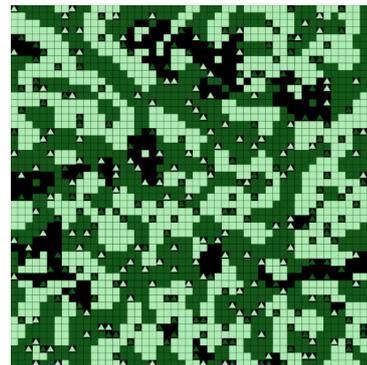


(d) $\tau=0.250$

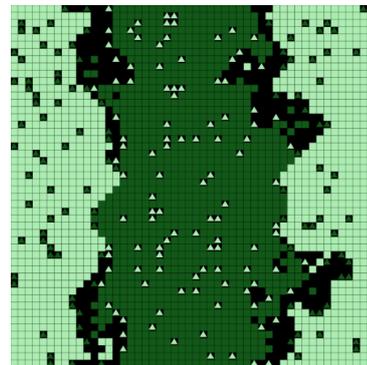
FIGURE 3 – Espace de ségrégation : premier ordre (4 exécutions significatives)
place vacante = carré noir ; marginaux=triangle



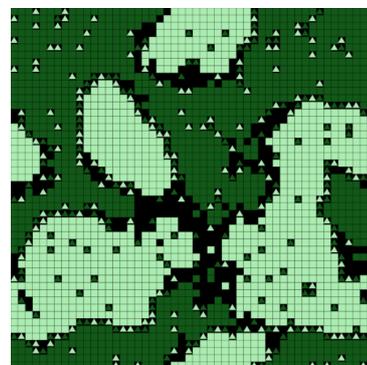
(a) $\tau=0.625; \hat{\tau}=0.250$



(b) $\tau=0.500; \hat{\tau}=0.375$



(c) $\tau=0.375; \hat{\tau}=0.375$



(d) $\tau=0.250; \hat{\tau}=0.500$

FIGURE 4 – Espace de ségrégation : second ordre (4 exécutions significatives)
place vacante = carré noir ; marginaux=triangle

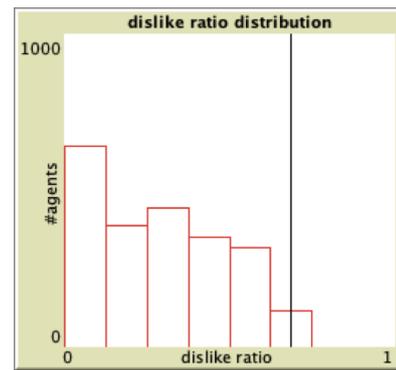
mique conduit donc à une inversion de la forme initiale ascendante de la moitié gauche de la distribution binomiale. Quelle que soit la tolérance, les décisions du second ordre réduisent significativement l'écart micro-macro. De plus, il y a un changement radical de la dynamique selon que les agents sont intolérants ou tolérants : par rapport à une dynamique du premier ordre, on observe une inversion de la pente de la distribution de décroissante à croissante. Pour les agents plutôt tolérants, tout se passe comme si la forme initiale ascendante de la moitié gauche de la distribution binomiale était conservée.

Conclusion

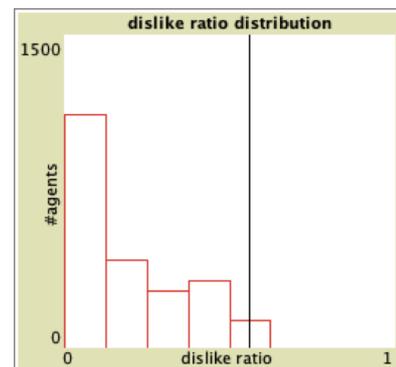
Dans le contexte de la ségrégation spatiale, nous avons proposé une nouvelle approche fondée sur un critère de décision du second ordre dans lequel la satisfaction d'un agent dépend de la satisfaction de ses voisins au regard du critère discriminatoire habituel de premier ordre. Au premier ordre les agents peuvent être perçus comme égoïstes alors qu'au second ordre ils peuvent être considérés comme altruistes ; dans cette optique, on peut dire que l'approche du second ordre met en œuvre une stratégie de contournement pour éviter que les personnes soient perçues sous un angle trop négatif.

Comme attendu, les expériences menées sur le modèle du premier ordre confirment les résultats obtenus par Schelling : principalement, la ségrégation se produit même si les agents sont plutôt tolérants et le système est globalement beaucoup plus ségrégationniste qu'il n'est localement nécessaire. Les expériences menées dans les mêmes conditions avec la discrimination du second ordre montrent que la dynamique : (i) conduit, quelle que soit la tolérance, à une satisfaction généralisée au second ordre, (ii) conduit à un espace de ségrégation où la très grande majorité des agents est également satisfaite au premier ordre, (iii) conduit à deux structures spatiales différentes pour la population, selon que les agents sont tolérants ou non, (iv) atténue à la fois l'écart micro-macro et le niveau de ségrégation. La stratégie de second ordre semble donc appropriée pour faire émerger une population où la grande majorité des agents sont satisfaits selon un critère discriminatoire qui évite de prendre une décision qui pourrait être discutable sur le plan éthique.

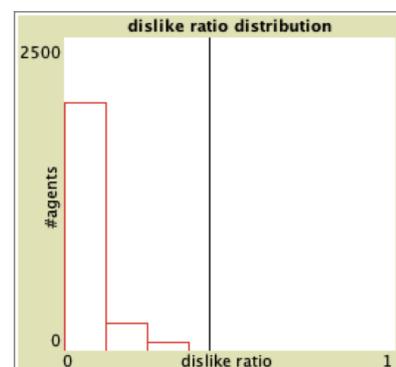
Sur la base de ces premiers résultats, nous proposons des lignes d'action possible pour de futures travaux : (i) sachant que dans la vie réelle il y a beaucoup d'hétérogénéité individuelle et que l'interaction entre altruistes et égoïstes est vitale pour la coopération, nous projetons de prendre en compte un mélange de motivations, (ii) sachant qu'il existe diverses définitions du concept de second ordre, nous projetons d'utiliser un modèle plus simple, comme un automate cellulaire 1D, pour étudier plus en profondeur ce concept, (iii) sachant que, plus que jamais, notamment avec l'émergence des réseaux sociaux, la préservation de la réputation est un défi majeur pour les individus, nous envisageons d'approfondir le lien entre recherche de réputation et approche du second ordre, (iv) nous projetons d'étendre



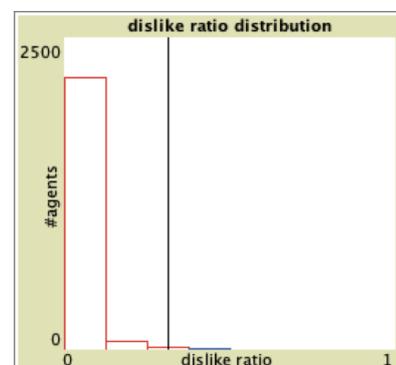
(a) $\tau=0.625$



(b) $\tau=0.500$



(c) $\tau=0.375$



(d) $\tau=0.250$

FIGURE 5 – Distribution de la dissemblance à la convergence : premier ordre (4 exécutions significatives).

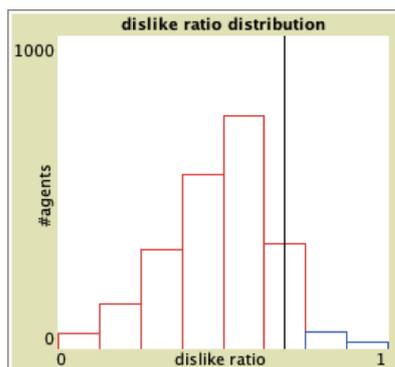
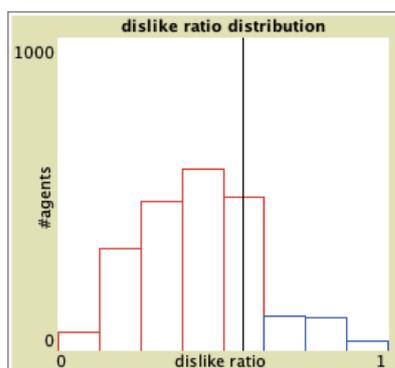
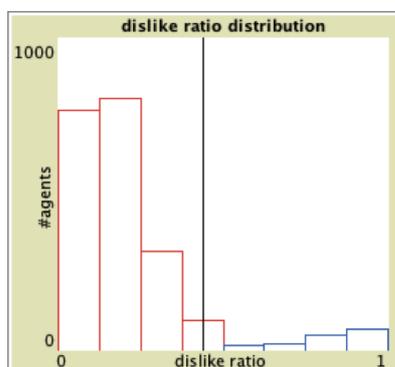
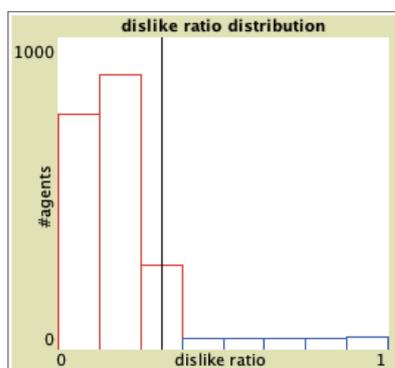
(a) $\tau=0.625$; $\hat{\tau}=0.250$ (b) $\tau=0.500$; $\hat{\tau}=0.375$ (c) $\tau=0.375$; $\hat{\tau}=0.375$ (d) $\tau=0.250$; $\hat{\tau}=0.500$

FIGURE 6 – Distribution de la dissemblance à la convergence : second ordre (4 exécutions significatives).

le modèle vers des niveaux supérieurs de méta-perception ; actuellement, les agents sont prévenants dans le sens où ils se soucient du "malheur" de leurs voisins causé par les décisions du premier ordre. Qu'arriverait-il alors si les agents se souciaient du malheur de leurs voisins causé par des décisions du deuxième ordre ? Les résultats de la simulation convergeraient-ils lorsque le niveau de perception s'approche de l'infini ?

Références

- [1] R. Hegselmann, Thomas C. Schelling and James M. Sakoda : The Intellectual, Technical, and Social History of a Model' , Journal of Artificial Societies and Social Simulation, vol. 20 (3) 15, 2017.
- [2] J. M. Sakoda, The checkerboard model of social interaction, The Journal of Mathematical Sociology, Vol. 1 (1), pp. 119–132, 1971.
- [3] T. C. Schelling, Micromotives and macrobehavior, Ed. WW Norton and Co, N.Y (new edition) 2006.
- [4] T.C. Schelling, Dynamic Models of Segregation, Journal of Mathematical Sociology 1, pp. 143–86, 1971.
- [5] U. Wilensky, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999.
- [6] W.J. Carrington and K.R. Troske, On measuring segregation in samples with small units, Ed. Journal of Business & Economic Statistics, pp. 402-409, 1997.
- [7] R. Pancs and N. Vriend, Schelling's spatial proximity model of segregation revisited, Ed. Journal Computing in Economics and Finances, 2003
- [8] A. Banos, Exploring Network Effects in Schelling Segregation Model, S4-MODUS workshop : Multi-scale interactions between urban forms and processes, 2009.
- [9] P. Collard and S. Mesmoudi and T. Ghetiu and F. Polack, Emergence of Frontiers in networked Schelling segregationist models, Complex Systems journal, Ed. Complex Systems Publications, Vol. 22 (1), 2013. <http://www.complex-systems.com>.
- [10] P. Collard, Strong and weak spatial segregation with multilevel discrimination criteria, Complex Systems journal, Ed. Complex Systems Publications, Vol. 28 (2), 2019.
- [11] P. Collard and T. Ghetiu, Segregation Landscape : A new view on the Schelling segregation space, Complex Systems journal, Vol. 25 (3), 2016.
- [12] Edna Ullmann-Margalit, No Wanting to Know, in Reasoning Practically, ed. Oxford University Press, 1998.
- [13] F. Squazzoni, Agent-Based Computational Sociology, ed. Wiley, ISBN-13 : 978-0470711743, 2012.
- [14] N. Gilbert, Varieties of emergence, in Social Agents : Ecology, Exchange, and Evolution, Agent 2002 Conference, Ed. D. Sallach, pp. 41-56, 2002.

- [15] Werner Raub, Vincent Buskens & Marcel A. L. M. Van Assen, Micro-Macro Links and Microfoundations in Sociology, *Journal of Mathematical Sociology*, Vol. 35 (1-3), 2011.
- [16] Erika N. Carlson, Meta-accuracy and relationship quality : Weighing the costs and benefits of knowing what people really think about you, *Journal of Personality and Social Psychology*, Vol. 111(2), pp. 250–264, 2016.
- [17] F. Goffette-Nagot and P. Jensen and S. Grauwin, Dynamic models of residential segregation : Brief review, analytical resolution and study of the introduction of coordination, 2009.
- [18] W. J. Carrington and K. R. Troske, On measuring segregation in samples with small units, *Journal of Business & Economic Statistics*, pp. 402-409, 1997.
- [19] E. Jara, J. Vila, and A. Maldonado, Second-order conditioning of human causal learning, *Learning and Motivation*, Vol. 37(3), pp. 230-246, 2006.
- [20] J. Baudrillard, *Simulacra and Simulation*, University of Michigan Press, pp. 6, 1994.
- [21] Erika N. Carlson, Simine Vazire, R. M. Furr, Meta-insight : do people really know how others see them ?, *Journal of Personality and Social Psychology*, Vol. 101(4), pp. 831-46 2011.
- [22] Simine Vazire, Erika N. Carlson, Others Sometimes Know Us Better Than We Know Ourselves, *Curr. Dir. Psychol. Sci.* 20, Vol. 20 (2), pp. 104-108, 2011.
- [23] Philippe Collard, Second-order micromotives and macrobehaviour, *Journal of Computational Social Science*, DOI : 10.1007/s42001-020-00062-z, Ed. Springer Nature, 2020.
- [24] P. Collard, S. Mesmoudi, T. Ghetiu, F. Polack Emergence of Frontiers in networked Schelling segregationist models. *Complex Systems Journal*. Vol. 22 (1), 2013.

Stratégie d'apprentissage proactif endogène par systèmes multi-agents coopératifs

B. Dato, M.-P. Gleizes, F. Migeon

Université Toulouse III Paul Sabatier, IRIT

{bruno.dato, marie-pierre.gleizes, frederic.migeon}@irit.fr

Résumé

Dans cet article, nous nous plaçons dans des environnements de systèmes complexes, imprévisibles et dynamiques pour lesquels nous avons pour objectif de montrer que les systèmes multi-agents adaptatifs sont une solution pertinente à leur apprentissage. Nous utilisons une approche d'apprentissage locale inspirée du constructivisme : l'auto-apprentissage de contextes par systèmes multi-agents adaptatifs. Nous cherchons à introduire des rétroactions endogènes comme moyen d'améliorer en interne le processus d'apprentissage en détectant et en résolvant les imprécisions entre les connaissances apprises. Nous proposons pour cela 3 stratégies de résolution des imprécisions d'apprentissage qui sont : une résolution réactive, un apprentissage proactif et un auto-apprentissage. Dans cet article, nous évaluons les performances de l'apprentissage proactif. Nous montrons que l'ajout de rétroactions endogènes facilité par l'auto-observation permet d'accélérer et généraliser l'apprentissage, de choisir intelligemment les données d'apprentissage et d'augmenter les performances sur les prédictions.

Mots-clés

Apprentissage par système multi-agent, Apprentissage proactif, Auto-apprentissage, Feedbacks endogènes.

Abstract

In this paper, we place ourselves in complex, unpredictable and dynamic system environments for which we aim to show that adaptive multi-agent systems are a relevant solution to their learning. We use a local learning approach inspired by constructivism : contexts self-learning by adaptive multi-agent systems. We seek to introduce endogenous feedback as a means of internally improving the learning process by detecting and resolving inaccuracies between the learned knowledge. We propose 3 strategies for resolving learning imprecisions : reactive resolution, proactive learning and self-learning. In this article, we evaluate the performance of proactive learning. We show that the addition of endogenous feedbacks facilitated by self-observation accelerates and generalizes learning, intelligently selects learning data and increases performance on predictions.

Keywords

Multi-Agent System Learning, Proactive Learning, Self-Learning, Endogenous Feedbacks.

1 Introduction

L'environnement sociotechnique des systèmes ambiants est ouvert, hétérogène, imprévisible et dynamique [1, 2, 3]. Ouvert car les entités ou les applications qui le composent peuvent apparaître et disparaître ; hétérogène car ces entités manifestent différents comportements ; imprévisible car la connaissance de l'état du système ne garantit pas son état suivant ; et dynamique car il est en constante évolution.

Toutes ces caractéristiques doivent être prises en compte lors de la conception d'une application intelligente capable d'évoluer dans cet environnement. Cependant, un système *ad hoc* traite mal toutes ces caractéristiques des environnements sociotechniques car il est impossible de spécifier *a priori* toutes les interactions qu'il aura avec son environnement. Il doit être capable de s'adapter à son environnement en tirant les leçons de ses expériences c'est-à-dire qu'il doit être capable d'apprendre même en présence de peu d'exemples. Dans ce contexte les méthodes classiques d'apprentissage automatique sont difficilement applicables ; c'est pourquoi il existe un fort besoin de nouvelles méthodes d'apprentissage.

Il est nécessaire de disposer de systèmes interactifs qui apprennent tout au long de leur vie sans avoir de connaissances intrinsèques des tâches qu'ils auront à résoudre, on les appelle les systèmes agnostiques [4]. Pour concevoir de tels systèmes, il est essentiel de générer des connaissances par le biais de processus et d'actions purement internes au système ou en interaction avec l'environnement. Nous appelons ce type de processus : "apprentissage par feedbacks endogènes".

Nos objectifs de recherche se concentrent sur l'auto-enrichissement de fragments de connaissance et l'apprentissage proactif par des systèmes multi-agents tout en s'inspirant des méthodes d'apprentissage développemental appliquées en robotique. On parle d'auto-enrichissement lorsque le système génère lui-même de la connaissance seulement à l'aide de ce qu'il a pu apprendre précédemment. En revanche, pour l'apprentissage proactif, le système choisit lui-même des exemples d'apprentissage.

Nous commençons par décrire le cadre de nos recherches en présentant les travaux déjà réalisés dans ce domaine et en analysant leurs faiblesses. Nous détaillons ensuite notre contribution qui est l'apprentissage endogène à travers 3 stratégies d'apprentissage en fonction de la disponibilité de données d'entraînement. Nous évaluons la stratégie de l'apprentissage proactif en le dotant ou le privant de nouvelles règles d'apprentissage endogènes pour en évaluer sa pertinence. Nous finissons par présenter succinctement quelques travaux connexes et en quoi ils diffèrent de notre approche.

2 Cadre de l'étude

Dans cette partie, nous introduisons l'apprentissage par contextes inspiré de l'apprentissage constructiviste et présentons l'état des divers travaux effectués dans notre équipe dans cette thématique.

2.1 Constructivisme et robotique développementale

L'apprentissage par contextes s'inspire de l'apprentissage constructiviste qui est une théorie issue des travaux de Piaget sur le développement de l'enfant [5, 6]. Selon cette théorie, la connaissance est une construction basée sur l'observation de l'environnement du sujet et l'impact de ses actions. L'unité de base de la connaissance dans cette théorie est le schéma. Il agrège plusieurs perceptions et, dans la plupart des cas, plusieurs actions [7]. Proposé par Drescher [8] puis formalisé par Holmes [9], il a été réutilisé en le combinant à des cartes auto-organisatrices (SOM) [10, 11] et l'apprentissage basé sur des modèles [12].

Dans l'optique d'appliquer nos travaux en robotique dans un contexte développemental, nous considérons un bras robotique simplifié avec 3 degrés de liberté dans un plan 2D : $(\theta_1, \theta_2, \theta_3)^T$ (fig. 1). Pour le commander, on utilise son modèle géométrique direct *MGD* et son modèle géométrique inverse *MGI* qui sont tous deux des matrices dépendantes des caractéristiques du bras. Le modèle géométrique direct permet de calculer la position de l'outil du robot à partir des angles de chaque articulation : $MGD(\theta_1, \theta_2, \theta_3) = (x_O, y_O)$. Le modèle géométrique inverse donne tous les vecteurs d'angles possibles pour une position de l'outil désirée : $MGI(x_O, y_O) = (\theta_1, \theta_2, \theta_3)$. Notre approche d'apprentissage permettrait d'apprendre la composante du *MGI* dans une dimension en associant à chaque degré de liberté θ_i une instance du système AMOEBA avec comme perceptions la position de l'outil désirée et les angles de toutes les articulations : $\mathcal{F}_{\theta_i}(x_O, y_O, \theta_1, \theta_2, \theta_3) = \theta_i$. L'apprentissage se fait par exploration supervisée des angles possibles θ_i dont l'oracle est le modèle direct. Le modèle de chaque articulation est interconnecté par la physique du système robotique pour atteindre l'objectif global désiré à l'exploitation.

2.2 Systèmes multi-agents

L'approche SMA (Systèmes Multi-Agent) [13], et en particulier l'approche AMAS (Systèmes Multi-Agents Adaptatifs) [14] confère à un système des propriétés d'adapta-

tion pour faire face à des situations imprévues ce qui est approprié pour les systèmes d'apprentissage [15, 16]. Un AMAS, qui est un système complexe artificiel favorisant l'émergence des propriétés globales attendues, permet de faire face à la complexité du monde (non-linéarité, dynamique, informations distribuées, données bruitées et imprévisibilité). De nombreuses expérimentations ont montré de telles propriétés dans des thématiques telles que le contrôle de processus biologiques [17], le contrôle optimal des moteurs [18] ou l'apprentissage en robotique [19]. Le système AMOEBA (Agnostic MODeI Builder by self-Adaptation) [20] est basé sur l'apprentissage de contextes et met en œuvre un apprentissage agnostique supervisé en ligne capable de généraliser avec des données d'entraînement continues.

2.3 Apprentissage par contextes (AMOEBA)

L'apprentissage par contextes est un problème d'exploration d'un espace de recherche à n dimensions et de calcul du modèle de régression correspondant. Une instance du système d'apprentissage apprend une sortie appelée prédiction $\mathcal{O}' \in \mathbb{R}$ selon une fonction cachée $\mathcal{F}(\mathcal{P}_n) = \mathcal{F}(p_1, p_i, \dots, p_n) = \mathcal{O}$ avec $\mathcal{O} \in \mathbb{R}$ la prédiction désirée (ou l'oracle) et $\mathcal{P}_n = [p_1, \dots, p_n] \in \mathbb{R}^n$ le vecteur des entrées appelé les *perceptions*. Le vecteur $\mathcal{L}_n = [\mathcal{P}_n, \mathcal{O}]$, composé de *perceptions* associées à une prédiction désirée, définit une *situation d'apprentissage*, ce qui s'apparente à un *schéma* dans la théorie de Piaget. Ici, on parle en terme d'*Agent Contexte* C_n^j pour nommer le $j^{\text{ème}}$ pavé en dimension n qui représente une partie du schéma. Un *Agent Contexte* est un agent intelligent qui représente localement une partie de la fonction globale \mathcal{F} et de l'espace de recherche avec un parallélotope de dimension n associé à un modèle linéaire. Le parallélotope est défini par des plages de validité $\mathcal{R}_n^j = [r_1^j, r_i^j, \dots, r_n^j]$ avec $r_i^j = [r_{i,start}^j, r_{i,end}^j]$ un intervalle de validité sur une perception p_i (fig. 2). À chaque *Agent Contexte* C_n^j est associé un modèle de régression linéaire local f_n^j . Pour chaque vecteur de *perceptions* \mathcal{P}_n dans les plages de validité, un *Agent Contexte* fournit une prédiction locale $o^j \in \mathbb{R}$ calculée à partir des coefficients de son modèle $[a_0^j, a_1^j, \dots, a_n^j] \in \mathbb{R}^{n+1}$ et des *perceptions* \mathcal{P}_n selon l'équation suivante : $o^j = f_n^j(p_1, p_i, \dots, p_n) = \sum_{i=1}^n a_i^j \cdot p_i + a_0^j$. Les *Agents Contextes* possèdent une confiance $c^j \in \mathbb{Z}$ afin de s'auto-évaluer par rapport aux autres. Un *Agent Contexte* est donc défini par ses plages de validité, son modèle et sa confiance $C_n^j = \{\mathcal{R}_n^j, f_n^j, c^j\}$.

2.4 Règles d'apprentissage exogènes

L'apprentissage à l'aide d'*Agents Contextes* se fait à partir de plusieurs règles simples [20]. Si les *perceptions* \mathcal{P}_n appartiennent aux plages de validité d'un *Agent Contexte*, celui-ci propose une prédiction avec son modèle. Dans les cas où plusieurs *Agents Contextes* sont valides, la prédiction de celui avec la meilleure auto-évaluation est retenue. Pour savoir si la prédiction d'un *Agent Contexte* est bonne ou mauvaise, on utilise une marge d'erreur et une marge d'inexactitude qui sont définies par l'utilisateur

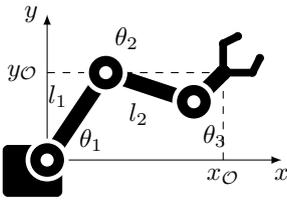


FIGURE 1 – Exemple de bras robotique simple

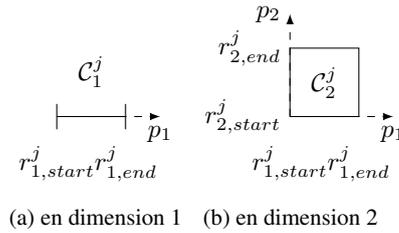


FIGURE 2 – Plage de validité de parallélotopes

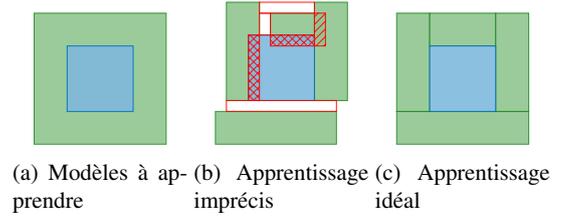


FIGURE 3 – Problème d'apprentissage simple de 2 modèles linéaires symbolisés par les couleurs vertes et bleues

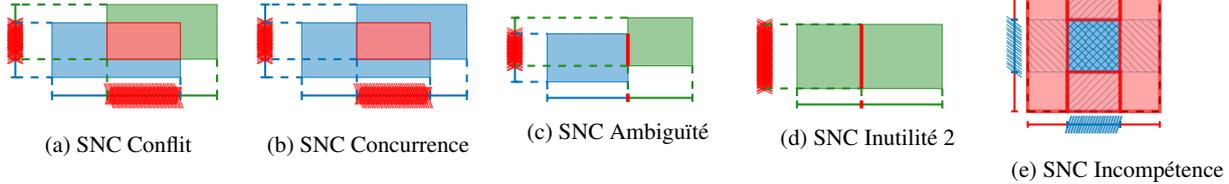


FIGURE 4 – Imprécisions 2D avec projections 1D

du mécanisme d'apprentissage. Il existe des situations anormales ou non optimales qui nécessitent des traitements particuliers que nous appelons SNC (Situations de Non Coopération). Quatre d'entre elles sont décrites ci-dessous.

SNC Prédiction Mauvaise. L'erreur entre la prédiction d'un *Agent Contexte* o^j et celle de l'oracle O est supérieure à la marge d'erreur. L'*Agent Contexte* ne devrait pas être valide pour ces *perceptions*; il modifie ses plages de validité afin de les exclure et il décrémente sa confiance c^j .

SNC Prédiction Inexacte. L'erreur entre la prédiction d'un *Agent Contexte* o^j et celle de l'oracle O est supérieure à la marge d'inexactitude. L'*Agent Contexte* met à jour son modèle avec la *situation d'apprentissage* courante $\mathcal{L}_n = [\mathcal{P}_n, \mathcal{O}]$ et il décrémente sa confiance. Si l'erreur est inférieure à la marge d'inexactitude, il incrémente sa confiance.

SNC Inutilité 1. Dans une des dimensions, la plage de validité d'un *Agent Contexte* est inférieure à une taille critique. L'*Agent Contexte* est considéré inutile et se suicide.

SNC Improductivité. Aucun *Agent Contexte* ne recouvre les *perceptions* \mathcal{P}_n réalisées. L'*Agent Contexte* le plus proche qui donne une prédiction correcte par rapport à la marge d'inexactitude étend ses plages vers le point de *perceptions* ciblé. S'il n'arrive pas à inclure le point ou qu'il n'existe pas de tel *Agent Contexte*, un nouveau est créé.

Sur un problème simple, avec deux modèles linéaires (fig. 3a), les règles d'apprentissage précédentes ne permettent pas au système de converger vers une représentation idéale (fig. 3c), c'est-à-dire avec le minimum d'*Agents Contextes*, sans aucun recouvrement ni aucun vide. Ces règles n'empêchent pas les imprécisions d'exploration (fig. 3b). Les incompétences (zones blanches ou vides) ne sont pas détectées et ne sont en partie comblées que lorsque le système d'apprentissage les rencontre. Les conflits et les concurren-

ces (chevauchements des plages de validité en hachuré) ne sont pas pris en compte pour modifier les plages des *Agents Contextes*. Il manque un mécanisme utilisant les propriétés du voisinage : pour chaque SNC, toutes les résolutions sont faites indépendamment des *Agents Contextes* voisins.

3 Apprentissage Endogène

Cette section présente l'apprentissage par feedbacks endogènes et les mécanismes sous-jacents permettant de palier les manques d'AMOEBAs.

On définit l'apprentissage endogène comme tout apprentissage qui est déclenché à partir de motivations internes au système ou entièrement générées sans intervention extérieure. Nous présentons la mise en place d'un voisinage, la détection et la résolution à n dimensions des imprécisions de l'apprentissage. Nous proposons trois stratégies de résolutions différentes en fonction des exemples d'apprentissages disponibles : réactive, proactive et auto-apprenante. Le processus d'apprentissage endogène présenté est opérationnel quel que soit le nombre de dimensions. Toutefois, pour simplifier les illustrations, seulement deux dimensions sont représentées.

3.1 Voisinage

Pour ajouter des interactions locales entre les *Agents Contextes*, à chaque \mathcal{P}_n est associé l'ensemble \mathcal{N}_C des *Agents Contextes* voisins. Un *Agent Contexte* est considéré comme un voisin des *perceptions* si ses plages de validité recoupent une zone de voisinage entourant les *perceptions* courantes. La taille de cette zone est choisie par l'utilisateur du mécanisme d'apprentissage avec le paramètre *precision_plage* et est proportionnelle à la taille de l'espace exploré dans chaque dimension. Pour une perception p_i , on a le rayon de voisinage r_i^V suivant : $r_i^V = (p_i^{max} - p_i^{min}) * 2 * precision_plage$ avec p_i^{max}

et p_i^{min} les perceptions maximale et minimale expérimentées par le mécanisme d'apprentissage suivant la perception p_i . Les *Agents Contextes* ont également une zone d'influence qui est la moitié de leur rayon pour chaque perception p_i . Si les *perceptions* actuelles se trouve dans cette zone d'influence, l'*Agent Contexte* se trouve également dans le voisinage. De ce voisinage découlent trois autres paramètres du système qui sont : le rayon de création d'un *Agent Contexte* $r_i^{creation} = 0.5 * r_i^y$, la distance de précision de l'apprentissage pour la fusion d'*Agents Contextes* $d_i^{ambiguite} = 0.25 * r_i^y$ et une erreur d'approximation des distances $d_i^{min} = 0.125 * r_i^y$ 3.2.

3.2 Imprécisions

On définit 5 types d'imprécisions de l'apprentissage qui, du point de vue des Systèmes Multi-Agents Adaptatifs, sont des Situations de Non Coopération (SNC) :

SNC Conflit. Chevauchement entre deux *Agents Contextes* qui ont des modèles différents et donnent des prédictions différentes. Il y a chevauchement si les longueurs des intersections entre les plages de validité des agents sont supérieures à d_i^{min} .

SNC Concurrence. Chevauchement entre deux *Agents Contextes* qui donnent des prédictions similaires.

SNC Incompétence. Vide entre les *Agents Contextes* dans un voisinage. Pour que ce vide soit retenu, la longueur des plages qui le définissent doit être supérieure à d_i^{min} .

SNC Ambiguïté. Différence de modèle entre deux *Agents Contextes* adjacents. Pour que 2 agents soient adjacents suivant la perception p_i , la distance entre leurs frontières suivant p_i doit être inférieure à d_i^{min} .

SNC Inutilité 2. Deux *Agents Contextes* adjacents qui donnent des prédictions similaires et qui peuvent fusionner leur frontière commune sans altérer leurs autres plages de validité. Dans ce cas là, 2 agents sont adjacents si la distance entre leurs frontières est inférieure à $d_i^{ambiguite}$.

SNC Modèle. Il existe une autre imprécision qui concerne seulement le modèle linéaire des agents et non les plages de validité. À n dimensions, pour estimer les $n + 1$ coefficients du modèle linéaire d'un *Agent Contexte* il faut $n + 1$ *situations d'apprentissage*. Un *Agent Contexte* déclenche alors une **SNC Modèle** à chaque fois qu'il n'a pas rencontré assez de *situations d'apprentissage* pour finaliser son modèle.

Hormis la SNC Inutilité 2 qui déclenche une fusion instantanée à chaque fois qu'elle est détectée, la résolution des autres SNC dépend des types d'apprentissage présentés section 3.4.

Pour un problème d'apprentissage à n dimensions, un chevauchement entre deux *Agents Contextes* (que ce soit une SNC Conflit ou une SNC Concurrence) est défini par n intersections non vides de plages de validité (figures 4a et 4b). Une SNC Ambiguïté idéale est définie par $n - 1$ intersections non vides et une intersection nulle (fig. 4c). Une SNC Inutilité 2 idéale est définie par $n - 1$ plages égales et une intersection nulle. Dans un espace continu, il est presque impossible d'avoir ces conditions pour les SNC Ambiguïté et Inutilité 2, c'est pour cela que l'on utilise

les paramètres d_i^{min} et $d_i^{ambiguite}$. Une SNC Incompétence permet de détecter les zones vides au sein d'un voisinage. Une zone est vide si elle a une intersection vide avec la plage de validité d'un *Agent Contexte* sur au moins une dimension. On construit le minimum de zones vides possibles. À 2 dimensions et avec un seul *Agent Contexte* au milieu du voisinage, le nombre minimal de zones vides à combler est 4 (fig. 4e).

Une fois détectées, les imprécisions (autres que les SNC Modèle et Inutilité 2) sont associées à une certaine priorité pour être traitées. L'ordre des priorités de la plus grande à la plus basse est : SNC Conflit, SNC Concurrence, SNC Incompétence et SNC Ambiguïté. L'imprécision la plus critique est le conflit car en plus des mauvaises plages de validité, elle peut entraîner une mauvaise prédiction. Ensuite, on trouve la concurrence dont la manifestation est la même mais sans risque de mauvaise prédiction. Ensuite, lorsqu'il n'y a pas de recouvrement, la priorité est de combler les vides. Enfin, la dernière amélioration possible pour une exploration complète est d'affiner les frontières. Chaque imprécision de l'apprentissage \mathcal{I}_n^k est définie par $\{\mathcal{R}_n^k, \mathcal{P}_n^k, \mathcal{M}^k, \mathcal{T}^k, p\}$. avec : **Identification** : k .

Plages : \mathcal{R}_n^k , elle définissent la zone de recouvrement ou de vide.

Position : \mathcal{P}_n^k , le vecteur qui correspond aux centres de toutes les plages, c'est le vecteur utilisé pour auto-générer une *situation d'apprentissage*.

Mémoire : \mathcal{M}^k , la liste des *Agents Contextes* qui détectent l'imprécision.

Type : \mathcal{T}^k ; conflit, concurrence, incompétence ou ambiguïté.

Priorité : p , elle détermine l'urgence dans laquelle doit être traitée l'imprécision.

Interroger tous les *Agents Contextes* pour chercher des imprécisions est trop coûteux en calcul si le nombre d'agents est important, la détection est donc seulement réalisée parmi l'ensemble des *Agents Contextes* voisins des *perceptions* considérées.

3.3 Règles d'apprentissage endogènes

La prise en compte des imprécisions dans l'apprentissage nécessite la modification des règles présentées section 2.4. La marge d'incertitude est supprimée et il existe une marge d'erreur dynamique pour chaque *Agent Contexte*. Cette erreur permet à l'agent d'auto-évaluer la précision de son modèle. Lors de *situations d'apprentissage* au sein de sa zone de validité, plus ses prédictions sont bonnes, plus il diminue sa marge d'erreur, plus ses prédictions sont mauvaises plus il l'augmente. On définit un marge d'erreur locale comme la moyenne des marges d'erreurs dynamiques des *Agents Contexte* voisins des *perceptions*.

SNC Prédiction Mauvaise. L'erreur entre la prédiction d'un *Agent Contexte* o^j et la valeur de l'oracle O est supérieure à la marge d'erreur locale. L'*Agent Contexte* n'est pas valide dans cette *situation d'apprentissage* car sa prédiction n'est pas assez précise compte tenu de la précision du voisinage. Il modifie une de ses plages en utilisant son

voisinage pour détruire les chevauchements en priorité et il décrémente sa confiance. On choisit toujours la plage qui nuit le moins à l'intégrité de l'*Agent Contexte*. Si l'erreur est inférieure à la marge d'erreur locale, la confiance de l'*Agent Contexte* augmente et il met à jour son modèle avec la *situation d'apprentissage* actuelle. L'ancienne SNC Prédiction Inexacte est fusionnée avec celle-ci afin de n'avoir qu'une seule marge d'erreur pour définir si un *Agent Contexte* est bon ou mauvais. Il met à jour le modèle régulièrement pour être robuste au bruit.

SNC Inutilité 1. Une des plages de validité d'un *Agent Contexte* est inférieure à la distance d_i^{min} . L'*Agent Contexte* est considéré inutile et se suicide. Il disparaît et ne représente plus une partie de l'espace à explorer.

SNC Improductivité. Aucun des *Agents Contextes* du système n'est valide pour les *perceptions* courantes. L'*Agent Contexte* le plus proche parmi les voisins qui donnent une proposition correcte étend ses plages vers celles-ci. Si elles ne sont pas incluses après l'extension, un nouvel *Agent Contexte* est créé. S'il y a des agents voisins, ils sont utilisés pour initialiser les propriétés du nouvel agent. Le modèle de l'agent est initialisé avec celui du voisin valide le plus proche. Les plages de validité sont initialisées avec la plage de validité la plus petite parmi les voisins afin d'éviter les recouvrements si la zone d'exploration est critique. La marge d'erreur dynamique est initialisée avec la marge d'erreur locale. S'il n'y a pas de voisins, l'agent créé utilise des valeurs d'initialisation basées sur les limites de perception de l'espace de recherche et sur les paramètres choisis par l'utilisateur. Il existe une limite à partir de laquelle une plage ne peut plus s'étendre. La seule façon pour un *Agent Contexte* de croître est alors de fusionner avec un autre *Agent Contexte* ce qui double sa confiance.

3.4 Stratégies de résolution

Nous proposons trois stratégies de résolution des imprécisions qui font appel ou non à de nouveaux exemples d'apprentissage. La résolution de chaque SNC d'imprécision est faite en fonction de sa priorité pour l'apprentissage actif et l'auto-apprentissage.

3.4.1 Exploitation réactive

La stratégie réactive est utilisée lorsque le mécanisme d'apprentissage ne peut pas consulter l'oracle : il est en phase d'exploitation. Elle propose une prédiction coopérative en cas d'imprécision des plages de validité. La prédiction est donnée en utilisant les *Agents Contextes* valides et les voisins des *perceptions*. La confiance d'un *Agent Contexte* est sa confiance classique c^j . Pour cette stratégie seulement les SNC Conflit, Concurrence et Incompétence sont traitées.

SNC Conflit et Concurrence Plusieurs *Agents Contextes* sont valables et donnent des prédictions contradictoires ou concurrentes. Celui dont la confiance est la plus élevée est choisi pour la prédiction.

SNC Incompétence Aucun *Agent Contexte* n'est valide. S'il y a des voisins, une combinaison linéaire de la prédiction des voisins pondérée par leur influence est donnée. L'influence d'un *Agent Contexte* est calculée par une gaus-

sienne à n dimensions centrée sur celui-ci, avec une dispersion proportionnelle à ses plages de validité et pondérée par sa confiance c^j . S'il n'y pas de voisins, la prédiction est donnée par l'*Agent Contexte* le plus proche.

Dans cette stratégie, aucun modèle et aucune plage de validité ne sont modifiés.

3.4.2 Apprentissage proactif

La stratégie d'apprentissage proactif est possible lorsque l'on peut demander à un oracle un exemple d'apprentissage précis. Les imprécisions sont détectées au sein du voisinage de manière endogène et engendrent des *situations d'apprentissage* qui sont demandées à l'oracle en fonction de leur priorité. Lorsque le système apprend, la confiance d'apprentissage $c_l^j = c^j/d_{\mathcal{P}_n}^j$ est utilisée à partir de la confiance classique c^j et la distance $d_{\mathcal{L}_n}^j$ entre la *situation d'apprentissage* actuelle \mathcal{L}_n et le modèle f_n^j de l'*Agent Contexte* C^j . C'est un calcul de distance entre un point à $n + 1$ dimensions qui est la situation $\mathcal{L}_n = [\mathcal{P}_n, \mathcal{O}]$, et l'hyperplan \mathcal{H} de dimension $n + 1$ représentant le modèle linéaire f_n^j . La dimension $n + 1$ est celle de la prédiction.

$$d_{\mathcal{P}_n}^j = \frac{|\mathcal{L}_n \vec{\mathcal{H}} \cdot \vec{n}|}{\|\vec{n}\|} = \frac{|\mathcal{O} \cdot 1 - a_1^j \cdot p_1 - \dots - a_n^j \cdot p_n - a_0|}{\|1^2 + a_1^{j2} + \dots + a_n^{j2}\|}$$

avec $o \cdot 1 - a_1^j \cdot x_1 - \dots - a_n^j \cdot x_n - a_0 = 0$ l'équation cartésienne du plan et $\vec{n} = [1, -a_1^j, \dots, -a_n^j]$ le vecteur normal à l'hyperplan \mathcal{H} , \mathcal{O} la prédiction de l'oracle, $[a_0^j, a_1^j, \dots, a_n^j]$ les coefficients du modèle et $[p_1, \dots, p_n]$ les *perceptions* \mathcal{P}_n . Les imprécisions sont détectées par les *Agents Contextes* dans le voisinage. Une fois détectées, les demandes sont envoyées à l'oracle. Pour cette stratégie toutes les SNC d'imprécision présentées section 3.2 peuvent être traitées.

SNC Modèle Tant qu'un *Agent Contexte* a besoin de *situations d'apprentissage* supplémentaires pour finaliser son modèle, il demande de nouvelles *situations d'apprentissage* situées à des positions aléatoires dans ses plages de validité.

SNC Conflit et concurrence Des conflits ou des concurrences sont détectées dans le voisinage d'une *situation d'apprentissage*. De nouvelles *situations d'apprentissage* sont demandées aux positions de ces imprécisions. Quand elles arrivent, plusieurs *Agents Contextes* sont valides et donnent des prédictions contradictoires ou concurrentes. L'*Agent Contexte* le plus pertinent dans ce cas là est celui qui présente la plus grande confiance c_l^j . Les autres *Agents Contextes* réduisent une de leurs plages pour exclure les *perceptions* actuelles et détruire le chevauchement tout en minimisant la perte de volume. Si la *situation d'apprentissage* actuelle est suffisamment proche du modèle de l'*Agent Contexte* choisi, il met à jour son modèle avec celle-ci. Sinon, c'est une SNC Prédiction Mauvaise.

SNC Incompétence Des zones de vide sont détectées dans le voisinage des *perceptions* courantes. De nouvelles *situations d'apprentissage* sont générées pour chacun des vides

détectés avec leurs plages. Quand ces situations se présentent, c'est la SNC Improductivité qui est détectée mais elle utilise les plages définies par la SNC Incompétence pour combler la zone incompétente.

SNC Ambiguïté Une adjacence entre deux *Agents Contextes* est détectée. Deux *situations d'apprentissage* sont générées à une distance de d_i^{min} des bordures de chacun des agent pour vérifier qu'elles sont bien placées.

3.4.3 Auto-apprentissage

Cette stratégie est utilisée lorsque l'on veut minimiser le nombre d'exemples demandés à l'oracle ou que l'on ne peut pas demander d'exemples supplémentaires spécifiques comme dans la stratégie précédente. La résolution des situations d'imprécision se fait à l'aide du voisinage. Pour chaque *situation d'apprentissage* donnée par l'oracle, les imprécisions sont détectées dans le voisinage et elles sont résolues immédiatement sous forme de *requête d'exploitation*, c'est-à-dire des *situations d'apprentissage* sans oracle, autrement dit, des *perceptions*. Pour cette stratégie seulement les SNC Conflit, Concurrence et Incompétence sont traitées.

SNC Conflit et concurrence Des conflits ou des concurrences sont détectées dans le voisinage des *perceptions*. Des *requêtes d'exploitation* sont générées aux positions de ces imprécisions. Pour chaque requête, plusieurs *Agents Contextes* sont valides et donnent des prédictions contradictoires ou concurrentes. L'*Agent Contexte* le plus pertinent dans ce cas là est celui qui présente la plus grande confiance c^j . Les autres *Agents Contextes* réduisent une de leurs plages pour exclure les *perceptions* actuelles et détruire le chevauchement tout en minimisant la perte de volume.

SNC Incompétence Des zones de vide sont détectées dans le voisinage des *perceptions* courantes. Des nouvelles requêtes sont générées pour tous les vides détectés avec leurs plages. Pour chacune de ces requête, s'il y a des voisins, un nouvel *Agent Contexte* est créé et initialisé en utilisant les propriétés de ses voisins comme pour la SNC Improductivité. Pour la prédiction à apprendre par le nouvel agent, on utilise une combinaison linéaire pondérée par l'influence des voisins (cf. SNC Incompétence 3.4.1). S'il n'y a pas de voisins, il n'y a pas de création d'un nouvel agent.

4 Expérimentations

Dans cette section, nous évaluons les performances de l'apprentissage au regard de différentes métriques. Nous présentons une partie des résultats réalisés uniquement avec la stratégie d'apprentissage proactif. Pour cela, nous comparons l'apprentissage sans aucune détection et résolution de SNC, avec les résultats prenant en compte une seule SNC à la fois, puis avec toutes les SNC réunies. Chaque expérience d'apprentissage est arrêtée à 1000 cycles. Le mécanisme d'apprentissage reçoit une *situation d'apprentissage* à chaque cycle. Les métriques sont moyennées sur 10 expériences d'apprentissage. Les modèles à apprendre sont deux modèles linéaires dont la répartition dans l'espace est

celle de la fig. 3a. En moyenne, la durée d'une expérience est d'environ 3 secondes pour les caractéristiques¹ de la machine que nous utilisons.

4.1 Métriques

Nous présentons ici les métriques utilisées sur l'exploration de l'espace de recherche, sur les données d'entraînement et sur les prédictions. Pour simplifier, nous supprimons les indices n qui symbolisent la dimension de l'espace.

4.1.1 Généralisation

Pour évaluer la capacité du mécanisme à généraliser, nous regardons le nombre d'*Agents Contextes* n_{Ctxt} utilisés pour représenter l'espace.

4.1.2 Exploration de l'espace de recherche

On s'intéresse à différents types de volumes qui mesurent l'exploration de l'espace de recherche.

- \mathcal{V}_{total} : le volume total de l'espace de recherche calculé à partir des perceptions minimales p_i^{min} et maximales p_i^{max} expérimentées par le mécanisme d'apprentissage :

$$\mathcal{V}_{total} = \prod_i (p_i^{max} - p_i^{min})$$

- \mathcal{V}_{flt} : le volume de conflits, somme normalisée des volumes explorés par deux *Agents Contextes* avec des modèles différents :

$$\mathcal{V}_{flt} = (\sum_{(j,k), 1 \leq j < k \leq n_{Ctxt}, f^j \neq f^k} \mathcal{V}_{C^j \cap C^k}) / \mathcal{V}_{total}$$

- \mathcal{V}_{conc} : le volume de concurrences, somme normalisée des volumes explorés par deux *Agents Contextes* avec des modèles similaires. Pour savoir si deux modèles sont similaires, nous comparons la somme des différences des coefficients à un seuil. Pour les expérimentations que nous présentons, ce seuil est défini à partir des modèles à apprendre.

$$\mathcal{V}_{conc} = (\sum_{(j,k), 1 \leq j < k \leq n_{Ctxt}, f^j \approx f^k} \mathcal{V}_{C^j \cap C^k}) / \mathcal{V}_{total}$$

- \mathcal{V}_{Ctxt} : le volume exploré par l'ensemble des *Agents Contextes*, somme normalisée des volumes de chaque *Agent Contexte* \mathcal{V}_{C^j} à laquelle on soustrait les volumes de tous les recouvrements entre *Agents Contextes*. $\mathcal{V}_P = (\sum_j \mathcal{V}_{C^j} - \mathcal{V}_{flt} - \mathcal{V}_{conc}) / \mathcal{V}_{total}$

- \mathcal{V}_{Inc} : le volume normalisé de l'espace de recherche non exploré. $\mathcal{V}_{Inc} = 1 - \mathcal{V}_{Ctxt}$

- \mathcal{V}_{Imp} : le volume des imprécisions : somme des volumes des imprécisions normalisés (conflits, concurrences et incompétences). $\mathcal{V}_{Imp} = \mathcal{V}_{flt} + \mathcal{V}_{conc} + \mathcal{V}_{Vides}$

4.1.3 Données d'apprentissage

Pour les données d'apprentissage, nous nous intéressons au nombre de *situations d'apprentissage* \mathcal{L} fournies aléatoirement au mécanisme d'apprentissage et au nombre de celles demandées par le système.

- \mathcal{L}_{Rdm} : *Situations d'apprentissage* aléatoires.
- \mathcal{L}_A : *Situations d'apprentissage* actives, c'est la somme de toutes les suivantes.
- \mathcal{L}_A^f : *Situations d'apprentissage* pour finaliser un modèle de régression incomplet.

1. Processeur Intel(R) Core(TM) i7-6600U CPU 2.20 GHz 2.81 GHz, RAM 32 Go

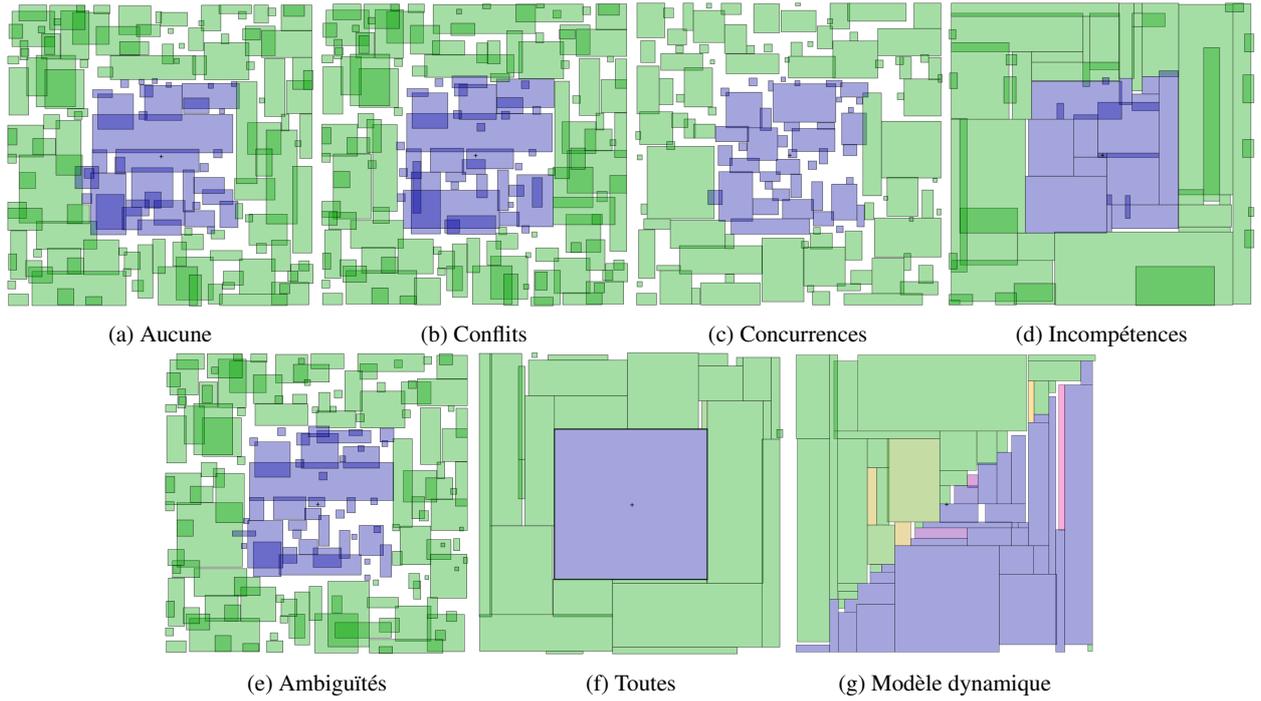


FIGURE 5 – Exemples d'apprentissage après 1000 cycles avec détection et résolution des SNC

	Aucune SNC	Conflits	Concurrences	Incompétences	Ambiguïtés	Toutes les SNC	Dynamique
\mathcal{V}_{Ctxt} (%)	80.23 ± 1.38	78.4 ± 1.11	73.36 ± 1.96	97.97 ± 0.42	75.71 ± 0.98	97.23 ± 0.58	92.21 ± 4.77
\mathcal{V}_{Imp} (%)	42.28 ± 1.44	43.39 ± 1.65	28.26 ± 1.94	12.7 ± 2.61	42.94 ± 0.92	3.67 ± 0.57	12.92 ± 5.53
\mathcal{V}_{Cflt} (%)	0.7 ± 0.27	0.12 ± 0.08	0.24 ± 0.15	0.82 ± 0.56	0.19 ± 0.18	0.13 ± 0.12	4.9 ± 3.59
\mathcal{V}_{Cnc} (%)	21.81 ± 2.24	21.67 ± 1.89	1.38 ± 0.2	9.85 ± 2.59	18.46 ± 0.91	0.76 ± 0.4	0.23 ± 0.22
\mathcal{V}_{Inc} (%)	19.77 ± 1.38	21.6 ± 1.11	26.64 ± 1.96	2.03 ± 0.42	24.29 ± 0.98	2.77 ± 0.58	7.79 ± 4.77
\mathcal{L}_{Rdm} (#)	923 ± 4	908 ± 4	786 ± 6	581 ± 45	792 ± 17	220 ± 55	127 ± 37
\mathcal{L}_A (#)	77 ± 4	93 ± 4	214 ± 6	419 ± 45	208 ± 17	780 ± 55	874 ± 37
\mathcal{L}_A^f (#)	77 ± 4	78 ± 3	78 ± 3	41 ± 9	77 ± 4	43 ± 8	8 ± 4
\mathcal{L}_A^{Cflt} (#)	0 ± 0	15 ± 3	0 ± 0	0 ± 0	0 ± 0	15 ± 6	26 ± 9
\mathcal{L}_A^{Cnc} (#)	0 ± 0	0 ± 0	136 ± 4	0 ± 0	0 ± 0	71 ± 11	58 ± 15
\mathcal{L}_A^{Inc} (#)	0 ± 0	0 ± 0	0 ± 0	378 ± 50	0 ± 0	415 ± 38	185 ± 31
\mathcal{L}_A^{Amb} (#)	0 ± 0	0 ± 0	0 ± 0	0 ± 0	132 ± 17	237 ± 34	598 ± 68
n_{Ctxt} (#)	159 ± 7	165 ± 6	125 ± 10	53 ± 6	159 ± 7	31 ± 5	59 ± 10
\mathcal{E}_O (%)	55.4 ± 28.85	51.29 ± 19.51	44.04 ± 19.07	2.68 ± 1.67	53.51 ± 18.82	1.81 ± 3.03	3.02 ± 3.99

TABLE 1 – Métriques d'exploration, de données d'apprentissage, de généralisation et de prédiction moyennées sur 10 épisodes de 1000 cycles d'apprentissage pour différentes résolutions de SNC.

- \mathcal{L}_A^{Cflt} : Situations d'apprentissage pour résoudre des conflits après leur détection.
- \mathcal{L}_A^{Cnc} : Situations d'apprentissage pour résoudre des concurrences après leur détection.
- \mathcal{L}_A^{Inc} : Situations d'apprentissage pour combler des incompétences après leur détection.
- \mathcal{L}_A^{Amb} : Situations d'apprentissage pour affiner des ambiguïtés après leur détection.

4.1.4 Prédiction

Pour évaluer les propositions du mécanisme d'apprentissage, on calcule l'erreur de prédiction normalisée \mathcal{E}_O entre les valeurs données par l'oracle et le système : $\mathcal{E}_O = |\mathcal{O} - \mathcal{O}'|/|\mathcal{O}|$. La métrique de la prédiction est calculée sur

250 cycles supplémentaires pour lesquels on demande au mécanisme des prédictions sur des *requêtes d'exploitation* aléatoires dans l'espace de recherche. Pendant ces cycles, le système n'apprend pas.

4.2 SNC d'exploration

Dans cette section, nous présentons les résultats obtenus en comparant les résolutions des différentes SNC d'imprécision de l'exploration. Les valeurs de métriques que nous donnons ici sont les moyennes disponibles sur la table 1.

4.2.1 Exploration sans traitement de SNC

Lorsqu'aucune SNC n'est prise en compte, on remarque visuellement (fig. 5a) que le nombre d'imprécisions d'exploration est important. Les résultats chiffrés le confirment :

42.28% et les *Agents Contextes* ne recouvrent que 80.23% de l'espace. On retrouve surtout des concurrences (21.81%) et des incompétences (19.77%). Sur les 1000 cycles, seulement 77 *situations d'apprentissage* sont demandées par le mécanisme pour finaliser des modèles. On compte un total de 159 *Agents Contextes* utilisés et il y a une erreur de 55.4% sur les valeurs des prédictions fournies.

4.2.2 Exploration avec traitement de SNC seules

Pour chacun des cas suivants, nous comparons les valeurs des métriques obtenues avec le cas précédent qui sert de référence.

Conflits Le traitement des conflits n'apporte pas de différence visuelle flagrante (intersections bleu/vert fig. 5b). Il baisse de 82.86% le volume de conflits dans l'exploration. On remarque que seulement 15 exemples d'apprentissage sont demandés pour résoudre des conflits. Le nombre d'agents et l'erreur de prédiction sont similaires à la section précédente 4.2.1.

Concurrences Le traitement des concurrences apporte une exploration presque sans recouvrement (intersections bleu/bleu et vert/vert fig. 5c). Il baisse de 93.67% le volume de concurrences et de 65.71% le volume de conflits. Cependant, il augmente de 35.75% le volume de vide et il baisse de 8.56% le volume exploré. 136 *situations d'apprentissage* sont demandées par le mécanisme pour résoudre les concurrences. L'erreur de prédiction est similaire à l'erreur sans traitement des SNC mais le nombre d'agents a baissé d'environ 20%.

Incompétences Le traitement des incompétences donne une exploration avec presque aucun vide (zones blanches fig. 5d). Le volume exploré est proche de 100% (97.97%), le volume de vide baisse de 89.73% et le volume de concurrences baisse aussi de 54.84%. On remarque qu'il y a 36 *situations d'apprentissage* de SNC Modèle en moins qui sont demandées et 378 situations sont demandées pour combler des vides. Le nombre d'agents baisse de 66.67% et l'erreur de prédiction baisse de 95.16%.

Ambiguïtés Le traitement des ambiguïtés donne une exploration similaire au cas sans traitement des SNC (fig. 5e). Le volume des conflits baisse de 72.86% mais le volume de vide augmente de 22.86% et le volume exploré baisse de 5.63%. 132 *situations d'apprentissage* sont demandées pour résoudre des ambiguïtés, le nombre d'agents et l'erreur de prédiction sont similaires au cas sans traitement des SNC.

4.2.3 Exploration avec intégration de toutes les SNC

Lorsque l'on traite toutes les SNC au cours de l'apprentissage, on remarque que l'exploration fig. 5f est le résultat le plus proche de l'idéal attendu 3c. Le volume obtenu est proche de 100% (97.23%), mais il est inférieur au volume obtenu pour le cas avec les SNC Incompétences seules. Le volume de toutes les imprécisions baisse de 91.32%, c'est le plus bas de tous les cas. Pour ce qui est des *situations d'apprentissage* actives, le nombre de situations demandées pour finaliser des modèles est le même que pour le cas

du traitement des SNC Incompétences ; il y a autant de détections de conflits que pour le cas des SNC Conflit seules ; il y a 65 SNC Concurrences détectées de moins que pour le traitement seul des SNC Concurrences ; il y a 37 vides de plus de détectés que pour le traitement seul des incompétences ; il y a 105 ambiguïtés détectées de plus que pour le traitement seul des SNC Ambiguïtés. La généralisation est la meilleure avec 22 *Agents Contextes* de moins que pour le meilleur cas rencontré pour les SNC Incompétences. La prédiction est aussi la meilleure obtenue avec une erreur de 1.81%.

4.2.4 Modèle Dynamique

Nous testons les capacités du mécanisme d'apprentissage à réutiliser ses connaissances et à s'adapter à une variation de l'espace de recherche après une expérience d'apprentissage de 1000 cycles avec toutes les SNC (fig. 5f). Après 1000 cycles supplémentaires d'apprentissage, on peut voir le résultat sous une forme diagonale (fig. 5g). Table 1 : on remarque que les performances de l'exploration sont un peu moins bonnes en raison de la discontinuité plus difficile à apprendre mais la quantité de situations d'apprentissage de modèle \mathcal{L}_A^f est bien inférieure.

4.3 Discussion

D'après la table 1, on remarque que lorsque l'on ne traite aucune des imprécisions de l'exploration, de nombreuses concurrences et incompétences résiduelles représentent environ 40% de l'espace. Il y a peu de conflits, seulement 0.7% de l'espace. Seulement 7.7% des *situations d'apprentissage* sont demandées par le mécanisme pour finaliser des modèles. Le nombre d'*Agents Contextes* est loin du cas idéal de 5 agents présenté section 2.4. L'erreur de prédiction est très mauvaise, ce qui s'explique par la très grande quantité d'espace non exploitable. Le traitement des SNC Conflit permet de réduire le volume de conflits mais cela impacte peu les résultats. Le traitement des SNC Concurrence améliore nettement les concurrence d'exploitation mais il augmente le volume des vides. En effet, la diminution de certaines plages de validité d'*Agents Contextes* pour cela crée en même temps de nouvelles zones de vide. On remarque que le volume de conflits diminue aussi, le système en confond certains avec des concurrences et les résout aussi. La résolution des concurrences combinée avec la fusion des *Agents Contextes* permet aussi de réduire le nombre d'*Agents Contextes*. Le traitement des SNC Incompétence permet d'améliorer nettement les performances car le volume de vides est en grande partie supprimé. La résolution des zones d'incompétence permet aussi de réduire la moitié des concurrences. Le fait de combler la plupart des vides et de fusionner les agents qui peuvent l'être diminue encore le nombre d'agents contextes. L'erreur de prédiction est meilleure car les situations d'ambiguïté qui correspondaient aux zones d'incompétence sont dans ce cas très rares. Le traitement des SNC Ambiguïté diminue le volume de conflits présents mais augmente le volume de vides. En effet, ces situations sont détectées proches des zones de conflits et permettent ainsi de les prévenir.

Lorsque l'on traite l'ensemble des SNC, on peut voir que toutes les imprécisions d'exploration sont traitées. Pour cela, 78% des *situations d'apprentissage* sont demandées par le mécanisme. On remarque qu'en combinant les SNC, moins de concurrences sont détectées, on les prévient en remplissant les vides rencontrés. Cependant, il y a plus d'incompétences car résoudre les conflits et les concurrences génère des vides supplémentaires. On détecte aussi beaucoup plus de SNC Ambiguïté, l'apprentissage ayant d'avantage convergé vers le modèle caché, les imprécisions qui dominent sont les ambiguïtés et non les conflits, concurrences et incompétences. Le nombre d'*Agents Contextes* et l'erreur de prédiction sont les plus bas de tous les cas rencontrés mais il existe encore d'infimes imprécisions. Bien que la fusion entre agents soit possible, on peut voir fig. 5f que le nombre d'*Agents Contextes* n'atteint pas l'optimum. Pour atteindre le cas idéal de 5 agents, il manque un traitement supplémentaire qui permettrait aux *Agents Contextes* de réorganiser leurs plages de validité, c'est-à-dire se scinder si cela permet une fusion.

L'expérience dynamique montre la capacité du mécanisme d'apprentissage à adapter son exploration et à réutiliser des modèles déjà connus par les agents pour accélérer l'apprentissage.

Le cas d'apprentissage que nous présentons ici est un cas simplifié d'une part car les frontières des zones à apprendre sont facilement apprises par les *Agents Contextes* qui sont de forme parallélépipédique (hormis l'expérience dynamique). D'autre part, les modèles à apprendre sont des modèles linéaires estimés par des modèles linéaires. La suite des évaluations portera sur l'exploration d'espaces de recherche plus complexes telles que des triangles, des disques ou encore des lois gaussiennes avec des modèles à apprendre non linéaires plus proches des systèmes complexes réels.

5 Travaux connexes

L'approche intégrant un modèle local de fonction affine au sein de chaque agent *Contexte* diffère des approches classiques de régression et de classification. La régression par morceau, agentifiée et distribuée, en ligne et dynamique, permet d'ajouter des mécanismes endogènes qui guident l'apprentissage en le rendant proactif. Nous présentons dans cette section des méthodes similaires dont nous mettons en évidence les similitudes et différences.

Choix des données L'approche présentée dans cet article est comparable à de l'apprentissage actif, qui consiste à décider quelles données doivent être labellisées à partir d'un jeu de données non labellisées [21], et de l'apprentissage semi-supervisé qui cherche à obtenir de meilleures performances en combinant des données non étiquetées et étiquetées plutôt que d'utiliser des données exclusivement étiquetées ou exclusivement non étiquetées [22].

Domaine, dimension et distribution/localité L'apprentissage par transfert utilise des domaines ou des tâches connexes pour améliorer un apprentissage [23]. L'appren-

tissage non supervisé utilise seulement des données non étiquetées pour des problèmes de regroupement de données ou de réduction de la dimension [24]. L'apprentissage distribué divise les données d'entraînement en sous-ensembles pour les traiter [25]. On trouve des travaux d'apprentissage local pour la classification basée sur la généralisation, la continuité en prédiction et le passage à l'échelle [26].

Motivation intrinsèque Au-delà de l'apprentissage par renforcement inverse et de l'apprentissage par l'intention, dans les domaines de la robotique, la motivation intrinsèque [27, 28] est un moyen de guider l'exploration en se concentrant sur le processus d'apprentissage lui-même. La motivation intrinsèque est indépendante de la tâche, elle favorise l'apprentissage hiérarchique des connaissances et la réutilisation des compétences [29]. La motivation intrinsèque est une approche relativement nouvelle et un certain nombre de comportements importants et intéressants n'ont pas encore été simulés. On la trouve notamment dans le domaine la robotique développementale, aussi inspirée du constructivisme, elle se concentre sur le développement de robots dotés de capacités mentales infantiles [30]. Cette approche est destinée aux systèmes "vierges" qui apprennent "from scratch". On trouve surtout des architectures qui font appel à différents mécanismes d'apprentissage : Multilevel Darwinist Brain (MDB) se concentre sur l'apprentissage en ligne tout au long de la vie [31], l'architecture cognitive émergente de l'iCub [32] comprend les principes d'auto-organisation, de motivation intrinsèque, d'apprentissage social, de cognition incarnée et active, d'apprentissage en ligne et cumulatif.

Parmi ces méthodes d'apprentissage, nos travaux rentrent dans le cadre de l'apprentissage actif et semi-supervisé. Cependant, nos intérêts se portent ici sur l'ajout de mécanismes endogènes à notre approche d'apprentissage par contexte et non la comparaison avec d'autres méthodes d'apprentissage actif ou semi-supervisé. Nous faisons un lien avec la robotique car nos objectifs sont d'utiliser notre mécanisme d'apprentissage dans le cadre d'apprentissage de tâches pour des robots tout au long de leur vie.

6 Conclusion et Perspectives

Comme les systèmes multi-agents sont bien adaptés pour faire face aux contraintes du monde sociotechnique, nous avons proposé un nouveau paradigme d'apprentissage basé sur l'apprentissage contextuel. Dans le cadre de l'apprentissage par contextes, nous avons proposé de le combiner avec un système multi-agent pour l'apprentissage fondé sur des motivations internes pour apprendre de manière intelligente et critique.

L'auto-observation des systèmes multi-agents auto-adaptatifs permet de détecter des imprécisions dans les modèles du mécanisme d'apprentissage. Nous avons proposé des rétroactions endogènes fondées sur la détection de situations non coopératives pour accélérer l'apprentissage. Elles améliorent également les performances de prédiction en anticipant les situations ambiguës et en les résolvant.

Tous les comportements des agents coopératifs pour résoudre les SNC sont génériques car ils ne dépendent en aucune manière de l'application sous-jacente.

Nous avons présenté ces mécanismes coopératifs d'auto-organisation endogène sur des problèmes synthétiques qui nous ont permis de les valider de manière rigoureuse. Nous focalisons désormais nos travaux sur des applications concrètes en robotique pour montrer le gain en performance et précision de ces mécanismes relativement à des approches passives d'apprentissage.

Références

- [1] S. J. Russell and P. Norvig, *Artificial intelligence : a modern approach*. Malaysia ; Pearson Education Limited., 2016.
- [2] J. Nigon, N. Verstaevel, J. Boes, F. Migeon, and M.-P. Gleizes, "Smart is a matter of context," in *International and Interdisciplinary Conference on Modeling and Using Context*, pp. 189–202, Springer, 2017.
- [3] H. Abbas, S. Shaheen, M. Elhoseny, A. K. Singh, and M. Alkhambashi, "Systems thinking for developing sustainable complex smart cities based on self-regulated agent systems and fog computing," *Sustainable Computing : Informatics and Systems*, vol. 19, pp. 204–213, 2018.
- [4] M. J. Kearns, R. E. Schapire, and L. M. Sellie, "Toward efficient agnostic learning," *Machine Learning*, vol. 17, no. 2-3, pp. 115–141, 1994.
- [5] J. Piaget, *La naissance de l'intelligence chez l'enfant*, vol. 370. Delachaux et Niestlé Neufchâtel, Switzerland, 1977.
- [6] J. Piaget, "La formation du symbole chez l'enfant : imitation, jeu et rêve, image et représentation," 1978.
- [7] F. Guerin, "Learning like a baby : a survey of artificial intelligence approaches," *The Knowledge Engineering Review*, vol. 26, no. 2, pp. 209–236, 2011.
- [8] G. L. Drescher, *Made-up minds : a constructivist approach to artificial intelligence*. MIT press, 1991.
- [9] M. P. Holmes *et al.*, "Schema learning : Experience-based construction of predictive action models," in *Advances in Neural Information Processing Systems*, pp. 585–592, 2005.
- [10] H. H. Chaput, *The constructivist learning architecture : A model of cognitive development for robust autonomous robots*. PhD thesis, 2004.
- [11] J. Provost, B. J. Kuipers, and R. Miikkulainen, "Developing navigation behavior through self-organizing distinctive-state abstraction," *Connection Science*, vol. 18, no. 2, pp. 159–172, 2006.
- [12] F. S. Perotto, "A computational constructivist model as an anticipatory learning mechanism for coupled agent-environment systems," *Constructivist Foundations*, vol. 9, no. 1, pp. 46–56, 2013.
- [13] J. Ferber, *Multi-agent systems : an introduction to distributed artificial intelligence*, vol. 1. Addison-Wesley Reading, 1999.
- [14] J.-P. Georgé, M.-P. Gleizes, and V. Camps, "Cooperation," in *Self-organising Software*, pp. 193–226, Springer, 2011.
- [15] S. Mazac, F. Armetta, and S. Hassas, "Approche décentralisée pour un apprentissage constructiviste en environnement continu : application à l'intelligence ambiante.," in *Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*, CÉPADUÈS ÉDITIONS, 2015.
- [16] M. Guériaux, *Systèmes multi-agents, auto-organisation et contrôle par apprentissage constructiviste pour la modélisation et la régulation dans les systèmes coopératifs de trafic*. PhD thesis, Université de Lyon I Claude Bernard, 2016.
- [17] S. Videau, *Contrôle de processus dynamiques par systèmes multi-agents adaptatifs : application au contrôle de bioprocédés*. PhD thesis, Toulouse, INSA, 2011.
- [18] J. Boes, *Apprentissage du contrôle de systèmes complexes par l'auto-organisation coopérative d'un système multi-agent : application à la calibration de moteurs à combustion*. PhD thesis, Université Toulouse III Paul Sabatier, 2014.
- [19] N. Verstaevel, *Self-organization of robotic devices through demonstrations*. PhD thesis, Université Toulouse III Paul Sabatier, 2016.
- [20] J. Nigon, M.-P. Gleizes, and F. Migeon, "Self-adaptive model generation for ambient systems," *Procedia Computer Science*, vol. 83, pp. 675–679, 2016.
- [21] B. Settles, "Active learning literature survey," tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.
- [22] Y.-F. Li, H.-W. Zha, and Z.-H. Zhou, "Learning safe prediction for semi-supervised regression," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [23] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [24] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The elements of statistical learning*, pp. 485–585, Springer, 2009.
- [25] S.-B. Lin, X. Guo, and D.-X. Zhou, "Distributed learning with regularized least squares," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 3202–3232, 2017.
- [26] V. Zantedeschi, *A Unified View of Local Learning : Theory and Algorithms for Enhancing Linear Models*. PhD thesis, 2018.
- [27] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE transactions on evolutionary computation*, vol. 11, no. 2, pp. 265–286, 2007.
- [28] A. Bondu and V. Lemaire, "Active learning using adaptive curiosity," in *International Conference on Epigenetic Robotics : Modeling Cognitive Development in Robotic Systems*, 2007.
- [29] M. Mirolli and G. Baldassarre, "Functions and mechanisms of intrinsic motivations," in *Intrinsically Motivated Learning in Natural and Artificial Systems*, pp. 49–72, Springer, 2013.
- [30] A. Cangelosi, M. Schlesinger, and L. B. Smith, *Developmental robotics : From babies to robots*. MIT Press, 2015.
- [31] F. Bellas, R. J. Duro, A. Faiña, and D. Souto, "Multilevel darwinist brain (mdb) : Artificial evolution in a cognitive architecture for real robots," *IEEE Transactions on autonomous mental development*, vol. 2, no. 4, pp. 340–354, 2010.
- [32] D. Vernon, C. Von Hofsten, and L. Fadiga, *A roadmap for cognitive development in humanoid robots*, vol. 11. Springer Science & Business Media, 2011.

Analyse des performances d'algorithmes DCOP pour l'association d'utilisateurs de réseaux HetNets

G. Picard^a P. Rust^b
 picard@emse.fr pierre.rust@orange.com

^aInstitut Henri Fayol, MINES Saint-Etienne, France

^bOrange Labs, France

Résumé

Cet article étudie l'applicabilité des techniques d'optimisation sous contraintes distribuée (DCOP) dans le cadre des réseaux radios hétérogènes (HetNets). Nous nous intéressons plus particulièrement au problème de l'association des terminaux utilisateurs aux stations de base, dont l'objectif est de déterminer quel utilisateur (e.g. téléphone mobile) est associé à quelle station de base pour créer un canal de communication. Toutes les stations ne garantissent pas les mêmes qualités de service et d'expérience aux utilisateurs, et la qualité de l'allocation finale (e.g. le débit total obtenu) est un critère à maximiser. Nous présentons ici un modèle DCOP de ce problème d'association, qui se base sur une agentification des stations de base se coordonnant pour obtenir une allocation efficace. Nous évaluons les performances d'algorithmes légers (MGM, MGM-2, DSA) sur un environnement simulé réaliste. Leurs performances sont comparées à celles d'une solution optimale, ainsi qu'à celle de la politique d'association actuellement utilisée dans ces réseaux, à savoir maxSINR.

Mots-clés : DCOP, HetNet, Allocation de ressources, SINR

Abstract

This paper studies the applicability of distributed constrained optimization techniques (DCOP) in the context of heterogeneous radio networks (HetNets). In particular, we are interested in the problem of associating user terminals to base stations, the objective of which is to determine which user (e.g. mobile phone) is associated to which base station to create a communication channel. Not all stations guarantee the same quality of service and experience to users, and the quality of the final allocation (e.g. the total throughput obtained) is a criterion to be maximized. We present here a DCOP model of this association problem, which is based on an agentification of base stations coordinating to obtain an efficient allocation. We evaluate the performance of lightweight algorithms (MGM, MGM-2, DSA) on a realistic

simulated environment. Their performances are compared to those of an optimal solution, as well as to those of the association policy currently used in these networks, i.e. maxSINR.

Keywords: DCOP, HetNet, Resource allocation, SINR

1 Introduction

Dans le contexte des réseaux de communication 5G, l'infrastructure sans fil est constituée de multiples stations de base (antennes émettrices et réceptrices) auxquels les utilisateurs du réseau sont associés pour obtenir un canal de communication et ainsi se connecter au réseau et à Internet. Les stations de base ayant des propriétés différentes (puissance, gain, portée, etc.), on parle alors de réseaux hétérogènes, ou *HetNet*. La plupart des infrastructures cibles considèrent des réseaux à plusieurs tiers –e.g. 3 tiers, avec des stations *macro* à longue portée, des stations *pico* à moyenne portée, et des stations *femto* à courte portée, comme illustré dans la Figure 1. Associer les utilisateurs à des stations de base est un problème d'allocation difficile, dont les objectifs peuvent être de maximiser le signal reçu par chaque utilisateur (pour maximiser sa qualité d'expérience), et de répartir la charge sur les stations de base afin de ne pas réduire le débit par station. Compte tenu de la mobilité des usagers, et donc de la nécessité de fournir des solutions de manière réactive, les approches par optimisation centralisée sont presque inapplicables dans ce contexte.

Nous proposons ici d'évaluer les performances des algorithmes d'optimisation sous contraintes distribuée (DCOP) à ce problème, naturellement distribuable et modélisable comme un système multi-agent. Ici, les stations de base se coordonnent via un algorithme d'optimisation sous contraintes distribuée, pour déterminer quels utilisateurs elles vont servir. L'objectif ici est de maximiser globalement le débit total fourni aux usagers. L'approche classique pour résoudre de tels problèmes d'allocation

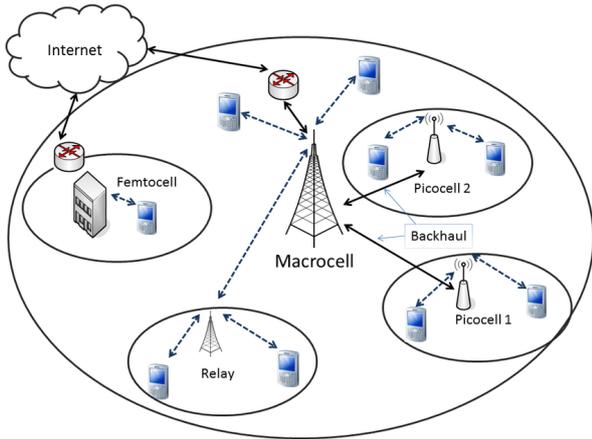


FIGURE 1 – Architecture d’un réseau radio hétérogène à trois tiers (macro, pico, femto) [4]

tion, pour évaluer la qualité des autres approches, est de suivre une approche par programmation mathématique et optimisation en nombres entiers [11]. Du point de vue applicatif, compte tenu de la difficulté à résoudre de tels problèmes en temps réel, des approches heuristiques sont utilisées, et principalement l’approche gloutonne maxSINR [12] qui consiste à associer le terminal à la station exhibant le meilleur rapport signal sur interférence plus bruit (SINR). Cependant, cette approche a le désavantage d’associer les usagers en priorité aux stations de hauts tiers (e.g. macro), et ainsi les surcharger, provoquant éventuellement des problèmes de connexion. A notre connaissance, seul [8] propose une approche par DCOP pour s’attaquer à ce problème d’allocation, mais fait le choix d’un algorithme complet, BnB-ADOPT, sans évaluer les performances d’algorithmes plus légers, comme MGM ou DSA. Nous proposons ainsi, dans cette étude, d’élargir nos connaissances sur l’applicabilité des DCOPs à ce problème.

Cet article est structuré comme suit. Le modèle et les algorithmes DCOP principalement utilisés dans la suite sont brièvement introduits en Section 2. Le problème d’association d’utilisateurs dans les HetNets, ainsi que le modèle du système sont présentés en Section 3. La modélisation de ce problème dans le formalisme DCOP y est également détaillé. La section 4 expose le protocole expérimental ainsi qu’une analyse des résultats obtenus sur un environnement simulé réaliste suivant les scénarios proposés dans la littérature du domaine [8, 2]. Enfin, l’article se conclut autour des perspectives offertes par cette investigation en Section 5.

2 Optimisation sous contraintes distribuée

Cette section introduit brièvement le cadre des DCOPs, ainsi que les algorithmes que nous allons utiliser.

2.1 Modèle canonique

Une façon de modéliser un problème distribué d’allocation de ressources est de le formaliser comme un problème d’optimisation distribuée sous contraintes (DCOP) [18].

Définition 1. *Un problème d’optimisation sous contraintes distribuée (ou DCOP pour Distributed Constraint Optimization Problem) est un tuple $P = \langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu, f \rangle$, où : $\mathcal{A} = \{a_1, \dots, a_m\}$ est l’ensemble d’agents ; $\mathcal{X} = \{x_1, \dots, x_n\}$ sont les variables appartenant aux agents ; $\mathcal{D} = \{\mathcal{D}_{x_1}, \dots, \mathcal{D}_{x_n}\}$ est un ensemble de domaines finis tels que la variable x_i prend ses valeurs dans $\mathcal{D}_{x_i} = \{v_1, \dots, v_{|\mathcal{D}_{x_i}|}\}$; $\mathcal{C} = \{c_1, \dots, c_k\}$ est un ensemble de contraintes souples, où chaque c_i définit une utilité $\in \mathbb{R} \cup \{-\infty\}$ pour chaque combinaison d’affectation de valeurs au sous-ensemble de variables impliquées ; $\mu : \mathcal{X} \rightarrow \mathcal{A}$ est une application associant les variables à leur agent. $f : \prod \mathcal{D}_{x_i} \rightarrow \mathbb{R}$ est une fonction objectif, représentant le coût global d’une affectation de valeurs aux variables.*

L’objectif d’optimisation est représenté par la fonction f , qui peut être de différente nature. En général, cette fonction est une somme des coûts des contraintes : $f = \sum_i c_i$.

Définition 2. *Une solution à un DCOP P est une affectation de valeurs à toutes les variables. Une solution est dite optimale si elle maximise f .*

Notons que la notion d’utilité peut être substituée par la notion de coût $\in \mathbb{R} \cup \{+\infty\}$. Dans ce cas, résoudre un DCOP consiste à minimiser la somme totale des coûts. De plus, il est possible d’intégrer des contraintes dures (comme dans le cas de la satisfaction) en leur attribuant un coût infini en cas de violation.

Comme souligné par [10, 19], les DCOP ont été largement étudiés et appliqués à de nombreux domaines de référence. Ils présentent de nombreuses propriétés intéressantes : (i) focus sur des approches décentralisées où les agents négocient une solution jointe au travers d’échanges locaux de messages ; (ii) exploitation de la structure du domaine (en l’encodant dans des contraintes) pour s’attaquer à des problèmes computationnels

difficiles ; (iii) grande variété de méthodes de résolution allant de méthodes exactes à des techniques heuristiques et approchées ; comme, par exemple, ADOPT [17], DPOP [18], MaxSum [9], DSA [25] ou MGM [15], pour ne citer que les plus connus.

2.2 Algorithmes

De nombreux algorithmes de résolution de DCOP existent et se distinguent par leur type de processus de résolution (recherche/inférence), leur complétude (optimal/approché), ou leur parallélisme (synchrone/asynchrone) [10]. De plus, certains algorithmes vont être capables de gérer mieux que d'autres certains types de contrainte ; e.g. les algorithmes par inférence, comme MaxSum et DPOP, nécessitent une mémoire exponentielle dans l'arité des contraintes, donc seront peu adaptés à des problèmes présentant des contraintes qui impliquent un grand nombre de variables. Nous allons utiliser dans cet article deux algorithmes approchés classiques, et très légers (en charge de calcul et en quantité d'information échangée), à savoir DSA [25] et MGM [15].

DSA. Plus qu'un simple algorithme, DSA (*Distributed Stochastic Algorithm*) est une famille d'algorithmes incomplets de recherche locale, très légers, basée sur une idée assez simple : les agents partent d'une valeur aléatoire de leur domaine et évaluent régulièrement si la qualité de leur propre affectation partielle, définie comme la somme des contraintes dans lesquelles elle est impliquée, pourrait être améliorée en sélectionnant une nouvelle valeur [25]. Cette évaluation est basée sur la connaissance des valeurs actuellement sélectionnées par leurs voisins. Si cette qualité peut être améliorée, l'agent décide aléatoirement, avec une probabilité d'activation p , de sélectionner la valeur correspondante et d'envoyer son état actualisé à ses voisins. Ce processus de recherche est local, car les agents ne fondent leur décision que sur leur connaissance des valeurs de leurs voisins directs. Bien sûr, cet algorithme de recherche locale peut être piégé dans un minimum local (même si la stochasticité de DSA l'aide parfois à échapper à ce minimum local) et ne garantit pas de trouver la solution optimale. Bien que ce ne soit pas strictement le cas, la DSA est un algorithme itératif et peut être utilisée pour obtenir une affectation complète à tout moment, en temps réel, avec une qualité de solution s'améliorant, en moyenne, au fil du temps. Cependant, dans le cas général, DSA ne fournit aucune garantie de monotonie : comme il n'y a pas de coordination dans le processus de décision et que les connaissances locales d'un agent peuvent être dépassées, deux agents peuvent prendre simultanément

des décisions contradictoires, ce qui entraîne une diminution de la qualité du résultat global.

Cinq variantes – à savoir DSA-A, DSA-B, DSA-C, DSA-D et DSA-E – de ce principe de base ont été étudiées [25], en fonction de la stratégie utilisée pour le changement de valeurs. Un agent peut sélectionner une nouvelle valeur de manière plus ou moins agressive, lorsque la qualité de son état peut être améliorée, strictement ou non, et lorsque des conflits subsistent même si la qualité ne peut être améliorée. Ces variantes présentent divers degrés de parallélisme et d'exploration de l'espace des solutions. L'approche DSA-B est considérée comme la plus efficace dans le cas général, mais nécessite une pré-exploration de l'espace des solutions. Il a également été démontré dans [25] que la valeur utilisée pour la probabilité d'activation p a une influence considérable sur l'efficacité et la qualité de DSA et présente une propriété de transition de phase. Lorsque la bonne variante et la bonne probabilité d'activation ont été sélectionnées pour une classe de problèmes donnée, DSA fournit des résultats de très bonne qualité, avec une charge de réseau et de calcul minimale, ce qui la rend très évolutive. Il convient de noter que la DSA est capable de travailler avec des contraintes n -aires sans aucune modification.

MGM. MGM (*Maximum Gain Message*) est une modification de l'algorithme DBA qui se concentre sur la transmission du message à gain maximal [15, 14]. Comme le DSA, MGM est un algorithme de recherche locale incomplète qui peut gérer des contraintes n -aires. MGM est un algorithme synchrone : à chaque tour, les agents calculent le changement maximal de qualité, appelé *gain*, qu'ils pourraient obtenir en sélectionnant une nouvelle valeur et envoient ce gain à leurs voisins. Un agent n'est alors autorisé à changer sa valeur que si son gain est supérieur à celui reçu de tous ses voisins. Ce mécanisme garantit que deux variables impliquées dans la même contrainte ne changeront jamais leur valeur dans le même cycle. Ce processus se répète jusqu'à ce qu'une condition de terminaison soit remplie. Bien qu'il ne fournisse aucune limite à la qualité de la solution, MGM est en mesure de garantir la monotonie ; l'élimination de l'aspect stochastique de la DSA garantit que la qualité de la solution ne s'améliore qu'avec le temps. La monotonie est une qualité très intéressante dans de nombreux domaines d'application, cependant, cette qualité est garantie au détriment d'une tendance plus élevée à se retrouver piégée dans un minimum local. Pour atténuer ce problème, [15] propose une version coordonnée de MGM (généralement MGM-2, mais elle peut être étendue à MGM- k), où k agents peuvent coordonner un changement

simultané (c'est-à-dire dans le même cycle) de valeurs. Cela permet d'éviter certains minima locaux tout en préservant la monotonie de l'algorithme.

3 Modèle du problème d'association

Cette section introduit le problème d'association dans les HetNet et détaille le modèle du système ainsi que sa physique. Une formulation DCOP est alors proposée pour ce problème.

3.1 Problème d'association d'utilisateurs

Le problème d'association d'utilisateurs peut être formulé comme un problème d'allocation de ressources distribuées dans lequel une ressource unitaire se réfère à un bloc de ressources (RB) qui couvre une certaine plage de fréquence et une certaine durée [26]. Pour obtenir une bonne qualité de service, un utilisateur devra réserver plusieurs blocs sur une station, dans la limite disponible. En effet, la qualité de service est proportionnelle au nombre de blocs réservés. Le nombre de blocs à réserver dépend de la puissance de la station et de la position de l'utilisateur par rapport à la station, à cause des pertes de signal.

Ces dernières années, des efforts de recherche importants ont été consacrés au développement de méthodes distribuées, en particulier des méthodes fondées sur la théorie des jeux et des méthodes d'optimisation combinatoire [3, 13, 11, 24]. Ici, la formulation du modèle de système s'inscrit dans le cadre de la programmation mixte non convexe en nombres entiers [11], qui est toujours NP-difficile. La manière d'obtenir la solution est souvent très coûteuse. Les résultats de simulation dans [3] et [24] montrent que la performance des méthodes dans les schémas basés sur la théorie des jeux et l'optimisation combinatoire est meilleure que certaines méthodes géométriques, par exemple maxSINR. Cependant, ces deux approches – théorie des jeux, et optimisation combinatoire – sont limitées par le fait qu'un HetNet est un système dynamique dans lequel le nombre d'utilisateurs et la configuration de la ressource varient fréquemment dans les stations de base.

3.2 Modèle du système

Reprenant le modèle de [8], un tier dans le HetNet indique une macrocellule, une picocellule ou une femtocellule où chaque niveau contient un ensemble de stations de base ayant les mêmes configurations (puissance d'émission et ressource). La ressource configurée au niveau d'une

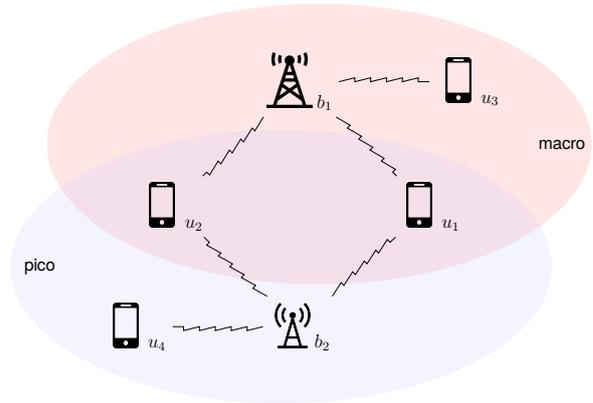


FIGURE 2 – Exemple de configuration de HetNet, avec 4 utilisateurs et 2 stations de base sur 2 tiers

station de base fait référence à un ensemble de blocs de ressources (RB) où chaque RB est constitué d'une certaine durée et d'une certaine largeur de bande [7]. Considérons un HetNet à k tiers comprenant des stations de base $\mathcal{B} = \{b_1, \dots, b_{|\mathcal{B}|}\}$ et des utilisateurs $\mathcal{U} = \{u_1, \dots, u_{|\mathcal{U}|}\}$, comme illustré en Figure 2. Notons N_i le nombre de blocs maximal sur b_i . En supposant que les informations sur l'état du canal soient disponibles au niveau des stations de base, les stations de base de différents niveaux se partagent également la largeur de bande totale, de sorte qu'il y a des brouillages à la fois intra- et inter-niveaux lorsque les stations de base attribuent instantanément des blocs aux utilisateurs. Le rapport signal sur bruit plus interférence (SINR) vécu par l'utilisateur u_j servi par b_i dans le k ième tier, illustré en Figure 3, est donné par :

$$\text{SINR}_{ij} = \frac{P_k g_{ij}}{\sum_{b_l \neq b_i} P_k g_{lj} + BN_0} \quad (1)$$

où P_k est la puissance d'émission pour les stations de base du k ième tier, g_{ij} est le gain de puissance du canal entre u_j et b_i , B est la largeur de bande et N_0 est la densité spectrale de puissance de bruit. Le gain de puissance du canal comprend l'effet de l'affaiblissement du signal et son évanouissement. On suppose que l'affaiblissement sur le trajet est statique et que son effet est pris en compte dans la valeur moyenne du gain de puissance du canal, tandis que les évanouissements sont supposés suivre la distribution exponentielle. Ensuite, l'efficacité de l'utilisateur u_j associé à la station b_i , notée e_{ij} est :

$$e_{ij} = \log_2(1 + \text{SINR}_{ij}) \quad (2)$$

En considérant la largeur de bande B , la durée T et l'intervalle de planification Γ configurés pour

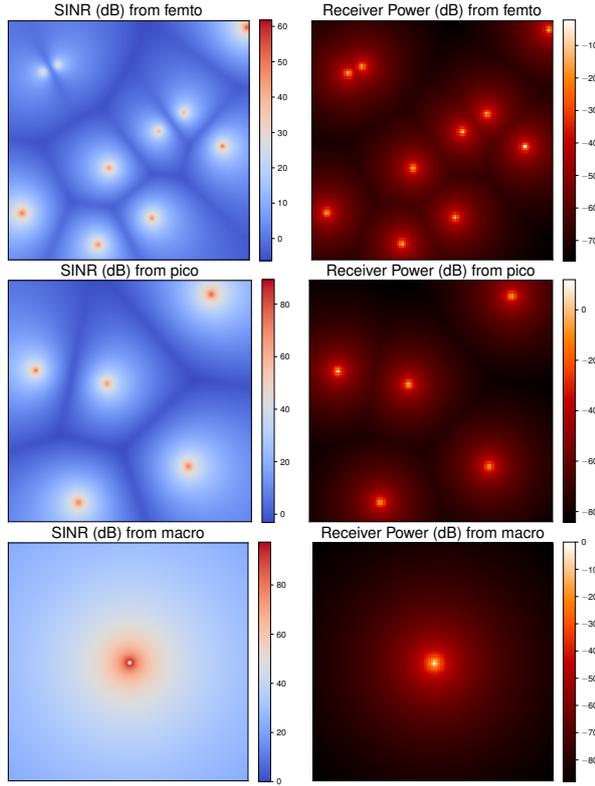


FIGURE 3 – SINR et puissance reçus en chaque point de l’environnement pour les différents tiers d’une infrastructure

chaque RB, on obtient le taux unitaire à u_j sur un RB suivant :

$$q_{ij} = \frac{BT e_{ij}}{\Gamma} \quad (3)$$

et ainsi le taux reçu par u_j avec n_{ij} blocs réservés sur b_j du k ième tier est :

$$r_{ij} = n_{ij} q_{ij} \quad (4)$$

La qualité de service (QoS) de chaque utilisateur est alors exprimée comme le taux total minimum que l’utilisateur pourra obtenir. En posant γ le taux minimal accepté par les utilisateur (e.g. on n’accepte que des services à 50 Mbps), nombre minimum de blocs requis est estimé à :

$$n_{\min}^{ij} = \lceil \frac{\gamma}{q_{ij}} \rceil \quad (5)$$

Maintenant que le modèle physique et les définitions du HetNet ont été posées, nous présentons une formulation DCOP de ce problème.

3.3 Formulation DCOP

Pour définir un DCOP, il faut spécifier le tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu, f \rangle$. Les variables de décision pour une station sont l’ensemble des décisions de servir ou non chaque utilisateur. Notons x_{ij} le nombre de blocs alloués à u_j par b_i , et donc $\mathcal{X} = \{x_{ij} \mid b_i \in \mathcal{B}_j, u_j \in \mathcal{U}\}$, où \mathcal{B}_j est l’ensemble des stations considérées par u_j , et donc nous considérons des domaines binaires $\mathcal{D}_{ij} = \{0, n_{\min}^{ij}\}$. Potentiellement, $\mathcal{B}_j = \mathcal{B}$, mais comme proposé dans [8], pour des raisons de simplification, \mathcal{B}_j peut être l’ensemble des η meilleures stations perçues par u_j du point de vue du SINR, noté \mathcal{B}_j^η . Les contraintes du DCOP, doivent assurer que le nombre de blocs utilisés sur chaque station ne dépasse pas la capacité maximale (contraintes intra-stations regroupées dans $\mathcal{C}_{\text{intra}}$), et que chaque utilisateur soit connecté à au plus une station (contraintes inter-stations regroupées dans $\mathcal{C}_{\text{inter}}$). On a l’ensemble des contraintes du DCOP $\mathcal{C} = \mathcal{C}_{\text{intra}} \cup \mathcal{C}_{\text{inter}}$. Avec ψ_i l’ensemble des utilisateurs intéressés par b_i , on a $\forall c_{\text{intra}}^j \in \mathcal{C}_{\text{intra}}$:

$$R(c_{\text{intra}}^j) = \begin{cases} -\infty & \text{si } \sum_{u_j \in \psi_i} x_{ij} > R_i \\ \sum_{u_j \in \psi_i} r_{ij} & \text{sinon} \end{cases} \quad (6)$$

et $\forall c_{\text{inter}}^i \in \mathcal{C}_{\text{inter}}$:

$$R(c_{\text{inter}}^i) = \begin{cases} 0 & \text{si } \sum_{u_j \in \psi_i} \frac{x_{ij}}{n_{\min}^{ij}} \leq 1 \\ -\infty & \text{sinon} \end{cases} \quad (7)$$

Enfin, on obtient l’équation suivante pour l’objectif global du DCOP :

$$\sum_{c \in \mathcal{C}} R(c) = \sum_{u_j \in \mathcal{U}} \sum_{b_i \in \mathcal{B}_j} r_{ij} \quad (8)$$

La fonction objectif f est donc ici est la somme des taux atteints chez tous les utilisateurs, ce qui peut être considéré comme une mesure du débit dans le HetNet. Ainsi, plus $R(c)$ est grand, plus le débit devrait être bon. Enfin, le dernier choix déterminant est celui de l’ensemble des agents. Une approche intuitive est de considérer les stations comme étant des agents qui vont se coordonner pour savoir quels utilisateurs servir, car elles sont capables de rapidement communiquer entre elles, contrairement aux utilisateurs. Ainsi, ici $\mathcal{A} = \mathcal{B}$. Ce modèle est illustré en Figure 4.

Cette formulation est assez simple, cependant, la présence de contraintes n-aires est une limitation

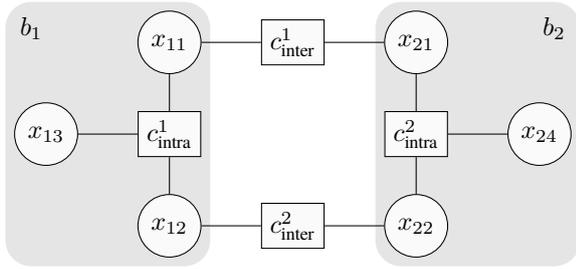


FIGURE 4 – Représentation graphique du DCOP pour le problème de la Figure 2

assez importante pour les algorithmes DCOP par inférence, comme MaxSum ou DPOP, qui deviennent inapplicables quand n est grand car ils reposent sur le calculs de tables de coûts de taille exponentielle en l’arité des contraintes. En effet, les contraintes intra seront connectées à autant de variables que d’utilisateurs intéressés par une station, et dans le cas d’une station macro, elle risque de devoir gérer les demandes de tous les utilisateurs, puisqu’elle proposera un très bon débit. De plus, les contraintes inter connectent autant de variables que de stations intéressant un utilisateur donné, donc potentiellement toutes les stations. Cette limitation des contraintes inter peut cependant être contournée par la fixation du paramètre η à un nombre limité de stations candidates par utilisateur, e.g. 2 ou 3 dans [8].

3.4 À propos d’une formulation binaire

Deux techniques pour binariser des contraintes sont (i) l’ajout de variables (dites cachées) pour chaque contrainte n -aire et de contraintes binaires ou (ii) l’approche duale consistant à remplacer les contraintes par des variables (dites duales) et les variables initiales (dites primales) par des contraintes binaires [1]. Cependant, les nouvelles variables créées ont des domaines qui représentent toutes les combinaisons faisables des valeurs des n variables impliquées dans la contrainte : les valeurs de ces variables sont des vecteurs de taille n . Ainsi, la taille du domaine d’une variable duale représentant une contrainte n -aire est de taille 2^n .

Dans le cadre des contraintes sur variables binaires, une solution efficace pour stocker ces domaines de taille exponentielle consiste à utiliser les diagrammes de décision binaire (ou BDD) [5, 8], c.-à-d. représenter toutes les combinaisons possibles sous la forme d’un graphe orienté non cyclique, compression de l’arbre représentant toutes ces combinaisons. Ainsi, au lieu de stocker une très longue liste de valeurs, on stocke un automate sur lequel on peut itérer pour obtenir les

valeurs du domaine. Cependant, cette technique requiert de compiler les contraintes pour générer les BDDs. Cette opération prend un temps exponentiel dans le nombre de variable dans le pire des cas, et les temps de compilation sont incompatibles avec une utilisation en cours de fonctionnement (prenant plusieurs minutes pour 30 variables, alors qu’ici nous avons jusqu’à $|\mathcal{U}| = 200$).

Ainsi, nous considérons qu’une formulation avec contraintes binaires est inenvisageable dans notre cas.

4 Analyse des performances des algorithmes DCOP

Maintenant que nous avons défini, le problème d’association des utilisateurs dans un HetNet, et sa formulations DCOP, nous allons évaluer les performances de divers algorithmes de la littérature à même de résoudre ces problèmes en un temps raisonnable.

4.1 Cadre expérimental

Nous considérons ici le HetNet à 3 tiers illustré dans la Figure 3, avec 1 station macro, 5 stations pico, et 10 stations femto, inspiré des scénarios de la littérature [8, 2]. Les puissances de transmission de ces stations sont de 40, 35 et 24 dBm, respectivement. L’environnement consiste en un carré de 1000 m \times 1000 m. La macro station est située au centre, et les autres stations sont positionnées aléatoirement par échantillonnage par hypercube latin [16]. Ici, nous considérerons un nombre d’utilisateurs aléatoirement positionnés souhaitant accéder au réseau dans $|\mathcal{U}| \in \{50, 100, 150, 200\}$. Avec $|\mathcal{U}| \geq 100$, le réseau est surchargé, et des utilisateurs seront non servis dans tous les cas. Le modèle de perte de puissance de signal entre une station macro ou pico et les utilisateurs est défini comme $L(d) = 34 + 40 \log_{10} d$, avec d la distance en mètres entre la station et l’utilisateur. De même, pour les stations femto avec $L(d) = 37 + 30 \log_{10} d$. La puissance du bruit est -111.45 dBm, qui est le bruit thermique à la température dans une pièce, avec une largeur de bande de 180kHz. L’intervalle de temps de l’équation (3) est $t = 1$ seconde et $\Gamma = 1$. Le nombre de blocs disponibles est de 100 pour les stations macro, 50 pour les stations pico, et 25 pour les stations femto. Nous considérons $\eta \in \{1, 2, 3\}$.

Les algorithmes suivants sont évalués : (i) allocation optimale, avec CPLEX, en formulant le DCOP n -aire sous forme de programme linéaire en nombres entiers ; (ii) allocation par maxSINR, ce qui revient pour chaque utilisateur à s’associer

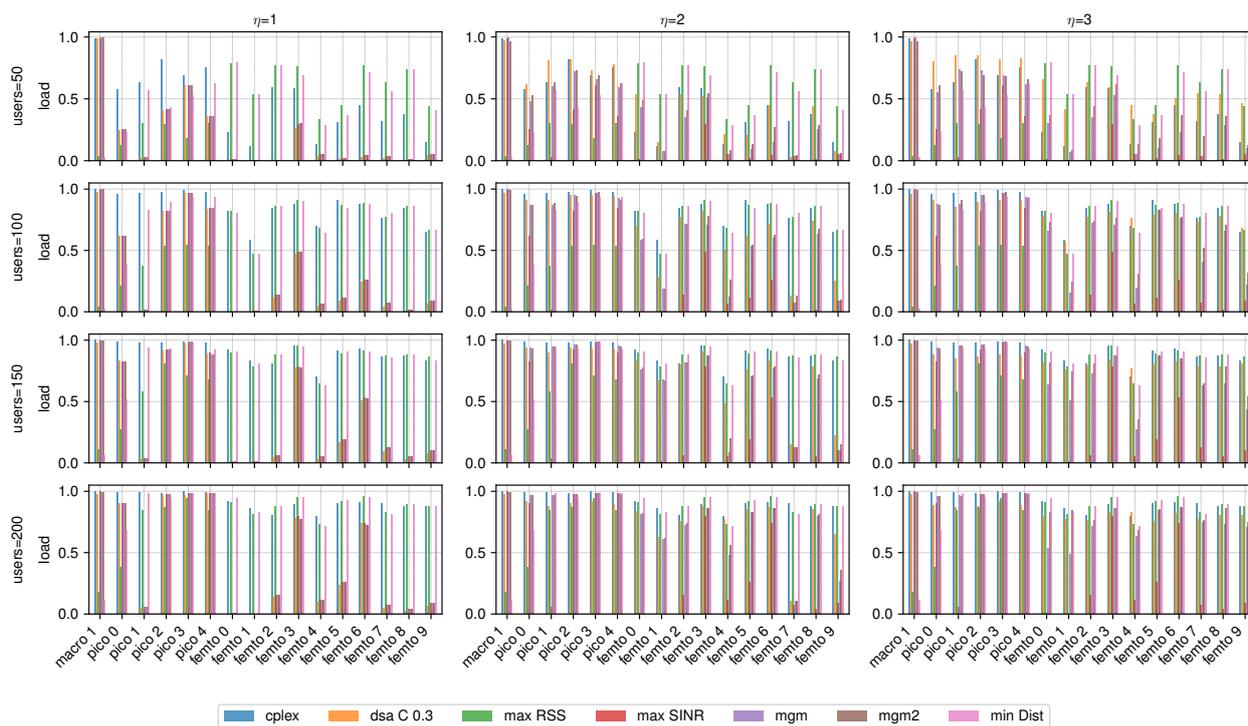


FIGURE 5 – Taux moyen d'utilisations des stations obtenus par chaque algorithme d'allocation pour différents nombres d'utilisateurs et valeurs du paramètre η

à la station disponible ayant le maximum de SINR (en cas de non disponibilité, l'utilisateur passe à un second choix, etc.); (iii) allocation par maxRSS, équivalent à maxSINR, mais avec la puissance sans interférence ni bruit comme critère de choix; (iv) minDist, équivalent à maxSINR, mais avec la distance comme critère de choix; (v) MGM; (vi) MGM-2; (vii) DSA-C (probabilité fixée à 0.3). Nous avons réduit l'étude à ces algorithmes car les algorithmes DCOP optimaux, n'ont pas fourni de résultats en un temps acceptable (timeout de 60s). Pour chaque valeur de η et de N , nous générons 20 instances de DCOP, et chaque instance est résolue 10 fois par chaque algorithme DCOP, partant de l'allocation obtenue par maxSINR. Les autres algorithmes étant déterministes, ils ne sont appliqués qu'une seule fois. Nous affichons la moyenne et l'écart-type (parfois très faible donc peu visible) des mesures sur ces 20 ou 200 instances. MGM, MGM-2 et DSA ont été paramétrés pour fournir des solutions au bout de 20 itérations.

Les calculs ont été effectués sur une machine avec processeur de type Intel(R) Xeon(R) CPU E5-1603 v3 @ 2.80GHz et 32GiB de mémoire, en utilisant la librairie pyDCOP [22, 21], des codes écrits en Python 3.8, excepté pour les solutions

optimales calculées avec CPLEX version 12.61 [6] en utilisant l'API Python via la librairie PuLP [20].

4.2 Analyse des performances

La Figure 5 présente la charge moyenne de chaque station de base pour chaque configuration et chaque algorithme. Comme précisé dans la littérature, maxSINR implique une forte charge sur la station macro, puis, lorsque le nombre d'utilisateurs augmente, la charge se déporte sur les stations de tiers inférieurs. Tous les algorithmes présentent ce comportement, excepté maxRSS et minDist qui vont principalement affecter les utilisateurs aux stations les plus proches, et comme les utilisateurs sont répartis sur tout l'environnement, toutes les stations ont une charge conséquente. Pour $\eta = 1$, notons que les algorithmes DCOP ont le même comportement que maxSINR, et vont donc mettre de côté les stations de tiers inférieurs, puisque les utilisateurs ne considèrent que la meilleure station du point de vue du SINR. Avec $\eta > 1$, les stations partent de l'allocation maxSINR, puis se coordonnent avec un algorithme DCOP pour améliorer la solution. Nous obtenons alors des allocations différentes à maxSINR, proches des allocations optimales calculées par CPLEX, qui sont de meilleure qualité du point de vue des utilisateurs.

1. Cette configuration de DSA a été celle présentant les meilleurs résultats parmi la famille DSA sur ce problème

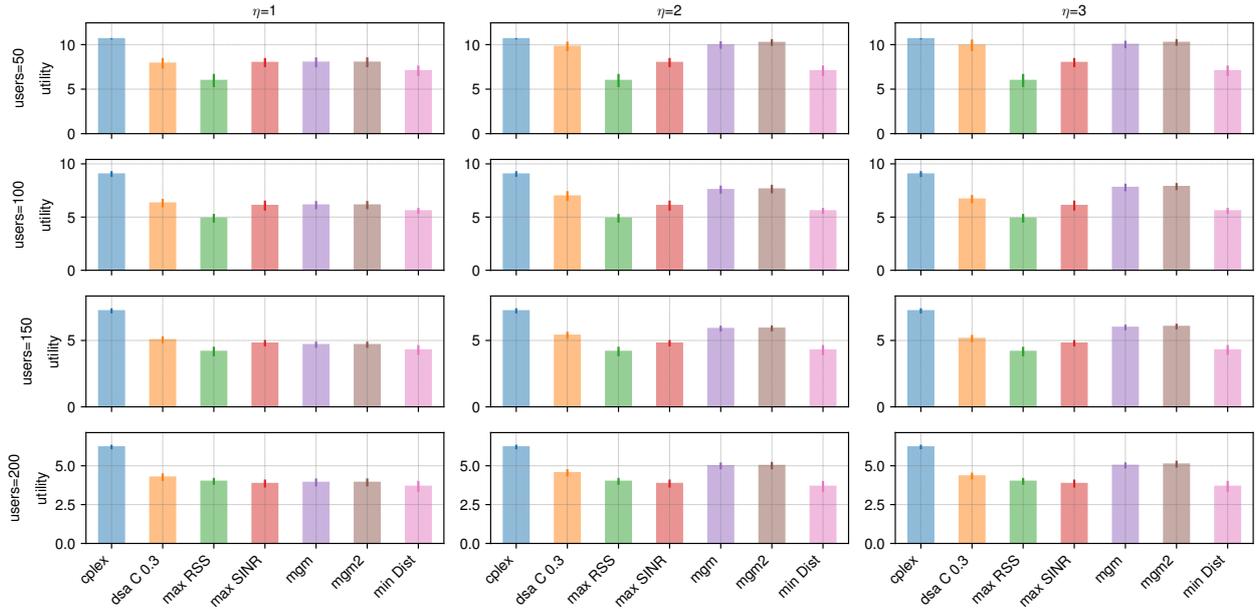


FIGURE 6 – Utilité finale (débit moyen par utilisateur en Mbps) obtenue par chaque algorithme d'allocation pour différents nombres d'utilisateurs et valeurs du paramètre η

En effet, en analysant l'utilité globale du système, c.-à-d. le débit global en Mbps, présenté dans la Figure 6, nous pouvons observer que pour $\eta = 1$, les algorithmes DCOP obtiennent la même qualité que maxSINR (puisque pas de réelle coordination), à part DSA. Ce dernier, grâce à ses mouvements aléatoires, change l'allocation maxSINR initiale, et donc retire des utilisateurs de certaines stations, et ajoutent des utilisateurs initialement non servis, pour obtenir en moyenne des solutions de meilleure qualité. Cependant, cette qualité n'est pas réellement opérationnelle, puisque DSA viole de nombreuses contraintes intra, et donc surcharge des stations au-delà de leur limite. Pour $\eta > 1$, la qualité des solutions MGM et MGM-2 augmente, et se place entre la qualité optimale et la qualité de maxSINR. Avec η augmentant, MGM-2 fournit des qualités de solutions très légèrement supérieures à celles fournies par MGM. Les pires solutions sont obtenues par maxRSS et minDist.

Cette qualité supérieure s'explique ainsi par l'ajout de coordination entre les stations, qui permet de servir plus d'utilisateurs, comme illustré dans la Figure 7. On y retrouve la même performance pour MGM-2 avec $\eta = 3$, qui se positionne entre la solution maxSINR et la solution optimale théorique obtenue par CPLEX. Notons que les instances que nous avons générées sont très stressantes pour le systèmes, puisque tous les utilisateurs ne peuvent être physiquement servis à partir de $|\mathcal{U}| \gtrsim 85$. Sur les systèmes non surchargés ($|\mathcal{U}| = 50$), MGM et MGM-2 améliorent

grandement la part d'utilisateurs servis. Cet écart diminue dans les systèmes surchargés, mais MGM-2 fournit toujours de meilleures solutions.

Cependant cette coordination nécessite une charge réseau conséquente pour la stations, afin de traiter les messages de coordination, comme illustré en Figure 8, mais notons que ces messages DCOP ne se propagent qu'entre stations de base (donc à haut débit, et faible latence), et surtout que ces protocoles sont peu sensibles à la perte de messages, contrairement à une solution qui consisterait à recueillir toutes les données sur les utilisateurs (position, signal, SINR, exigences, etc.) et sur les stations de base (charge), puis les faire remonter à un serveur global pour effectuer une optimisation globale, avec CPLEX, par exemple.

Au final, MGM, pour une quantité d'information moindre que MGM-2, permet toutefois d'obtenir des solutions de bonne qualité, si une surcharge de messages entre stations est problématique.

5 Conclusions

Dans cet article, nous avons étudié les performances d'algorithmes DCOP légers pour l'association d'utilisateurs à des réseaux HetNet. Ici, l'objectif étant une optimisation du débit global obtenu par les utilisateurs, nous avons comparé les algorithmes MGM, MGM-2, et DSA-C à des algorithmes d'allocation classiques dans le domaines (maxSINR, maxRSS) ainsi qu'à

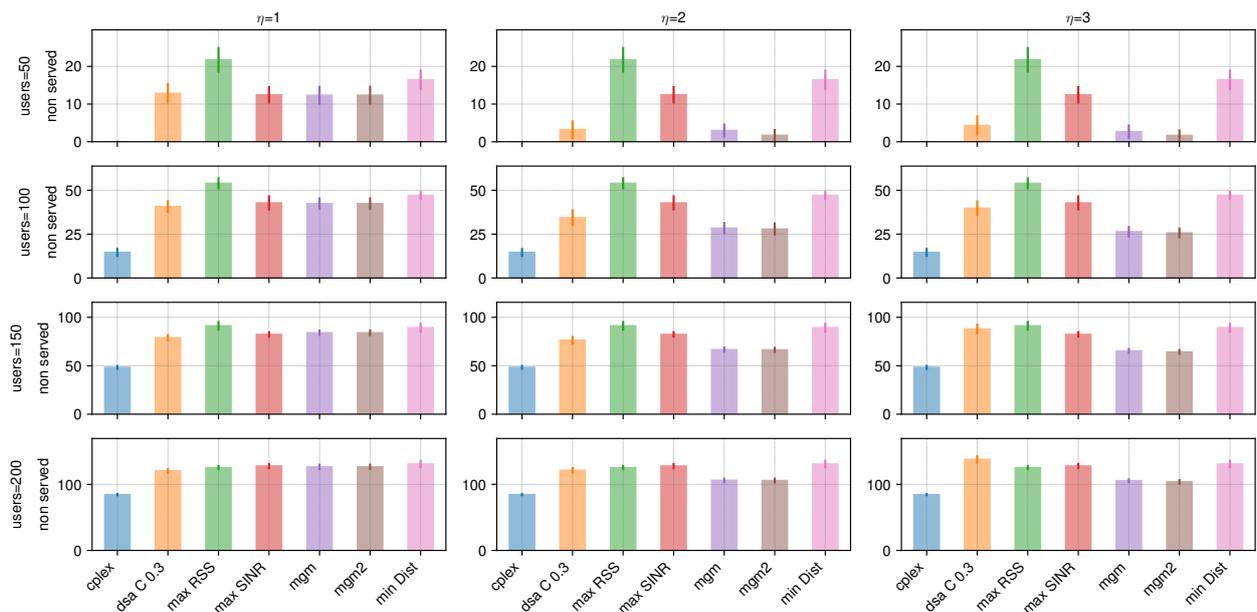


FIGURE 7 – Nombre final d'utilisateurs sans connexion obtenu par chaque algorithme d'allocation pour différents nombres d'utilisateurs et valeurs du paramètre η

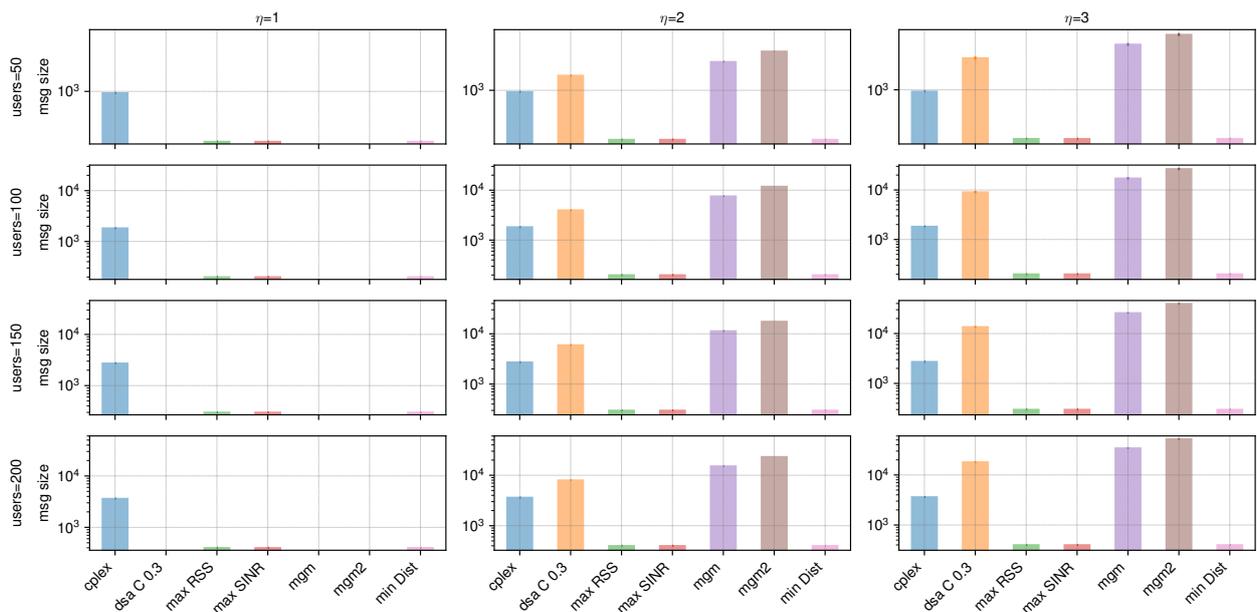


FIGURE 8 – Charge réseau induite par chaque algorithme d'allocation pour différents nombres d'utilisateurs et valeurs du paramètre η

la solution optimale théorique obtenue de manière centralisée par programmation mathématique (CPLEX). Dans des environnements chargés, MGM et MGM-2 exhibent des performances se situant entre la solution distribuée classiquement utilisée, maxSINR, et l'optimum théorique. Ainsi, en ajoutant une couche de coordination par DCOP entre les stations de base, le système améliore les performances normalement obtenues dans un

HetNet. Cette amélioration requiert uniquement un échange de messages entre les stations (et donc à haut débit et sans goulot d'étranglement).

Cette étude aboutit ainsi sur des résultats prometteurs quant à l'utilisation des techniques DCOP dans le cadre des HetNet. Cependant, nous n'avons utilisé que des algorithmes génériques issus de la littérature. Afin d'améliorer encore les

performances, il nous semble nécessaire d'adapter ces algorithmes et leurs structures de données au cas particulier des HetNets, notamment à cause de la présence de contraintes n -aires et de la mixité contraintes souples/contraintes dures. Enfin, l'un des apports envisagés d'une approche décentralisée est la capacité à adapter le système en local, en cas de défaillance d'équipements. Nous envisageons ainsi de poursuivre cette étude dans un cadre DCOP dynamique, en utilisant notamment les mécanismes de résilience proposés dans [23].

Références

- [1] F. Bacchus and P. van Beek. On the conversion between non-binary constraint satisfaction problems. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, page 311–318, USA, 1998. American Association for Artificial Intelligence.
- [2] B. Bikram Kumar, L. Sharma, and S.-L. Wu. Online distributed user association for heterogeneous radio access network. *Sensors*, 19(6), 2019.
- [3] H. Boostanimehr and V. K. Bhargava. Unified and distributed qos-driven cell association algorithms in heterogeneous networks. *IEEE Transactions on Wireless Communications*, 14(3):1650–1662, March 2015.
- [4] S. Boudko, W. Leister, and S. Gjessing. Heterogeneous wireless network selection : Load balancing and multicast scenario. *International Journal on Advances in Networks and Services*, 6 :118–135, 12 2013.
- [5] R.E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, C-35(8) :677–691, 1986.
- [6] <https://www.ibm.com/analytics/cplex-optimizer>.
- [7] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi. A survey on 3gpp heterogeneous networks. *IEEE Wireless Communications*, 18(3) :10–21, June 2011.
- [8] P. Duan, C. Zhang, G. Mao, and B. Zhang. Applying distributed constraint optimization approach to the user association problem in heterogeneous networks. *IEEE Transactions on Cybernetics*, 48(6) :1696–1707, 2018.
- [9] A. Farinelli, A. Rogers, and N. R. Jennings. Agent-based decentralised coordination for sensor networks using the max-sum algorithm. *Autonomous Agents and Multi-Agent Systems*, 28(3) :337–380, May 2014.
- [10] F. Fioretto, E. Pontelli, and W. Yeoh. Distributed constraint optimization problems and applications : A survey. *Journal of Artificial Intelligence Research*, 61 :623–698, 2018.
- [11] D. Fooladivanda and C. Rosenberg. Joint resource allocation and user association for heterogeneous wireless cellular networks. *IEEE Transactions on Wireless Communications*, 12(1) :248–257, 2013.
- [12] K. Gomadam, V. R. Cadambe, and S. A. Jafar. A distributed numerical approach to interference alignment and applications to wireless interference networks. *IEEE Transactions on Information Theory*, 57(6) :3309–3322, June 2011.
- [13] V. N. Ha and L. B. Le. Distributed base station association and power control for heterogeneous cellular networks. *IEEE Transactions on Vehicular Technology*, 63(1) :282–296, Jan 2014.
- [14] M. Jain, M. Taylor, M. Tambe, and M. Yokoo. DCOPs meet the real world : Exploring unknown reward matrices with applications to mobile sensor networks. In *International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 181–186, 2009.
- [15] R.T. Maheswaran, J.P. Pearce, and M. Tambe. Distributed algorithms for DCOP : A graphical-game-based approach. In *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems (PDCS)*, pages 432–439, 2004.
- [16] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1) :55–61, 2000.
- [17] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT : Asynchronous Distributed Constraint Optimization with Quality Guarantees. *Artificial Intelligence*, 161(2) :149–180, 2005.
- [18] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 266–271, 2005.
- [19] Gauthier Picard. Optimisation sous contraintes distribuée : une introduction au domaine. In *Journées Francophones sur les Systèmes Multi-Agents*, pages 43–52. Cépaduès, 2018.
- [20] <https://pythonhosted.org/PuLP/>.
- [21] <https://github.com/Orange-OpenSource/pyDcop>.
- [22] P. Rust, G. Picard, and F. Ramparany. pyDCOP, a DCOP library for IoT and dynamic systems. In *International Workshop on Optimisation in Multi-Agent Systems (OptMAS@AAMAS 2019)*, 2019.
- [23] P. Rust, G. Picard, and F. Ramparany. Resilient distributed constraint optimization in physical multi-agent systems. In *European Conference on Artificial Intelligence (ECAI)*. IOS Press, 2020.
- [24] Z. Wang, X. Zhu, X. Bao, and S. Zhao. A novel resource allocation method in ultra-dense network based on noncooperation game theory. *China Communications*, 13(10) :169–180, Oct 2016.
- [25] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. Distributed stochastic search and distributed breakout : Properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2) :55–87, January 2005.
- [26] L. Zhao, Y. Qin, M. Ma, X. Zhong, and L. Li. Qos guaranteed resource block allocation algorithm in lte downlink. In *7th International Conference on Communications and Networking in China*, pages 425–429, Aug 2012.

Les meilleurs agents sont ceux qu'on ne simule pas : vers des architectures de simulation multi-paradigmes ?

Sébastien Picault, Vianney Sicard

INRAE, Oniris, BIOEPAR, 44300, Nantes, France

{sebastien.picault, vianney.sicard}@inrae.fr

Résumé

La simulation multi-agents, outil privilégié pour l'étude des systèmes complexes, s'est orientée récemment vers deux questions de recherche : la distribution des simulations à large échelle et la modélisation multi-niveaux. Nous souhaitons proposer ici une troisième voie complémentaire : identifier les éléments d'un modèle qui peuvent n'être pas simulés par des agents (et par quoi les remplacer). Cette approche encore prospective, en contribuant paradoxalement à attaquer des problèmes difficiles dans la modélisation de systèmes complexes, est aussi un prétexte à s'ouvrir à d'autres disciplines dont les apports aux SMA pourraient s'avérer particulièrement féconds.

Mots-clés

Simulation multi-agents, modélisation multi-niveaux, interdisciplinarité, épistémologie, parcimonie

Abstract

Agent-based simulation, a key tool for the study of complex systems, recently focused on two research topics : the distribution of large-scale simulations, and multi-level modelling. We wish to suggest here a third and complementary way : to identify model components that would not be simulated by agents (and what to use instead). This approach, still prospective, could paradoxically help tackle hard problems in the modelling of complex systems, while opening up to other research fields whose contribution to MAS could prove particularly fruitful.

Keywords

Agent-based simulation, multi-level modelling, interdisciplinarity, epistemology, parcimony

1 Introduction

En une trentaine d'années, les systèmes multi-agents (SMA) se sont imposés comme un des outils incontournables pour l'étude des systèmes complexes, d'abord pour l'étude de certaines questions relatives aux mécanismes explicatifs de tels systèmes [6], mais aussi pour prédire ou contrôler leur évolution. Nous souhaitons dans cet article jeter un bref regard rétrospectif sur cette « agentification » des méthodes d'investigation scientifique, mais surtout examiner les avancées récentes dans les méthodes de

simulation et proposer un point de vue sur des fronts de science qu'il nous semble indispensable d'explorer dans la prochaine décennie pour conforter la position des SMA en tant que discipline scientifique autonome, pourvoyeuse de méthodes fructueuses pour d'autres domaines de recherche. L'idée quelque peu paradoxale que nous défendons ici est qu'une de ces pistes consiste à chercher à *ne pas simuler d'agents*, aussi souvent que possible (et chaque fois que cela est pertinent).

Après avoir rappelé brièvement la place qu'occupe désormais la simulation multi-agents pour l'étude des systèmes complexes, nous examinerons deux grandes avancées des dernières années, puis nous expliquerons ce que nous entendons par « ne pas simuler d'agents » et la plus-value de cette démarche, avant d'en discuter les difficultés.

2 Des agents infiltrés partout !

Les SMA sont progressivement sortis des laboratoires d'informatique pour s'installer durablement dans les boîtes à outils des scientifiques d'autres disciplines. D'abord présent en sciences sociales [2, 18, 22], en éthologie [16] et en écologie [14], les SMA ont été employés pour étudier des systèmes de plus en plus diversifiés en économie [12], biologie moléculaire [24, 68], microbiologie [41], épidémiologie [62], par des chercheurs de ces domaines.

Dans le même temps, la simulation multi-agents a été largement employée pour apporter un éclairage complémentaire de l'ingénierie sur des systèmes fortement anthropisés pour anticiper leur évolution et la comparaison de scénarios de contrôle ou de pilotage, dans des domaines variés : aménager les transports [19] ou les espaces publics [65] pour la circulation, gérer les agro-écosystèmes [7, 56], prédire la propagation des maladies [71] et comparer des mesures de maîtrise [31].

La maturité atteinte par les SMA permet désormais à tout scientifique un peu curieux, assez persévérant pour acquérir des notions élémentaires de programmation, s'il ne les a déjà reçues durant sa formation, de concevoir et d'implémenter lui-même des modèles multi-agents pour sa discipline. De nombreuses plateformes de simulation ergonomiques se sont imposées comme des outils standard : par exemple, NetLogo [72], Repast [54], Repast Simphony [53], ou encore MASON [42] et GAMA [25].

On pourrait qualifier cette situation d'âge d'or de la simu-

Les meilleurs agents sont ceux qu'on ne simule pas : vers des architectures de simulation multi-paradigmes ?

lation multi-agents. Après une focalisation sur des questions qui ont accompagné la naissance de cette approche (par exemple, « en quoi les agents ne sont-ils pas juste des objets ? qu'est-ce qu'une véritable autonomie ? êtes-vous réactif ou cognitif ? quelle méthodologie de conception employer ? vos agents sont-ils de vrais agents ? » etc.) et qui désormais peuvent sembler quelque peu byzantines, les concepts fondateurs (agents, environnements, comportement, interaction, organisation, phénomène collectif émergent) ont percolé dans de très nombreuses autres disciplines scientifiques qui s'y sont acculturées.

Si tout le monde désormais peut « faire des agents », cela n'est pas sans poser aux chercheurs du domaine des SMA une question majeure : quelles sont les questions brûlantes de notre communauté (pour paraphraser [55]), autrement dit quelles sont les voies de recherche nouvelles à explorer dans les prochaines années pour continuer à apporter une plus-value significative par nos méthodes ? Nous tentons d'apporter ici quelques propositions de réponse, ne serait-ce que pour amorcer le débat.

3 Questions de recherche récentes

Durant la dernière décennie, le développement de la puissance de calcul disponible sous diverses formes a permis aux SMA de s'attaquer à des systèmes à large échelle. Cette extension vers des systèmes bien plus massifs que ceux manipulés auparavant a entraîné en retour le développement de nouvelles méthodes pour répondre à ces défis de complexité croissante. Deux d'entre elles ont été particulièrement motrices ces dernières années.

3.1 Simulations multi-agents distribuées

La première s'intéresse à la distribution des calculs de simulation, en particulier à travers les spécificités des SMA [63]. Cette distribution peut s'appuyer sur des infrastructures variées, telles que des grilles de calcul [11, 57] ou l'utilisation générique de GPU [28, 36]. Quelques plateformes commencent à proposer des capacités de calcul à haute performance, par exemple Repast HPC [13] ou DMASON [15].

Par-delà la question de bien ou mal équilibrer les charges de calcul, il est évidemment crucial de garantir que la distribution préserve certaines propriétés d'ordonnement, de synchronisation, de reproductibilité souhaitées dans la simulation, voire permette d'implémenter de façon efficace des méthodes dont le coût de calcul serait rédhibitoire autrement (par exemple un algorithme de flocking [61] basé sur les seules règles originales de Reynolds [29]). Paradoxalement, les SMA, bien qu'issus de l'IA distribuée, ont tardé à tirer pleinement profit des travaux menés en informatique distribuée : on ne peut donc que se féliciter de ce regain d'intérêt pour ces méthodes sans lesquelles la simulation de systèmes à très large échelle ne serait pas envisageable.

3.2 Simulation multi-agents multi-niveaux

De façon concomitante, une deuxième branche a connu un essor considérable depuis dix ans : la simulation multi-

niveaux (ou multi-échelles, souvent notée ML-ABM, *Multi-Level Agent-Based Modelling*), dans laquelle les niveaux d'organisation, d'observation, d'abstraction intermédiaires entre des individus atomiques (insécables, non décomposables) et le système dans son ensemble sont représentés en tant que tels par des agents dotés de comportements spécifiques.

Les premiers travaux en la matière sont assez anciens mais rares : on se souviendra notamment de la plateforme SWARM [48] à l'architecture récursive, ou de la simulation hydrologique d'agents « boules d'eau » capables de s'agréger en mares ou en ravines [64]. Ce n'est toutefois qu'assez récemment que ce domaine a pris une ampleur majeure. Cela s'est traduit par un nombre considérable d'applications dans de très nombreux domaines [49], mais aussi par l'élaboration de méta-modèles pour traiter cette problématique selon des perspectives complémentaires [8, 33, 50, 44].

Certaines plateformes de simulation commencent également à proposer des mécanismes pour prendre en compte explicitement divers niveaux, par exemple NetLogo sous la forme de l'extension LevelSpace [30]. Par ailleurs, l'étude comparée des applications multi-niveaux développées ces dernières années a permis d'identifier des *design patterns* [43] qui reflètent les solutions mobilisées pour articuler les divers niveaux d'agents dans ces simulations. Ces patterns ont également montré leur intérêt pour la conception d'architectures de simulation multi-niveaux, par exemple en épidémiologie [59].

Les questions abordées par la modélisation multi-niveaux sont intrinsèquement difficiles. Agréger des comportements microscopiques en un comportement macroscopique n'est pas simple, mais l'opération contraire, c'est-à-dire décomposer une entité en un système composé des entités sous-jacentes, créées dans un état cohérent et avec les comportements appropriés, est un verrou scientifique majeur.

3.3 Abondance de biens nuirait-elle ?

La distribution des simulations et l'approche multi-niveaux sont deux voies assurément fécondes et nous n'entendons pas dissuader quiconque de s'y engager, bien au contraire – nous y avons même contribué. Néanmoins, il nous semble qu'une troisième piste reste encore largement inexploree, peut-être simplement parce qu'elle n'est pas perçue en tant que telle. Pour en donner une formulation un peu provocatrice, il s'agit de la *non-simulation*. Autrement dit : il est fort utile de développer des méthodes permettant de simuler des millions, des milliards d'agents, ou des myriades d'agents composés eux-mêmes de myriades d'agents, mais ne devrait-on pas parfois tout simplement ne pas les simuler ?

Cette question n'est évidemment pas une simple boutade. Dès les origines de la simulation multi-niveaux [64], l'un des enjeux affichés en agrégeant des agents « boule d'eau » en agents macroscopiques « mares » ou « ravines », était précisément de s'affranchir du coût de calcul sous-jacent, en le réduisant à des grandeurs agrégées (hauteur d'eau,

volume, vitesse moyenne, etc.). Toute une partie des recherches sur les simulations multi-agents multi-niveaux porte même sur ce sujet.

Mais on peut aller plus loin en réfléchissant à ce que serait une stratégie *d'évitement* de la réification sous forme d'agent. Une telle stratégie consiste à se placer délibérément à contre-courant de la tendance actuelle. L'accroissement de la puissance de calcul disponible a certes favorisé l'émergence des deux grandes questions de la distribution et du multi-niveaux, mais aussi sans doute instillé de façon plus insidieuse l'idée qu'au fond, simuler des systèmes complexes à large échelle consiste d'abord à simuler toujours plus d'agents.

D'une certaine façon, simuler toujours plus d'agents, c'est céder à une solution de facilité qui, certes, fait progresser les architectures de simulation notamment en termes de performances, mais n'aide pas à résoudre des problèmes durs inhérents aux systèmes complexes à large échelle (couplage d'échelles, aller-retour entre plusieurs niveaux de granularité, meilleure intelligibilité des mécanismes à l'œuvre, etc.). Pire, il n'est pas exclu que cela nous amène à construire des systèmes aussi complexes que ceux que nous voulons comprendre, prédire ou contrôler, avec une perte d'intelligibilité importante [4]. Des inquiétudes similaires se font jour d'ailleurs concernant la manière dont le « Big Data » peut conduire à un délitement de la méthode scientifique en général, en remplaçant une démarche expérimentale inscrite dans la recherche ciblée de liens de causalité par une recherche de motifs et de corrélation [66]. De même, pour les SMA, aller vers des systèmes toujours plus massifs sous couvert d'une modélisation sans cesse plus détaillée des systèmes cibles, pourrait conduire à une perte progressive du caractère explicatif des modèles multi-agents, qui constituait originellement leur plus-value principale.

4 Éviter les agents : pourquoi, comment ?

Supposons maintenant qu'un chercheur confronté à un système complexe à large échelle, plutôt que de se demander : « comment vais-je faire pour simuler tous ces agents ? », ait au contraire pour réaction première, à la façon de Bartleby : « je préférerais ne pas » [47]. Pour éviter de sombrer comme Bartleby dans un non-agir perpétuel, la question suivante pour ce chercheur serait d'identifier ce qui, dans le modèle, peut être simulé autrement qu'avec des agents, et avec quelles méthodes. Nous connaissons tous le principe de parcimonie (le fameux « rasoir d'Ockham »), qui pose comme règle méthodologique de ne pas multiplier les entités au-delà du nécessaire, et l'appliquons ordinairement de bon gré aux classes d'agents. Il s'agit donc ici au fond de l'appliquer aussi aux instances. Quels seraient les avantages d'adopter de façon systématique ce type de questionnement ?

4.1 Renouer avec une interdisciplinarité méthodologique

Tout d'abord, une telle démarche permettrait d'éviter un enfermement disciplinaire. Depuis leur conceptualisation au début des années 1990, les SMA sont apparus comme une méthode extrêmement féconde. La conjonction de problématiques à l'interface entre informatique (modèles de tableaux noirs, langages orientés objets, langages d'acteurs) et sciences du comportement (sociologie, microéconomie, éthologie) a induit un questionnement spécifique en termes de méthodologie, d'algorithmique, d'architecture, de communication, de sorte que ces recherches se sont structurées en une discipline autonome.

Depuis, l'interdisciplinarité reste fortement implantée au cœur de très nombreux travaux et s'avère mutuellement bénéfique : les SMA sont un outil efficace pour aider à comprendre ou prédire des systèmes complexes, et inversement appliquer des architectures de simulation, des métamodèles, etc. à des problèmes concrets est une pierre de touche de la pertinence des méthodes développées.

Dans cette dynamique, se profile néanmoins le risque de restreindre peu à peu l'interdisciplinarité originelle des SMA à une juxtaposition de questions disciplinaires, avec d'un côté, les problématiques propres aux SMA, et de l'autre, les questions pratiques des « domaines d'application ».

Réfléchir d'une façon radicalement neuve à la façon de remplacer des agents par d'autres types de modèles nous obligerait à renouer avec les origines intrinsèquement interdisciplinaires des SMA en allant chercher dans d'autres disciplines scientifiques (statistiques, processus stochastiques, systèmes dynamiques, physique, biologie des systèmes...) des *méthodes* éprouvées et des *questionnements* sur les fronts de science.

À titre d'exemple, on peut citer les travaux qui s'appuient sur des méthodes issues des statistiques, de la physique ou même de l'apprentissage automatique pour approximer ou discrétiser des modèles multi-agents [20, 21], pour calibrer les modèles [73] ou explorer l'espace de paramètres [17, 32, 39].

4.2 Comparer et coupler les formalismes

Deuxièmement, chercher à ne pas agentifier, c'est-à-dire à remplacer des agents par une dynamique décrite par un autre type de modèle, c'est s'obliger à une formalisation plus nette de « ce qui se passe » dans le système, dans l'agent, dans l'environnement, pour permettre une connexion effective, opérationnelle, à des modèles plus fortement mathématisés.

Cela suppose aussi de travailler sur la comparaison fine entre les résultats produits par une simulation multi-agents et ceux produits par d'autres méthodes de simulation : équations différentielles, modèles à compartiments, autres approches stochastiques, en s'attachant à mieux comprendre les apports spécifiques de chacune et comment passer de l'une à l'autre [1, 10, 23, 40, 51].

Il est indispensable de développer des méthodologies fon-

Les meilleurs agents sont ceux qu'on ne simule pas : vers des architectures de simulation multi-paradigmes ?

dées sur des critères objectifs, quantitatifs, pour déterminer ce qui doit être agentifié pour rendre compte d'un phénomène donné, et ce qui peut ne pas l'être – et dans ce cas, quelle méthode utiliser à la place pour obtenir la même dynamique qu'avec des agents. Dans certains cas en effet une dynamique agrégée est largement suffisante pour décrire un phénomène ou une partie d'un phénomène, et les agents doivent être mobilisés lorsqu'ils apportent une plus-value clairement identifiable.

La simulation dans le domaine écologique a été régulièrement confrontée à cette question, d'abord en raison du nombre d'entités potentiellement très grand à représenter, et ensuite, du fait de l'existence de processus physiques ou biologiques déjà bien décrits sous des formes mathématiques, qui peuvent être couplés aux comportements des agents (par exemple dans la plateforme MAELIA [67]).

Une autre voie intéressante pour définir des méthodes capables de connecter différentes sortes de modèles consiste à s'inspirer des travaux sur la multi-modélisation [74], la co-simulation [9] ou le couplage de modèles [5, 60] : il est alors nécessaire d'explicitier les règles qui doivent être mises en œuvre à l'interface entre un modèle SMA et un modèle d'une autre nature.

4.3 Attaquer les problèmes difficiles

Une autre conséquence de l'approche que nous défendons est qu'elle oblige à se focaliser sur les problèmes durs de la simulation multi-agents multi-niveaux. Se pose particulièrement la question du changement de granularité non pas des seules entités mais de leurs comportements : comment agréger des comportements individuels en un comportement macroscopique et inversement, de quels comportements doter les entités issues de la désagrégation d'un agent en ses composantes sous-jacentes, restées implicites jusqu'alors ?

Si l'on transpose ces questions à un système multi-niveaux multi-paradigmes, on voit que la question de l'agrégation se rapporte plutôt à la façon de paramétrer par exemple un système équationnel à partir des comportements individuels [60]. La désagrégation, elle, requiert d'être capable de reconstruire des populations synthétiques [27, 45, 73] dans un état cohérent à partir d'une description macroscopique synthétique.

4.4 Formaliser la dualité agent/information

Nous avons régulièrement soutenu qu'il était préférable, dans une simulation multi-agents, de représenter toutes les entités de façon homogène par des agents [38]. Néanmoins, il n'est pas nécessaire de toujours simuler toutes les entités possibles d'un modèle. Cette idée n'est d'ailleurs pas neuve : on en trouve des exemples récurrents dès qu'on aborde la question de l'information portée par l'environnement [46].

En particulier, la notion de phéromone est un bon exemple de ce qui, en toute rigueur, devrait être modélisé par des agents (ce sont bien, après tout, des molécules qui se déplacent, s'évaporent, etc.) mais qu'il est plus « simple » de représenter par une information. Cette dualité agent/in-

formation évoque la notion de champ en physique [69] et mériterait d'être généralisée.

Ainsi, par exemple, en épidémiologie, on considère ordinairement que des individus susceptibles (S) peuvent s'infecter au contact d'individus infectieux (I). Chaque S subit donc une force d'infection calculée en fonction de la probabilité de rencontre avec un I . Ce qui ne porte pas à conséquence pour une population de taille raisonnable et peu structurée (i.e. où les contacts sont aléatoires) devient problématique si l'on envisage une grande métapopulation spatialisée, dominée par des populations d'individus entièrement S : itérer sur ces populations saines pour savoir si un de leurs membres va s'infecter est extrêmement coûteux. Mieux vaut alors inverser le processus, c'est-à-dire ne simuler que les populations contenant des individus I , pour identifier à quelles populations S ils vont transmettre la maladie. Pour ce faire, il faut être capable de *ne pas simuler* les populations saines tout en sachant où elles sont virtuellement situées, de façon à calculer une probabilité d'infection fonction de la distance aux diverses populations I . Le champ de probabilité d'infection autour de chaque population I doit donc s'instancier en certains points, par la création d'une population nouvellement infectée. Cela suppose également de savoir créer cette nouvelle population dans un état cohérent, notamment si d'autres processus que l'infection influent sur sa dynamique (démographie, mouvements...).

4.5 Renforcer la robustesse et la validation des simulations

Enfin, la démarche consistant à restreindre les entités agentifiées à un petit noyau pertinent permettra de faire évoluer ou de renforcer les méthodes de validation des SMA. D'abord, remplacer un agent ou un groupe d'agents par un comportement « résumé » (statistiquement ou stochastiquement), pose évidemment la question de ce que cela change aux résultats. Identifier en quoi l'utilisation d'agents fortement hétérogènes, ou d'un résumé statistique, ou d'une équation déterministe impacte les résultats d'une simulation [1, 10] est un enjeu méthodologique majeur en termes de prédiction.

Or, cette capacité prédictive des SMA prend une importance croissante avec leur utilisation à des fins d'appui à la décision pour le contrôle d'un système complexe (simulation du marché de l'emploi, des politiques d'aménagement urbain, smart cities, transports, compréhension des dynamiques de marchés financiers, évacuation de bâtiments, propagation et maîtrise de maladies à l'échelle de populations). Les SMA ne sont plus cantonnés désormais à un rôle exploratoire où leur capacité explicative faisait l'essentiel de leur plus-value : actuellement leur pouvoir prédictif, quantitatif, est challengé par d'autres approches, notamment les systèmes « boîte noire » (issues par exemple des dernières avancées en apprentissage automatique comme les réseaux de neurones profonds [34]) où seule compte l'exactitude de la prédiction. Nous sommes tous conscients des limitations de ce type d'approche (biais, dépendance forte à la qualité des données, mauvaise capacité d'extra-

polation ou d'exploration de situations alternatives) mais il importe que les SMA, porteurs d'une intelligibilité plus grande sur le système modélisé, soient en mesure de leur tenir tête pour les aspects prédictifs.

La robustesse des prédictions d'un SMA est souvent l'objet de critiques, d'autant que nous avons quelque retard sur des pratiques d'analyse numérique de modèles déployées depuis longtemps pour d'autres paradigmes de calcul. En particulier, dans la partie expérimentale des articles de simulation multi-agents, des informations élémentaires telles que le nombre de répétitions stochastiques effectuées ou la variance des résultats sont trop souvent absentes. L'analyse de sensibilité est une pratique quasi-inexistante alors qu'elle apporte des indications précieuses sur le lien entre paramètres d'entrée et sorties [70] et peuvent faciliter l'identification de paramètres pertinents lors de la calibration des modèles [3]. Elles sont généralement coûteuses en temps de calcul [52] et seraient donc facilitées par le développement de modèles parcimonieux.

Enfin, des méthodes statistiques peuvent être mobilisées pour gérer l'incertitude des prédictions due à la stochasticité généralement forte des SMA [32, 37] : elles pourraient ainsi jouer un rôle analogue aux traitements numériques utilisées pour la correction des aberrations des instruments optiques, en permettant de tirer le plus d'information possible de simulations mêmes non représentatives de comportements moyens.

4.6 Le génie logiciel pour une approche intégrative de la modélisation

Les efforts à déployer pour combiner des approches multi-agents avec d'autres types de modèles, d'une façon systématique et basée sur des critères objectifs, ne pourront pas se limiter à une vague méthodologie descriptive. Il sera nécessaire d'élaborer, conjointement aux avancées théoriques, une méthodologie véritablement *opérationnelle*, étroitement couplée à des méthodes de génie logiciel tout particulièrement adaptées pour la modélisation de systèmes complexes [58].

En particulier, l'élaboration de *design patterns* est la voie la plus prometteuse pour systématiser le lien entre problèmes récurrents et solutions efficaces [26, 35, 43, 46] Cette approche permet de capitaliser de façon incrémentale et modulaire ces solutions, tout en les formalisant et en leur donnant une forme générique qui permet leur ré-emploi. Un couplage de tels patterns avec des langages dédiés (*Domain-Specific Languages, DSL*) permettrait en outre de faciliter la conception de simulations de systèmes complexes en permettant l'assemblage de briques de simulation correspondant à des paradigmes de modélisation différents, et en garantissant une accessibilité de scientifiques non-modélisateurs aux hypothèses, à la structure du modèle, aux connexions entre sous-modèles, etc. [59].

5 Difficultés et remèdes

La voie que nous défendons n'est certes pas dénuée d'embûches, dont certaines, évidentes, sont identifiables à ce

stade, ainsi que les solutions générales pour passer outre. Certaines difficultés inattendues ne manqueront pas de se dresser devant nous, mais nous ne les découvrirons (et ne les vaincrons) qu'après nous être engagés sur le chemin.

5.1 Les difficultés de l'interdisciplinarité

L'interdisciplinarité est chronophage, nous en faisons régulièrement l'expérience dans tous nos travaux « applicatifs ». Il faut assimiler la langue de l'autre, expliquer la spécificité et la plus-value de nos méthodes, s'acculturer avant de pouvoir produire des résultats mutuellement bénéfiques. Le domaine d'application récolte alors le fruit de méthodes dont la finesse de grain permet de représenter un niveau de détail précieux, tout en servant de pierre de touche à nos toutes dernières idées.

Toutefois, la valorisation de tels travaux est difficile : il existe tant d'applications des SMA à des domaines de plus en plus divers, à partir d'architectures classiques, et souvent le fait de chercheurs qui ne sont pas informaticiens, qu'il est dur de montrer en quoi une nouvelle application se démarque *par ses méthodes*. Quant aux scientifiques de la discipline cible, le détail de nos innovations n'attire que rarement leur attention.

S'il est parfois difficile de développer une collaboration de relativement court terme pour modéliser tel ou tel système concret, on se doute qu'une interdisciplinarité « amont », focalisée sur une collaboration de long terme pour échanger des méthodes et en construire de nouvelles, est un investissement d'importance. Dans un contexte où la recherche est soumise à une pression « productiviste », malheur à l'homme seul ! Le risque pour un chercheur (notamment en début de carrière) à s'engager dans cette voie n'est pas négligeable, ne serait-ce qu'en termes d'isolement disciplinaire. Il n'en va plus de même si cette interdisciplinarité méthodologique devient systémique, soutenue par des équipes entières.

5.2 Trouver les bonnes méthodes

Quelles méthodes mobiliser pour construire des architectures de simulation hybrides, avec quels formalismes, quels algorithmes, issus de quelles disciplines ? C'est précisément le cœur du sujet et le programme des recherches à mener durant la prochaine décennie.

Nous avons cité plus haut quelques exemples de travaux qui démontrent un souci de couplage entre les SMA et d'autres méthodes : physique statistique, analyse spatiale, analyse numérique de modèles, populations synthétiques... Il sera nécessaire de prendre le temps de comprendre comment d'autres disciplines abordent les questions de modélisation. Si certaines, rares, font partie des quelques domaines où la bonne parole multi-agents n'a pas encore été portée, la majorité au contraire connaît leur existence et les utilise, à côté d'approches équationnelles ou d'autres méthodes stochastiques, voire de techniques d'apprentissage automatique qui promettent monts et merveilles en matière de simulation [34].

Cette étape sera longue et pénible, car les SMA s'appliquent plus souvent sur des systèmes à bases de règles que

Les meilleurs agents sont ceux qu'on ne simule pas : vers des architectures de simulation multi-paradigmes ?

sur des formalismes mathématiques. Néanmoins, si nous ne sommes pas moteurs pour nous approprier ces méthodes et les incorporer dans les approches de simulation multi-agents, d'autres disciplines ne manqueront pas d'élaborer leurs propres méthodes hybrides dont les SMA seront un composant parmi d'autres.

5.3 Des modèles parcimonieux face à des ressources de calcul toujours plus grandes ?

Pendant que nous œuvrons à réduire la quantité d'agents à simuler, les infrastructures de calcul n'en continueront sans doute pas moins à croître. Faudra-t-il vraiment dépenser de la sueur et des larmes à coupler nos modèles multi-agents, avec leurs élégantes architectures et leurs règles de comportements, à toutes sortes de formalismes dont nous avons su jusqu'à présent nous passer, pour gagner un peu de temps de calcul ? Les temps héroïques de la programmation où il fallait économiser le moindre octet sont révolus ! À ceci, on pourrait d'abord donner une réponse qui ne relève pas d'arguments scientifiques mais d'attentes sociétales : viendra peut-être un moment où l'on attendra des méthodes de simulation qu'elles justifient de leur contribution à une meilleure connaissance, à un contrôle efficace des systèmes modélisés, au regard de leur empreinte carbone. De plus, la simulation étant utilisée dans des proportions de plus en plus large, il faut être capables de partager une ressource qui reste limitée.

Mais surtout, des modèles plus parcimonieux trouveront leur avantage à disposer de ressources de calcul en croissance, car réduire le coût de calcul des simulations permettra d'en faire plus. En particulier, ce sera l'occasion de comparer un plus grand nombre de scénarios ou d'explorer plus largement l'espace des paramètres possibles. Il sera nécessaire de systématiser les analyses de sensibilité, afin de mieux caractériser l'impact des variations de paramètre sur les sorties des modèles, et par conséquent renforcer la confiance qui peut être accordée aux prédictions de nos simulations. Ce type d'analyses, extrêmement coûteux mais indispensable pour approfondir notre capacité à comprendre et contrôler les systèmes complexes, ne sera possible que sur des modèles moins gourmands en ressources.

6 Conclusion

La simulation multi-agents a fait si efficacement la preuve de son intérêt pour l'étude des systèmes complexes à large échelle, que désormais nombre de scientifiques non-informaticiens en font usage de façon autonome pour l'appliquer à leur domaine de recherche. De plus, les méthodes que nous développons pour traiter de systèmes de plus en plus diversifiés et à des échelles toujours plus larges, en particulier la simulation distribuée et la simulation multi-niveaux, contribuent à élargir les possibilités offertes à nos collègues, tout en faisant courir le risque de nous transformer en simples pourvoyeurs de plateformes.

La plus-value de nos recherches ne peut venir que du développement de méthodes « subtiles » pour concevoir des architectures, des méthodes, des algorithmes novateurs, mais

aussi rationaliser et fiabiliser les approches par simulation, construire progressivement des élaborations théoriques à partir de méthodes empiriques pour gagner en maturité, et enfin renforcer la capacité à délimiter la crédibilité des résultats de simulation par une analyse numérique des propriétés de nos modèles.

Pour cela nous avons suggéré ici, à rebours de la tendance générale, de penser systématiquement en termes parcimonieux, pour identifier les éléments de modélisation qui peuvent gagner à n'être pas agentifiés, et inversement ceux où l'agentification est pleinement justifiée. Cela suppose un effort collectif pour être moteurs sur l'incorporation au sein des approches multi-agents de méthodes issues d'autres disciplines.

Références

- [1] Marco Ajelli, Bruno Gonçalves, Duygu Balcan, Vittoria Colizza, Hao Hu, José J. Ramasco, Stefano Merler, and Alessandro Vespignani. Comparing large-scale computational approaches to epidemic modeling : Agent-based versus structured metapopulation models. *BMC Infectious Diseases*, 10(1), 2010.
- [2] Robert Axelrod. Advancing the art of simulation in the social sciences. In Rosaria Conte, Rainer Hegselmann, and Pietro Terna, editors, *Simulating Social Phenomena*, volume 456 of *Lecture Notes in Economics and Mathematical Systems*, pages 21–40. Springer, 1997.
- [3] Rémy Beaudouin, Gilles Monod, and Vincent Ginot. Selecting parameters for calibration via sensitivity analysis : An individual-based model of mosquitofish population dynamics. *Ecological Modelling*, 218(1-2):29–48, 2008.
- [4] U Bhalla. Understanding complex signaling networks through models and metaphors. *Progress in Biophysics and Molecular Biology*, 81(1):45–65, 2003.
- [5] Georgiy V. Bobashev, D. Michael Goedecke, Feng Yu, and Joshua M. Epstein. A hybrid epidemic model : Combining the advantages of agent-based and equation-based approaches. *Winter Simulation Conference*, 2007.
- [6] Eric Bonabeau. Agent-based modeling : Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences (PNAS)*, 99(3):7280–7287, 2002.
- [7] François Bousquet and Christophe Le Page. Multi-agent simulations and ecosystem management : a review. *Ecological Modelling*, 176(3-4):313–332, 2004.
- [8] Benjamin Camus, Christine Bourjot, and Vincent Chevrier. Multi-level modeling as a society of interacting models. In *Agent-Directed Simulation Symposium (in SpringSim)*. SCS/ACM, 2013.
- [9] Benjamin Camus, Thomas Paris, Julien Vaubourg, Yannick Presse, Christine Bourjot, Laurent Ciar-

- letta, and Vincent Chevrier. Co-simulation of cyber-physical systems using a DEVS wrapping strategy in the MECSYCO middleware. *Simulation*, 94(12):1099–1127, 2018.
- [10] Hung-Jui Chang, Jen-Hsiang Chuang, Tsurng-Chen Chern, Mart Stein, Richard Coker, Da-Wei Wang, and Tsan-sheng Hsu. A Comparison Between a Deterministic, Compartmental Model and an Individual Based-stochastic Model for Simulating the Transmission Dynamics of Pandemic Influenza. In *Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pages 586–594. Scitepress, 2014.
- [11] Dan Chen, Georgios K. Theodoropoulos, Stephen J. Turner, Wentong Cai, Robert Minson, and Yi Zhang. Large scale agent-based simulation on the grid. *Future Generation Computer Systems*, 24(7):658–671, 2008.
- [12] Shu-Heng Chen. Varieties of agents in agent-based computational economics : A historical and an interdisciplinary perspective. *Journal of Economic Dynamics and Control*, 36(1):1–25, 2012.
- [13] Nicholson Collier and Michael North. Parallel agent-based simulation with Repast for High Performance Computing. *Simulation*, 89(10):1215–1235, 2013.
- [14] Patrick Coquillard and David Hill. *Modélisation et simulation d'écosystèmes*. Masson, Paris, 1997.
- [15] Gennaro Cordasco, Rosario De Chiara, Ada Mancuso, Dario Mazzeo, Vittorio Scarano, and Carmine Spagnuolo. Bringing together efficiency and effectiveness in distributed simulations : The experience with D-Mason. *SIMULATION*, 89(10):1236–1253, 2013.
- [16] Alexis Drogoul and Jacques Ferber. Multi-agent simulations as a tool for modeling societies : Application to social differentiation in ant colonies. In Cristiano Castelfranchi and Eric Werner, editors, *Artificial Social Systems. 4th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'92)*, volume 830 of *Lecture Notes in Computer Science*, pages 3–23. Springer, 1994.
- [17] Mert Edali and Gönenç Yücel. Exploring the behavior space of agent-based simulation models using random forest metamodels and sequential sampling. *Simulation Modelling Practice and Theory*, 92:62–81, 2019.
- [18] Joshua M. Epstein and Robert Axtell. *Growing Artificial Societies. Social Science from the Bottom Up*. The Brookings Institution, Washington, DC, USA, 1996.
- [19] Stéphane Espié. Vehicle driven simulator versus traffic-driven simulator : the INRETS approach. In *Proceedings of Driving Simulation Conference (DSC'99)*, Paris, France, 1999.
- [20] Nabil T. Fadai, Ruth E. Baker, and Matthew J. Simpson. Accurate and efficient discretizations for stochastic models providing near agent-based spatial resolution at low computational cost. *Journal of The Royal Society Interface*, 16(159):20190421, 2019.
- [21] Gadi Fibich and Ro'i Gibori. Aggregate Diffusion Dynamics in Agent-Based Models with a Spatial Structure. *Operations Research*, 58(5):1450–1468, 2010.
- [22] Nigel Gilbert and Rosaria Conte. *Artificial Societies*. UCL Press, London, 1995.
- [23] Chad M. Glen, Melissa L. Kemp, and Eberhard O. Voit. Agent-based modeling of morphogenetic systems : Advantages and challenges. *PLOS Computational Biology*, 15(3):e1006577, 2019.
- [24] Thomas E. Gorochoowski. Agent-based modeling in synthetic biology. *Essays In Biochemistry*, 60(4):325–336, 2016.
- [25] Arnaud Grignard, Patrick Taillandier, Benoit Gaudou, Duc An Vo, Nghi Quang Huynh, and Alexis Drogoul. GAMA 1.6 : Advancing the Art of Complex Agent-Based Modeling and Simulation. In David Hutchison et al., editors, *PRIMA 2013 : Principles and Practice of Multi-Agent Systems*, volume 8291, pages 117–131. Springer, 2013.
- [26] Volker Grimm, Eloy Revilla, Uta Berger, Florian Jeltsch, Wolf M. Mooij, Steven F. Railsback, Hans-Hermann Thulke, Jakob Weiner, Thorsten Wiegand, and Donald L. DeAngelis. Pattern-Oriented Modeling of Agent-Based Complex Systems : Lessons from Ecology. *Science*, 310(5750):987–991, 2005.
- [27] Kirk Harland, Alison Heppenstall, Dianna Smith, and Mark Birkin. Creating Realistic Synthetic Populations at Varying Spatial Scales : A Comparative Critique of Population Synthesis Techniques. *Journal of Artificial Societies and Social Simulation*, 15(1):1, 2012.
- [28] Emmanuel Hermellin and Fabien Michel. GPU delegation : Toward a generic approach for developing MABS using GPU programming. In Catholijn M. Jonker et al., editors, *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pages 1249–1258. ACM, 2016.
- [29] Emmanuel Hermellin and Fabien Michel. Complex flocking dynamics without global stimulus. In Carole Knibbe et al., editors, *Proceedings of the 14th European Conference Artificial Life (ECAL)*, pages 513–520. MIT Press, 2017.
- [30] Arthur Hjorth, Bryan Head, Corey Brady, and Uri Wilensky. Levelspace : A netlogo extension for multi-level agent-based modeling. *Journal of Artificial Societies and Social Simulation*, 23(1):4, 2020.
- [31] Thomas J. Hladish, Carl A. B. Pearson, Kok Ben Toh, Diana Patricia Rojas, Pablo Manrique-Saide,

- Gonzalo M. Vazquez-Prokopec, M. Elizabeth Haloran, and Ira M. Longini. Designing effective control of dengue with combined interventions. *Proceedings of the National Academy of Sciences*, 117(6):3319–3325, 2020.
- [32] Yu-Lin Huang, Gildas Morvan, Frédéric Pichon, and David Mercier. SPSC : A New Execution Policy for Exploring Discrete-Time Stochastic Simulations. In Matteo Baldoni et al., editors, *PRIMA 2019 : Principles and Practice of Multi-Agent Systems*, volume 11873, pages 568–575. Springer, 2019.
- [33] Thomas Huriaux, Nicolas Sabouret, and Yvon Haradji. A Multi-level Model for Multi-agent based Simulation. In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence (ICAART)*, volume 2, pages 139–146. SCITEPRESS, 2014.
- [34] Matthew Hutson. AI shortcuts speed up simulations by billions of times. *Science*, 367(6479):728–728, 2020.
- [35] Joanna Juziuk, Danny Weyns, and Tom Holvoet. Design Patterns for Multi-agent Systems : A Systematic Literature Review. In Onn Shehory and Arnon Sturm, editors, *Agent-Oriented Software Engineering*, pages 79–99. Springer, 2014.
- [36] Mozghan Kabiri Chimeh, Peter Heywood, Marzio Pennisi, Francesco Pappalardo, and Paul Richmond. Parallelisation strategies for agent based simulation of immune systems. *BMC Bioinformatics*, 20(S6):579, December 2019.
- [37] Le-Minh Kieu, Nicolas Malleson, and Alison Heppenstall. Dealing with uncertainty in agent-based models for short-term predictions. *Royal Society Open Science*, 7(1):191074, 2020.
- [38] Yoann Kubera, Philippe Mathieu, and Sébastien Picault. IODA : An interaction-oriented approach for multi-agent based simulations. *Journal of Autonomous Agents and Multi-Agent Systems*, 23(3):303–343, 2011.
- [39] Van-Minh Le, Yann Chevalere, Ho Tuong Vinh, and Jean-Daniel Zucker. Hybrid of linear programming and genetic algorithm for optimizing agent-based simulation. Application to optimization of sign placement for tsunami evacuation. In *The 2015 IEEE RIVF International Conference on Computing & Communication Technologies (RIVF)*, pages 138–143, Can Tho, Vietnam, 2015. IEEE.
- [40] Demian Levis, Albert Diaz-Guilera, Ignacio Pagonabarraga, and Michele Starnini. Flocking and spreading dynamics in populations of self-propelled agents. *arXiv:1901.08831*, 2019.
- [41] Charlie Lin, Joshua Culver, Bronson Weston, Evan Underhill, Jonathan Gorky, and Prasad Dhurjati. Gut-Logo : Agent-based modeling framework to investigate spatial and temporal dynamics in the gut microbiome. *PLOS ONE*, 13(11):e0207072, 2018.
- [42] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. MASON : A Multiagent Simulation Environment. *Simulation*, 81(7):517–527, 2005.
- [43] Philippe Mathieu, Gildas Morvan, and Sébastien Picault. Multi-level agent-based simulations : Four design patterns. *Simulation Modelling Practice and Theory*, 83:51–64, 2018.
- [44] Philippe Mathieu and Sébastien Picault. An interaction-oriented model for multi-scale simulation. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 332–337. IJCAI/AAAI, 2011.
- [45] Philippe Mathieu and Sébastien Picault. From real purchase to realistic populations of simulated customers. In Yves Demazeau et al., editors, *Proceedings of the 11th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'2013)*, volume 7879 of *Lecture Notes in Computer Science*, pages 216–227. Springer, 2013.
- [46] Philippe Mathieu, Sébastien Picault, and Yann Secq. Design patterns for environments in multi-agent simulations. In Q. Chen et al., editors, *Proceedings of the 18th Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2015)*, volume 9387 of *LNCS*, pages 678–686. Springer, 2015.
- [47] Herman Melville. Bartleby, the scrivener : A story of wall street. *Putnam's Monthly Magazine*, 1853.
- [48] Nelson Minar, Roger Burkhart, Chris Langton, and Manor Askenazi. The SWARM simulation system : a toolkit for building multi-agent simulations. Working Paper 96-06-042, Santa Fe Institute, 1996.
- [49] Gildas Morvan. Multi-level agent-based modeling – bibliography. Work in progress, CoRR, 2013. arXiv:1205.0561 [cs.MA].
- [50] Gildas Morvan, Alexandre Veremme, and Daniel Dupont. IRM4MLS : The influence reaction model for multi-level simulation. In *Multi-Agent-Based Simulation XI*, volume 6532 of *LNCS*, pages 16–27. Springer Nature, 2011.
- [51] Ngoc Doanh Nguyen, Patrick Taillandier, Alexis Drogoul, and Pierre Auger. Inferring Equation-Based Models from Agent-Based Models : A Case Study in Competition Dynamics. In Nirmal Desai et al., editors, *Principles and Practice of Multi-Agent Systems*, volume 7057, pages 413–427. Springer, 2012.
- [52] Atsushi Niida, Takanori Hasegawa, and Satoru Miyano. Sensitivity analysis of agent-based simulation utilizing massively parallel computation and interactive data visualization. *PLOS ONE*, 14(3):e0210678, 2019.
- [53] Michael J North, Nicholson T Collier, Jonathan Ozik, Eric R Tatara, Charles M Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with Repast Symphony. *Complex Adaptive Systems Modeling*, 1(3), 2013.

- [54] Michael J. North, Nicholson T. Collier, and Jerry R. Vos. Experiences creating three implementations of the Repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulations*, 16(1):1–25, 2006.
- [55] Vladimir I. Oulianov. *Que faire ? Questions brûlantes de notre mouvement*. Seuil, traduit du russe (1966) edition, 1902.
- [56] Dawn C. Parker, Steven M. Manson, Marco A. Jansen, Matthew J. Hoffmann, and Peter Deadman. Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change : A Review. *Annals of the Association of American Geographers*, 93(2):314–337, 2003.
- [57] Jon Parker and Joshua M. Epstein. A distributed platform for global-scale agent-based models of disease transmission. *ACM Trans. Model. Comput. Simul.*, pages 2:1–2:25, 2011.
- [58] H. Van Dyke Parunak and Sven A. Brueckner. Software engineering for self-organizing systems. *The Knowledge Engineering Review*, 30(4):419–434, 2015.
- [59] Sébastien Picault, Yu-Lin Huang, Vianney Sicard, Sandie Arnoux, Gaël Beaunée, and Pauline Ezanno. EMULSION : Transparent and flexible multiscale stochastic models in human, animal and plant epidemiology. *PLoS Computational Biology*, 15(9):e1007342, 2019.
- [60] Nghi Quang Huynh, Tri Nguy en Huu, Arna ud Grignard, Hiep Xuan Huynh, and Alexis Drogoul. Coupling equation based models and agent-based models : example of a multi-strains and switch SIR toy model. *EAI Endorsed Transactions on Context-aware Systems and Applications*, 4(11):152334, 2017.
- [61] Craig W. Reynolds. Flocks, Herds and Schools : A Distributed Behavioral Model. *Computer Graphics*, 21(4):25–34, 1987.
- [62] Benjamin Roche, John M Drake, and Pejman Rohani. An Agent-Based Model to study the epidemiological and evolutionary dynamics of Influenza viruses. *BMC Bioinformatics*, 12(1):87, 2011.
- [63] Alban Rousset, Bénédicte Herrmann, Christophe Lang, and Laurent Philippe. A survey on parallel and distributed multi-agent systems for high performance computing simulations. *Computer Science Review*, 22:27–46, 2016.
- [64] David Servat. *Modélisation de dynamiques de flux par agents. Application aux processus de ruissellement, infiltration, et érosion*. Thèse de doctorat, Université Paris VI, 2000.
- [65] Wei Shao and Demetri Terzopoulos. Autonomous pedestrians. *Graphical Models*, 69(5–6):246–274, 2007.
- [66] Sauro Succi and Peter V. Coveney. Big data : the end of the scientific method? *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 377(2142), 2019.
- [67] Olivier Therond, Christophe Sibertin-Blanc, Romain Lardy, Benoit Gaudou, Maud Balestrat, Yi Hong, Thomas Louail, Van Bai Nguyen, David Panzoli, José-Miguel Sanchez, Sabine Sauvage, Patrick Taillandier, Maroussia Vavasseur, and Pierre Mazzeza. Integrated modelling of social-ecological systems : The MAELIA high-resolution multi-agent platform to deal with water scarcity problems. In *7th International Environmental Modelling and Software Society (iEMSs 2014)*, page 9, 2014.
- [68] Alessandro Troisi, Vance Wong, and Mark A. Ratner. An agent-based approach for modeling molecular self-organization. *Proceedings of the National Academy of Sciences (PNAS)*, 102(2):255–260, 2005.
- [69] H. Van Dyke Parunak. Interpreting digital pheromones as probability fields. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 1059–1068. IEEE, 2009.
- [70] Franck Varenne. Modèles et simulations : pluriformaliser, simuler, remathématiser. In *Modéliser & simuler. Épistémologies et pratiques de la modélisation et de la simulation*, volume 1, pages 297–326. Éditions Matériologiques, 2013.
- [71] Srinivasan Venkatramanan, Bryan Lewis, Jiangzhuo Chen, Dave Higdon, Anil Vullikanti, and Madhav Marathe. Using data-driven agent-based models for forecasting emerging infectious diseases. *Epidemics*, 22:43–49, 2018.
- [72] Uri Wilensky. NetLogo. plateforme de simulation, Center for Connected Learning and Computer-Based Modeling (CCL), Northwestern University, Evanston, IL, 1999.
- [73] David Wolinski, Stephen J. Guy, Anne-Hélène Olivier, Ming C. Lin, Dinesh Manocha, and Julien Pettré. Parameter estimation and comparative evaluation of crowd simulations : Parameter estimation and comparative evaluation of crowd simulations. *Computer Graphics Forum*, 33(2):303–312, 2014.
- [74] Levent Yilmaz and Tuncer I. Ören. Dynamic Model Updating in Simulation with Multimodels : A Taxonomy and a Generic Agent-Based Architecture. In *Proceedings of SCSC 2004 - Summer Computer Simulation Conference*, page 6, 2004.

Comportements hétérogènes de piétons et comportement émergent d'une foule

Anjara Nobby Rakotoarivelo, Jean-Michel Auberlet, Roland Brémond

Univ. Gustave Eiffel, COSYS/PICS-L, F-77447 Marne La Vallée, France

anjara.rakotoarivelo@univ-eiffel.fr, jean-michel.auberlet@univ-eiffel.fr,
roland.bremond@univ-eiffel.fr

Résumé

Dans cet article, nous proposons de simuler une foule uni-directionnelle composée de piétons aux comportements hétérogènes, et se déplaçant dans un couloir. Sous certaines conditions, nous montrons que les piétons adaptent leurs interactions avec leurs voisins, ce qui fait émerger un comportement de foule particulier : le phénomène de stop-and-go, phénomène très connu en trafic routier. Pour cela, nous avons utilisé un modèle à base d'agents pour la simulation de piétons, appelé ORCA (Optimal Reciprocal Collision Avoidance), et un modèle de file d'attente. Comme la plupart des modèles de simulation piétons, les agents du modèle ORCA utilisent tous le même raisonnement pour se déplacer, raisonnement qui repose sur le concept d'évitement de collisions. Nous avons modifié le modèle ORCA en introduisant un autre comportement de déplacement, le suivi, permettant ainsi de produire des comportements hétérogènes. En effet, un piéton peut avoir plusieurs objectifs dans son déplacement, éviter les autres, rester en groupe mais également en fonction du contexte suivre quelqu'un. Nous nous sommes concentrés sur les comportements d'évitement et de suivi. Dans certaines situations, un piéton détermine ses déplacements en fonction de ses voisins mais également par rapport à celui du piéton qui le précède. Ainsi nous avons testé plusieurs combinaisons entre comportements d'évitement et de comportements de suivi. Nous avons vérifié que nos modèles sont cohérents avec les propriétés d'une foule en utilisant le diagramme fondamental qui décrit la relation entre la densité et la vitesse moyenne d'un flux. Nos données ont été comparées entre elles, mais aussi avec des travaux antérieurs. L'un des modèles testés permet de faire émerger le phénomène de stop-and-go.

Mots-clés

Simulation de piétons, modèle mixte, hétérogénéité, évitement de collisions, file d'attente, émergence, stop-and-go.

1 Introduction

La simulation de foule est utilisée dans plusieurs domaines d'application tels que l'évacuation d'urgence, la gestion des foules, les grands événements sportifs ou religieux. Ces événements sont complexes et les mouvements de foule sont difficiles à prévoir; des accidents peuvent

arriver dans certaines conditions. La compréhension des comportements émergents de la foule pourrait permettre ainsi d'anticiper d'éventuelles catastrophes qui pourraient arriver. Cependant, reproduire ces comportements émergents reste un défi. Parmi ces comportements émergents, le "stop-and-go" est réputé comme le plus méconnu et le plus étudié, que ce soit avec des équations mathématiques [17] ou des méthodes heuristiques [14, 21]. Nous avons opté pour une modélisation microscopique contrairement aux modèles macroscopiques qui considèrent la foule comme un fluide, et une approche multi-agents, en permettant aux agents d'adapter leurs décisions à leur environnement, conduisant à des comportements observables hétérogènes au sein de la population simulée.

Dans la plupart des simulations microscopiques de foule, cette dernière est composée de piétons qui utilisent le même raisonnement et ont le même comportement, à savoir l'évitement de collisions. Pourtant selon les situations, les piétons peuvent se comporter de manière différente selon qu'ils souhaitent éviter, doubler, suivre quelqu'un ou rester en groupe. Cela pose la question de la mixité de comportements et donc de la mixité de modèles de comportements. Une hétérogénéité faible est souvent obtenue, dans les simulations, en variant les caractéristiques individuelles des agents (vitesses préférentielles, destinations, distances de perception, tailles, etc.), mais dans ce cas, les comportements suivent tous le même raisonnement, les mêmes règles, et ne permettent pas au piéton d'interagir avec un autre piéton différemment.

L'objectif de l'article est de produire des comportements hétérogènes de piétons dans un sens plus fort, avec des piétons utilisant des règles différentes de décision et d'adaptation au sein d'une même foule, et d'observer ainsi une hétérogénéité des comportements émergents. Pour avoir des comportements hétérogènes, nous avons ajouté des conditions qui permettent aux agents de basculer d'un comportement à un autre. Les types de comportement que nous avons implémentés sont l'évitement de collision et le suivi. En effet, sous certaines conditions, dans une foule, nous pouvons suivre un piéton qui nous précède sans nécessairement s'en rendre compte. Ainsi tous les agents piétons ne suivent pas tous le même raisonnement pour se déplacer, ils utilisent l'un des deux modes de raisonnement à leur

disposition, en fonction du contexte qu'ils rencontrent. La combinaison de ces deux comportements nous a permis de construire quatre modèles de comportement : en fonction des voisins perçus, les piétons basculent entre suivi et évitement ou le mélange des deux. Ces modèles mixtes ont été développés à partir du modèle ORCA [19] en y introduisant un modèle de file d'attente pour simuler le suivi d'un "leader" [13].

L'un des modèles mixtes que nous avons testé permet de faire émerger le phénomène de stop-and-go. Le stop-and-go est un phénomène observé à la fois dans les déplacements de foules et en trafic routier. Il se manifeste par des périodes successives, cycliques, d'accélération et de décélération, qui se propagent au niveau macroscopique comme des ondes en sens contraire du flux. Traduit au niveau microscopique, un piéton est contraint de décélérer puis d'accélérer parce que ceux qui sont devant lui en font autant. La perturbation va se propager à l'ensemble de la foule, et ne s'arrête que lorsque l'espace libre devant permet d'accélérer sans contrainte [22]. Cette situation est typiquement rencontrée dans des situations relativement denses et unidirectionnelles.

Pour valider ces quatre modèles, nous avons comparé les caractéristiques des foules simulées, en nous appuyant sur le diagramme fondamental. Il illustre la loi de conservation du trafic : le nombre de piétons qui entre dans un couloir doit être égal au nombre de piétons qui en sort. Ce diagramme exprime la relation fondamentale entre la vitesse moyenne V de la foule dans une zone donnée en fonction de la densité K dans cette zone. A faible densité, les piétons marchent à leur vitesse préférentielle, ce qui conduit à une vitesse moyenne du flux relativement élevée (plus de 1,4m/s pour des jeunes adultes). Lorsque la densité augmente, les piétons subissent davantage de contraintes et la vitesse moyenne de la foule va ainsi décroître, la vitesse moyenne d'une foule dépend de la vitesse des piétons les plus lents. Issu de la modélisation du trafic routier, le diagramme fondamental est largement utilisé dans les simulations de trafic piéton [17, 14, 15].

L'organisation de cet article est la suivante : la section 2 dresse un court panorama relatif à notre champ de recherche ; le modèle ORCA, le modèle de file d'attente et les modèles mixtes proposés sont décrits en section 3 ; les résultats des simulations sont présentés en section 4 ; les limites et les perspectives de notre étude viennent clore cet article.

2 Travaux antérieurs

L'objectif de l'article étant de produire des comportements hétérogènes afin d'observer l'apparition des comportements émergents. Dans cette partie, on va recenser uniquement les modèles microscopiques d'évitement de collision, les modèles qui permettent de manipuler individuellement chaque agent. Ensuite, on va identifier les travaux sur les modèles permettant d'avoir des comportements hétérogènes. Enfin, lister les comportements émer-

gents existants.

2.1 Évitement de collision

Le *Social Force Model* (SFM) [10] est un modèle de simulation utilisant une analogie électrostatique pour simuler le déplacement des piétons, avec des lois de répulsion et d'attraction. Les forces de répulsion permettent au piéton d'éviter les collisions avec des obstacles et d'autres piétons, tandis que les forces d'attraction les attirent vers leur destination ou leur permet de rester en groupe. Dans ce modèle, les piétons n'anticipent pas les mouvements des autres piétons. En effet, pour calculer son déplacement, l'agent piéton ne prend en compte que la position des autres mais pas leur vitesse, et ne peut anticiper le mouvement suivant des voisins. Dans une approche multi-agent, les agents peuvent alors choisir la même position lorsqu'ils se déplacent ; des risques de collision apparaissent ainsi. La plupart des recherches avec ce modèle utilise donc des pas de simulation faible, souvent de l'ordre de 1/10s, pour que l'agent rectifie assez vite sa trajectoire et évite d'éventuelles collisions.

Le modèle *BioCrowd* permet de transposer un algorithme de développement de branches d'arbre à la simulation de piétons [4]. L'espace est discrétisé avec un ensemble de marqueurs spatiaux, et les piétons ne peuvent se déplacer que d'un marqueur à un autre. Le choix d'un marqueur disponible dans son voisinage va lui permettre d'atteindre sa destination finale. Par construction, ce modèle de simulation garantit l'évitement de collisions. Toutefois, les agents ne pouvant se déplacer que d'un marqueur à un autre, leur vitesse dépend conjointement de la densité spatiale de marqueurs et de la fréquence de calcul de la simulation, rendant la calibration de ces deux paramètres peu aisée.

Le modèle *ORCA* est une extension du modèle *RVO* (*Reciprocal Velocity Obstacle*). Chaque agent calcule le vecteur vitesse admissible optimal (dans un certain sens) afin d'atteindre sa destination tout en évitant les collisions. L'agent du modèle *ORCA* calcule sa nouvelle vitesse en prenant en compte la position et la vitesse de tous ses voisins, dans son champ de perception. La prise en compte de la vitesse permet à l'agent d'anticiper la future position de ses voisins. Ce modèle garantit l'absence d'oscillations et de collisions sous réserve d'initialiser correctement les paramètres. Des oscillations peuvent apparaître dans une situation de face à face, quand chacun évite l'autre de manière exagérée dans un premier temps, et dans un second temps revient vers lui parce qu'il s'est trop écarté de son but, et s'écarte à nouveau, etc.

Toutefois, sur les 3 modèles présentés, les agents interagissent avec les autres agents de manière identique, rendant les simulations homogènes. Les paramètres individuels peuvent être différents, comme la vitesse préférentielle ou la taille ; par contre les agents dans chaque modèle utilisent le même raisonnement. Nous voulons travailler dans un espace continu et les modèles *ORCA* et *SFM* permettent de simuler dans un espace continu. Contrairement

au modèle SFM qui ne prend en compte que les positions des voisins, le modèle ORCA prend en compte aussi les vitesses des voisins, rendant possible l'anticipation et garantissant l'évitement de collision.

2.2 Comportements hétérogènes

Le modèle ORCA a été modifié par différents auteurs de manière à créer des comportements asymétriques lors des interactions [3, 16]. Dans le modèle ORCA original, chaque agent prend en compte les autres de la même manière, ce qui fait que l'évitement de collision est symétrique (par exemple, quand ils sont front face à face). La symétrie est obtenue parce qu'un paramètre, appelé *Shared Effort* (SE), vaut 0,5, ce qui signifie d'après les auteurs que chacun fait la moitié de l'effort d'évitement. Dans les évolutions proposées [3, 16], le SE varie d'un agent à l'autre. Plus le SE est élevé, plus l'agent fait d'effort pour éviter les collisions. Dans ces modèles, les agents respectent toujours les contraintes du modèle ORCA à savoir pas de collision et pas d'oscillation.

Bourgeois et al. [1] ont aussi redéfini le modèle SFM pour prendre en compte des comportements hétérogènes, en particulier deux comportements différents : « Normal » et « Dangereux ». Lorsqu'un piéton « Normal » rencontre un piéton « Dangereux », il accepte d'être plus proche des autres piétons « Normaux » pour davantage éviter le piéton « Dangereux ».

Dans ces trois articles [3, 1, 16], les auteurs ont enrichi les possibilités de comportement des agents pour créer une hétérogénéité au niveau local. Dans notre article, nous cherchons à produire des comportements hétérogènes observables au niveau global, pouvant dans certains cas faire émerger un comportement collectif.

Toutefois, sur ces modèles hétérogènes explorés, aucune piste n'a été explorée sur l'impact de la modification des comportements locaux sur les comportements globaux et les phénomènes émergents.

2.3 Comportement émergent

Certains chercheurs ont utilisé le modèle ORCA pour la simulation de foule à base d'agents autonomes [20, 8]. D'autres ont utilisé une approche multi-agent pour étudier les comportements émergents des agents à partir des mouvements locaux, qui sont la coopération ou la compétition entre les agents [2, 7]. L'idée est d'observer des comportements collectifs qui ne sont pas implémentés dans les agents mais émergent par interactions locales, comme la formation de files ou le stop-and-go. La formation de files est observée lorsqu'une foule se déplace dans deux sens opposés et s'auto-organise pour faciliter le mouvement de l'ensemble. Le stop-and-go par contre est observé lorsqu'un flux se déplace dans une seule direction. Des recherches ont démontré que le stop-and-go peut être un indicateur de situation critique [17, 14].

Dans le cas d'un flux unidirectionnel, on peut faire le lien entre comportements locaux et globaux au moyen du dia-

gramme fondamental (relation entre la vitesse moyenne et la densité d'un flux). Plus la densité augmente, plus la vitesse moyenne diminue, jusqu'au point de blocage où les agents ne peuvent presque plus avancer. Des auteurs ont proposé un modèle de comportement qui varie en fonction de la densité de piétons dans l'environnement proche en utilisant directement le diagramme fondamental [15]. Mais ces travaux posent la question de la validité, car les résultats sont souvent comparés au diagramme fondamental, qui est une entrée du modèle.

Les recherches menées par [21] ont montré que les paramètres individuels des piétons tels que la taille de l'espace personnel, la distance de sécurité et le nombre de voisins perçus conduisent à l'apparition de phénomènes émergents d'auto-organisation comme la formation de files. Tordeux et al. [18] ont essayé de reproduire le stop-and-go sur une foule de piétons à partir du calcul de la vitesse et de l'espace entre les piétons. Ils ont optimisé la vitesse des agents avec la méthode OV (*Optimal Velocity*). Il existe aussi des recherches qui étudient la relation entre la densité, la vitesse et les phénomènes émergents; Seyfried et al. [17] ont utilisé le diagramme fondamental pour comprendre les congestions.

Différents travaux antérieurs ont donc introduit des modèles hétérogènes; d'autres ont recherché, avec des modèles homogènes, l'émergence de comportements comme le stop-and-go. Dans cet article, on s'intéressera au comportement émergent d'une foule hétérogène.

3 Comportements proposés

Le modèle proposé dans cet article se base essentiellement sur deux comportements : le comportement d'évitement de collision et le comportement de suivi. Sous certaines conditions, les piétons basculent d'un comportement à l'autre, sous une forme pure ou sous une forme qui implique un compromis entre les deux types de comportements. Notre propre contribution dans cet article est la modélisation de 3 modèles d'interactions de ce type, que nous appellerons « mixtes ».

3.1 Comportement d'évitement

Nous avons choisi le modèle ORCA pour le comportement d'évitement : c'est en effet un modèle qui est limité à l'évitement de collision. Mais comme le modèle prend en compte en même temps la position et la vitesse des voisins, le piéton peut anticiper la position future de ses voisins, ce qui garantit l'apparition d'éventuelle collision. Le piéton est doté d'un paramètre d'anticipation de collision appelé `TimeHorizon`. Plus cette valeur est élevée, plus le piéton anticipe une collision à l'avance. Des travaux se sont intéressés à l'anticipation de collision de deux piétons qui se croisent sur un couloir étroit, en variant le `TimeHorizon` des piétons [5]. Dans le cadre de ce travail, tous les piétons ont le même temps d'anticipation, avec un `TimeHorizon` de 1 seconde.

3.2 Comportement de suivi

Lorsqu'on observe les piétons se déplacer dans un couloir unidirectionnel ou bidirectionnel, on voit apparaître une ligne imaginaire dans le couloir. Sur chaque ligne, on observe des piétons qui se suivent entre eux formant une sorte de queue ou de fil d'attente. Des expérimentations ont montré que la file d'attente observée dans les couloirs provoque le phénomène de stop-and-go lorsqu'on force les piétons à ne pas se dépasser entre eux [13]. Lemerrier et al. ont intégré un modèle de file d'attente dans une simulation de foule [12], qui permet de simuler le fait qu'un piéton soit influencé par un piéton qui le précède (que nous appellerons « leader »), si certaines conditions sont remplies. C'est à partir de ce dernier modèle que nous avons développés les modèles mixtes proposés.

3.3 Modèles mixtes proposés

Les piétons pendant leur déplacement peuvent adopter plusieurs comportements d'interactions. Dans cette contribution, nous intéressons aux comportements d'évitement de collision et de suivi. Nous avons combiné ces deux modes de comportement pour une foule unidirectionnelle dans un couloir. Les piétons peuvent basculer d'un comportement à l'autre ou le mélanger des deux. Comme dans le monde réel, le contexte, et donc la situation rencontrée, détermine les comportements à adopter.

Quand le piéton A se trouve derrière le piéton B, A le percute si B s'arrête et que A n'effectue aucune action correctrice en termes de vitesse ou de cap. Le piéton A se trouve alors dans une situation de suivi. Une autre définition de cette situation serait que le piéton A, dans une foule, est plus influencé par B que par ses autres voisins. C'est dans cette situation de contrainte que notre agent adapte son comportement. Nous avons testé plusieurs modèles : évitement pur, suivi, et combinaison des deux modes. Le modèle original ORCA a été également simulé, comme référence, pour faire des comparaisons.

Pour adapter son comportement, un piéton doit reconnaître qu'il est dans une situation de suivi et donc qu'il a un piéton singulier qui le précède (qui sera appelé « leader » par la suite). Pour identifier un leader, l'agent parcourt (à chaque pas de temps) la liste de ses voisins (c'est-à-dire ceux qu'il perçoit). Pour être leader un voisin doit satisfaire 4 conditions (ci-après). Si aucun ne vérifie ces conditions, aucun leader n'est identifié; s'il n'y en a qu'un, c'est lui le leader; s'il y en a plusieurs c'est qu'ils se trouvent à égale distance, l'agent choisit alors pour leader celui qui a l'ID le plus faible (un identifiant étant donné à chaque agent lors de l'initialisation de la simulation).

Les 4 conditions nécessaires pour identifier un leader sont les suivantes :

- L'agent et son voisin se déplacent dans la même direction (c'est toujours le cas dans notre étude avec une foule unidirectionnelle);
- Le leader potentiel doit se trouver à une distance longitudinale inférieure à D_x (voir figure 1). Cette

hypothèse trouve sa source dans le concept de proxémie introduit par Hall [9]. Il permet de distinguer chez l'homme une distance intime, une distance interpersonnelle.... Dans notre cas c'est la distance interpersonnelle que nous utilisons ;

- La distance latérale D_y (voir figure 1) entre le piéton et le leader potentiel doit être inférieure à la somme des rayons des deux piétons. Cette condition signifie que si le leader candidat s'arrête et que l'agent ne procède à aucune action correctrice, alors il le percute.
- L'angle formé par les vecteurs vitesse des deux piétons est inférieur à ϵ_α ; si la différence de cap est trop importante, l'agent considère que celui qui le précède ne sera bientôt plus devant lui, et que cette interaction sera de courte durée ;

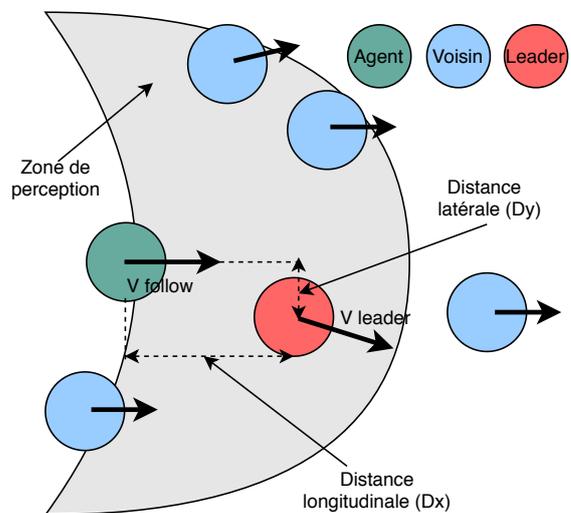


FIGURE 1 – Perception des voisins et identification d'un leader. Le champ de perception est délimité par la distance de perception. Sous certaines conditions, l'agent vert peut trouver un leader parmi ses voisins.

Nous avons modifié le modèle ORCA en introduisant la proposition suivante (voir figure 2) : pour chaque agent, à chaque pas de temps, si aucune situation de suivi n'est identifiée, absence de leader, alors on conserve la vitesse calculée par le modèle ORCA. Si un leader est identifié, l'agent appliquera l'accélération d'un des modèles proposés (Cf. ci-après).

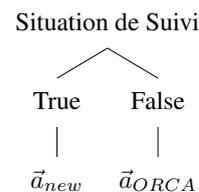


FIGURE 2 – Processus d'activation du modèle mixte et du calcul de la nouvelle accélération \vec{a}_{new} .

Pour chaque pas de simulation le modèle ORCA calcule pour un agent une nouvelle vitesse à partir de la position et de la vitesse de l'agent et de ses voisins. Le modèle de suivi calcule par contre une accélération (à partir de la position et de la vitesse de l'agent et du leader). Pour comparer ou mixer les deux modèles, nous devons calculer l'accélération du modèle ORCA, obtenue en dérivant la vitesse calculée.

Nous devons préciser ici qu'à chaque pas de temps, les agents calculent la vitesse produite par le modèle ORCA ; si un agent se trouve en situation de suivi, il pourra modifier cette vitesse en utilisant la nouvelle accélération calculée. Dans la continuité des travaux initiés dans [12], si l'agent modifie la vitesse calculée par ORCA, seule la composante longitudinale sera modifiée. Cette hypothèse très forte se justifie par le fait que le modèle ORCA calcule le meilleur cap possible pour atteindre la destination.

Dans la suite de cet article, nous proposons de tester 4 modèles de comportement pour les situations de suivi obtenues à partir de la combinaison du modèle ORCA et du modèle file d'attente. Le modèle 5 est le modèle ORCA initial, qui nous sert de référence.

Modèle 1 : c'est le modèle proposé par Lemercier & Auberlet [12]. L'accélération qui sert à calculer la nouvelle vitesse, est le minimum entre l'accélération produite par le modèle de suivi a_{follow_x} et l'accélération ORCA a_{ORCA_x} .

$$a_{new_x} \leftarrow \min(a_{follow_x}; a_{ORCA_x})$$

Prendre l'accélération minimale implique que l'agent veut satisfaire la contrainte la plus forte entre celle de ORCA qui vient de l'ensemble des voisins, et celle venant du leader.

Modèle 2 : dans cette modalité, nous faisons l'hypothèse que l'agent désire minimiser ses variations de vitesse quand il est dans une situation de suivi.

$$\text{Si } |a_{follow_x}| < |a_{ORCA_x}|$$

$$a_{new_x} \leftarrow a_{follow_x}$$

Si non

$$a_{new_x} \leftarrow a_{ORCA_x}$$

Autrement dit, les piétons modifient le moins possible leur vitesse actuelle (principe d'inertie).

Modèle 3 : dans cette modalité nous faisons l'hypothèse que l'agent entre en situation de suivi, et en sort le plus rapidement possible.

$$a_{new_x} \leftarrow \max(a_{follow_x}; a_{ORCA_x})$$

On peut dire que le piéton cherche à minimiser la durée d'interaction.

Modèle 4 : dans cette modalité, nous faisons l'hypothèse que l'agent souhaite être le plus possible en situation de suivi ; dit autrement, l'interaction avec le leader est plus important que les autres.

$$a_{new_x} \leftarrow a_{follow_x}$$

Les autres voisins ont donc une influence limitée sur l'agent.

Modèle ORCA : c'est la modalité de référence, qui servira pour effectuer les comparaisons.

$$a_{new_x} \leftarrow a_{ORCA_x}$$

4 Résultats des simulations

La plateforme de simulation que nous avons utilisée est le moteur de jeux vidéo UNITY 3D (unity.com), avec la librairie ORCA développée en C# [19].

4.1 Scénario

Lors de l'initialisation, les piétons sont placés aléatoirement dans un couloir de taille : 30m de longueur et 10m de largeur. Les vitesses préférentielles des piétons sont initialisées aléatoirement avec une loi de distribution normale ; les piétons sont initialisés avec leur vitesse préférentielle. Ils se déplacent tous dans le couloir dans le même sens, vers la droite. Arrivés au bout du couloir, les agents sont replacés au début du couloir, sans modifier les valeurs des autres variables (vitesse, position latérale).

Comme les modèles utilisés tiennent compte des voisins devant eux, nous avons ajouté des piétons virtuels en fin de couloir (Cf. [12]). Ainsi nous simulons un couloir circulaire qui permet de garder le même nombre de piétons tout au long de la simulation, et donc d'avoir une densité constante.

4.2 Paramètres

Plusieurs paramètres ont été utilisés pour l'ensemble des simulations : les paramètres agents qui décrivent le comportement individuel des agents ; les paramètres pour déterminer si le suivi est possible, et les paramètres de simulations. Nous donnons ci-dessous les valeurs de ces paramètres et les valeurs initiales des variables :

* Paramètres agent

- Vitesse préférentielle $\in \mathcal{N}(1,4m/s; 0,3m/s)$. La vitesse moyenne des piétons d'âge moyen étant de 1,4m/s [6, 11], nous avons utilisé un écart type de 0.3m/s pour avoir des piétons qui se déplacent à des vitesses différentes.

- Distance de perception = 10m

- Nombre maximal de voisins à considérer par un agent = 10

- Diamètre d'un piéton = 0,6m (disque non déformable)

* Paramètres pour identifier les possibilité de suivi

- $\epsilon_{\alpha f} = \pi/6$ (angle maximal entre les caps de l'agent et du leader)

- $D_x = 1,5m$ (distance maximale, de centre à centre, avec le leader)

* Paramètres de simulations

- Durée = 300s

- Nombre de piétons $\in \{100, 200, \dots, 600\}$

- Pas de simulation = 0,1s

4.3 Situations permettant le suivi

En observant si les agents sont dans des situations permettant le suivi, c'est-à-dire si un leader potentiel est identifié,

on peut comprendre si un modèle en particulier favorise la situation de suivi. C'est pourquoi nous avons comptabilisé la proportion d'agents ayant détecté un leader pour les 5 modèles simulés.

La figure 3 représente la proportion d'agents en situation permettant le suivi, en fonction de la densité dans le couloir, et pour chacun des modèles simulés. Nous pouvons voir que quelle que soit la densité, le modèle 1 favorise le suivi, tandis que le modèle 3 est celui dans lequel ces situations sont les plus rares. Cela s'explique peut-être par le fait qu'en choisissant l'accélération maximale, sans modifier la composante latérale, rend plus difficile la condition 4 (angle entre les caps de l'agent et du leader) pour identifier un leader. ORCA est proche du modèle 3 à faible densité, et du modèle 1 pour des densités plus élevées, pour lesquelles les agents identifient presque toujours un leader (la proportion est proche de 100% alors que pour les autres modèles, elle est en dessous de 90%). Globalement, la proportion augmente avec la densité, ce qui est logique, mais il y a une exception avec le modèle 4, ce que nous n'expliquons pas à ce stade de nos travaux.

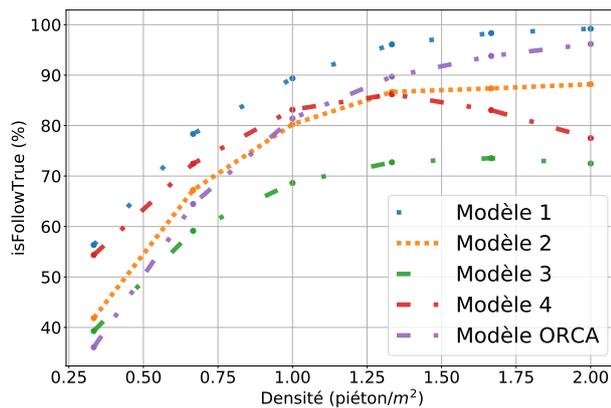


FIGURE 3 – Proportion d'agents en situation permettant le suivi (existence d'un leader potentiel) en fonction de la densité dans le couloir.

On voit Fig. 3 que pour une densité donnée, le choix d'un modèle a un impact sur la proportion de situations dans laquelle le suivi est possible. Les règles d'interaction ont donc un impact sur la possibilité d'identifier un leader.

Nous pouvons donc affirmer que les modèles proposés favorisent l'hétérogénéité de comportements, puisque à l'exception du modèle 1 pour des fortes densités, seule une partie des piétons a la possibilité d'identifier un « leader », et donc de modifier son accélération en tenant compte de ce leader. Nous avons ainsi deux classes de populations, et pour l'une d'entre elles, selon les modèles (modèle 1 et modèle 3), elle peut se subdiviser entre deux sous classes, favorisant l'observation de comportements hétérogènes.

4.4 Évolution de l'utilisation de \vec{a}_{follow}

En situation de suivi, nous calculons pour chaque agent deux accélérations : \vec{a}_{follow} et \vec{a}_{ORCA} . En fonction des modalités utilisées, une accélération entre les deux est utilisée. La figure 4 montre la proportion d'agents, ayant identifié un leader, qui utilisent l'accélération \vec{a}_{follow} en fonction de la densité dans le couloir, et pour chaque modèle de comportement proposé. Avec le modèle 1, quelque soit la densité, l'agent utilise dans 90% des cas, l'accélération \vec{a}_{follow} . Pour les modèles 2 et 3, on observe la même tendance : une augmentation à faible densité, puis une stabilisation autour de 1,25 piéton/m². Globalement, l'accélération \vec{a}_{follow} est plus utilisée que \vec{a}_{ORCA} à partir de 1 piéton/m² (sauf bien entendu pour le modèle ORCA, qui ne l'utilise jamais). Pour le modèle 4, l'agent utilise systématiquement \vec{a}_{follow} , nous avons donc un taux de 100%.

Nous pouvons en conclure que les modèles 2, 3 et 4 sont les plus hétérogènes, dans la mesure où dans le couloir, coexistent des sous-populations qui utilisent, les uns \vec{a}_{ORCA} , les autres \vec{a}_{follow} .

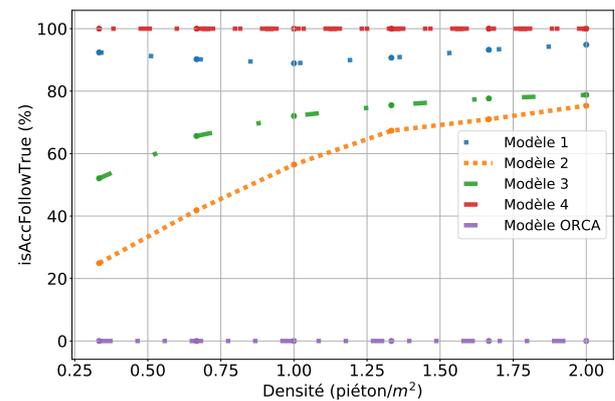
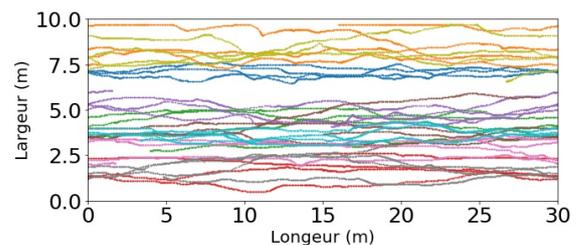


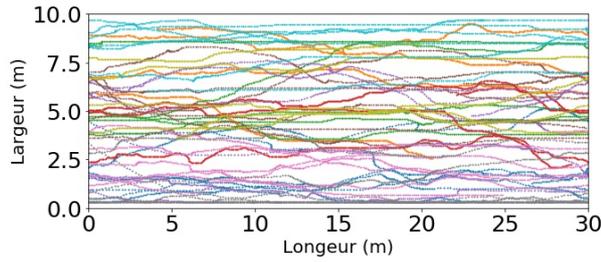
FIGURE 4 – Pourcentage d'agents qui utilisent \vec{a}_{follow} parmi ceux qui suivent un leader.

4.5 Trajectoires

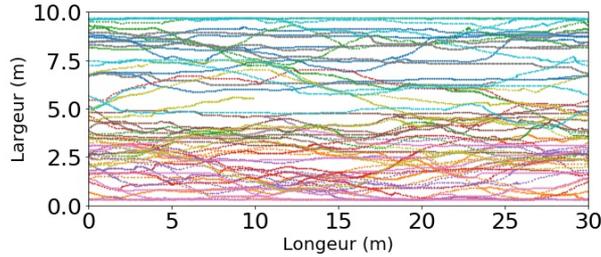
Les traces des trajectoires des piétons est différente selon le modèle qu'on utilise. Pendant nos simulations, avec une densité de 1.33 piétons/m² (400 piétons), nous avons remarqué une similarité des traces pour les modèle 2 et 4 avec celui du modèle de référence ORCA (figure 5c, 20 trajectoires tirées au hasard). On observe plus de différence entre modèles 1 figure 5a et 3 figure 5b.



(a) Traces des trajectoires obtenues par le modèle 1.



(b) Traces des trajectoires obtenues par le modèle 3.



(c) Traces des trajectoires obtenues par le modèle ORCA.

FIGURE 5 – Traces des trajectoires de 20 piétons tirés au hasard sur 400 au total, pour une durée de 50 secondes de simulation.

Pour le modèle 1 dans lequel l'agent utilise l'accélération minimale entre \vec{a}_{ORCA} et \vec{a}_{follow} , les piétons ont tendance à conserver leur cap, limitant les variations latérales qui permettent de se faufiler. Par contre, avec le modèle 3, les piétons ont plus tendance à se faufiler. Les modèles étudiés semblent donc avoir une influence sur la stabilité du flux en termes de changement d'orientation.

4.6 Émergence et comportement émergent

Pour caractériser les flux de piétons, les indicateurs de densité, de vitesse moyenne et de débit sont couramment utilisés. Des relations physiques les relient entre eux. Le diagramme fondamental permet d'illustrer ces indicateurs, en donnant la relation vitesse-densité. De plus, pour capturer l'apparition du phénomène de stop-and-go, nous analyserons l'évolution de la vitesse moyenne dans le temps, dans une zone donnée du couloir.

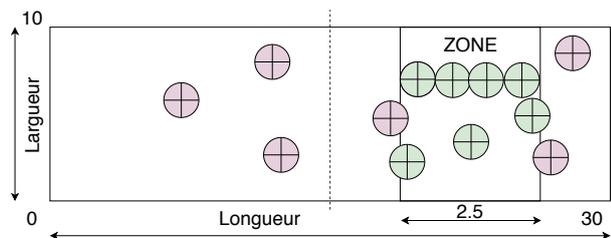


FIGURE 6 – Zone de mesure pour la vitesse moyenne et la densité de la foule. Les piétons rouges sont en dehors de la zone, les verts sont à l'intérieur.

4.6.1 Zone de mesure

Le couloir mesure 30m de longueur et 10m de largeur. Nous avons calculé la vitesse moyenne des piétons dans une zone du couloir de 2.5m de longueur (approximativement 4 piétons en file indienne) sur toute la largeur du couloir. Cette zone se trouve entre 17.5m et 20m par rapport à l'entrée du couloir (Cf. Fig. 6). Elle a été choisie de manière à éviter les potentiels effets de bord au début et à la fin du couloir.

4.6.2 Diagramme fondamental

La forme des diagrammes fondamentaux figure 7 indique pour tous les modèles la corrélation attendue entre vitesse et densité. On retrouve aussi les caractéristiques des diagrammes fondamentaux, c'est-à-dire une vitesse moyenne assez élevée au début de la courbe (lorsque la densité est faible), ensuite une diminution progressive de la vitesse.

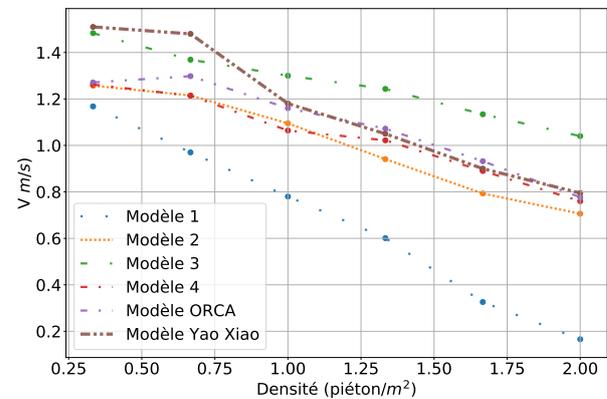


FIGURE 7 – Diagramme fondamental qui illustre la relation entre la vitesse moyenne et la densité de piétons, pour les 5 modèles simulés et pour le modèle de Yao Xiao [21].

Lorsque les agents qui sont en mode « suivi », utilisent l'accélération minimale (le modèle 1), la vitesse moyenne de la population est ralentie par rapport aux autres modèles, quelle que soit la densité. À partir de 2 piétons/m², le modèle 1 est bloqué alors que dans les autres modèles, les piétons arrivent encore à se déplacer. À partir de 1 piéton/m², les modèles 2, 4, ORCA et le modèle Yao Xiao ont des allures similaires. La vitesse des piétons, avec le modèle 3 diminue plus lentement (avec la densité) qu'avec les autres modèles. Cela s'explique du fait que les agents utilisent l'accélération maximale. Pour autant nos modèles et le modèle Yao Xiao ont les mêmes caractéristiques générales : la présence de comportements hétérogènes ne modifie pas la forme générale du diagramme fondamental, tout en favorisant une meilleure représentativité des comportements.

On remarquera que dans cette étude, il n'était pas possible de simuler des densités supérieures à 2,0 piétons/m² due à la taille des piétons (0,6m) et de la représentation par des disques indéformables.

4.6.3 Stop-and-go

Nous avons calculé la vitesse moyenne des piétons dans la zone de mesure à chaque pas de temps, pour toutes nos simulations. Le phénomène de stop-and-go est visible sur la figure 8 : on observe des « phases » dans le profil de vitesse moyenne en fonction du temps, dans cette zone. Ce résultat est obtenu uniquement avec le modèle 1, ici pour une densité de $1,33 \text{ piétons}/m^2$ (400 piétons). Ce phénomène n'apparaît dans aucun des autres modèles testés, y compris le modèle 4 qui dans les files d'attente, reproduit du stop-and-go [13]. On aurait pu penser un modèle qui n'utilise que l'accélération \vec{a}_{follow} pendant le suivi exhiberait le phénomène.

Nos simulations étant limitées à une densité maximale de $2,0 \text{ piétons}/m^2$, il est toutefois possible que des stop-and-go puissent apparaître pour des densités plus élevées.

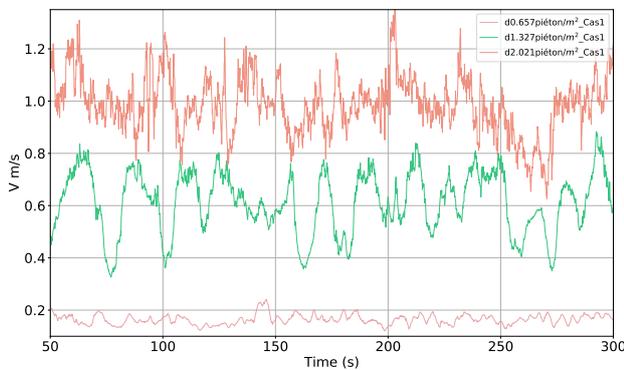


FIGURE 8 – Apparition du phénomène de stop-and-go sur la zone de mesure, avec une densité de $1,33 \text{ piétons}/m^2$.

Nos résultats avec le modèle 1 sont cohérents avec ceux de Moussaid et al. [14], qui démontrent que le phénomène de stop-and-go apparaît lorsque l'espace est couvert entre 40 et 65%. Dans nos résultats, avec 400 piétons de $0,6m$ de diamètre ($1,33 \text{ piétons}/m^2$), dans un espace de $300m^2$, on obtient un taux de couverture de 37,68%. Nos simulations ont montré qu'à 500 piétons ($1,66 \text{ piétons}/m^2$), avec un taux de couverture de 47%, le phénomène est toujours présent mais avec une plus faible amplitude pour les phases de décélération et d'accélération du flux. Avec 600 piétons ($2 \text{ piétons}/m^2$) et un taux de recouvrement de 56,52%, le phénomène disparaît complètement. D'autres simulations seraient nécessaires pour mieux comprendre le phénomène.

5 Conclusion

Dans ces travaux, nous nous sommes intéressés à deux concepts : l'hétérogénéité des comportements de déplacement piétons et le comportement émergent de la foule. Différentes approches existent pour améliorer l'hétérogénéité des comportements, comme l'ajout de nouvelles règles, de conditions ou de paramètres ; nous avons opté pour la combinaison de modèles de comportements, qui sont activés en fonction de l'environnement local du piéton. Différentes

approches existent aussi pour faire apparaître des comportements émergents, comme l'utilisation d'équations mathématiques ou de méthodes heuristiques ; nous avons utilisé une approche à base d'agents pour faire émerger des comportements collectifs qui ne sont pas explicitement décrits dans les modèles de comportement individuel.

L'utilisation d'un modèle mixte, qui calcule deux valeurs de l'accélération, nous a permis de proposer plusieurs modèles de comportement, selon la manière de combiner ces accélérations. L'un de ces modèles a permis de faire apparaître le phénomène de stop-and-go. Toutes les combinaisons n'ont pas été explorées, qui pourraient produire d'autres comportements émergents. Du fait de la représentation des piétons par des disques indéformable, la densités lors de nos simulations était limitée à $2 \text{ piétons}/m^2$. Modéliser les piétons par des ellipses ou par des surfaces déformables, en supposant que les piétons pourraient accepter une plus grande proximité, permettrait de simuler des densités plus élevées.

Les modèles 1, 2, 3 et 4 présentés dans cet article sont mixtes, en ce sens qu'à chaque instant, une partie des piétons utilise une accélération tandis que l'autre partie utilise l'autre. En fonction des conditions locales, chaque agent bascule d'une règle vers l'autre ou le mélange des deux. Une perspective serait de tenir également compte de la densité locale de piétons pour basculer le modèle de l'un vers l'autre.

Dans de futurs travaux, nous pourrions aussi envisager de mixer les modèles étudiés dans cette contribution, à savoir que chaque agent pourrait choisir un modèle en fonction de la situation rencontrée, favorisant ainsi une plus grande hétérogénéité.

Par ailleurs, le phénomène de stop-and-go peut être utilisé comme indicateur d'alerte avant une catastrophe [17]. Il serait donc utile d'être capable d'anticiper l'arrivée du phénomène, et d'identifier les indicateurs qui permettent de donner en avance un signe d'apparition du stop-and-go. Une analyse des propriétés dynamiques de la foule, telles que l'amplitude et la périodicité du stop-and-go, pourrait nous aider à mieux comprendre le phénomène.

Enfin il serait intéressant de réaliser une analyse de sensibilité des paramètres liés à l'identification d'un leader, en particulier la distance entre les piétons. Varier les paramètres permettrait-il d'influencer l'identification d'un leader, et donc du déclenchement du suivi ?

Références

- [1] Laure Bourgois, Haifa Rabai, and Jean-Michel Auberlet. Adaptation dynamique du comportement : vers un modèle guidé par la perception. In *RFIA 2012 (Reconnaissance des Formes et Intelligence Artificielle)*, pages 978–2–9539515–2–3, Lyon, France, January 2012. Session "Posters".
- [2] Francesca Camillen, Salvatore Capri, Cesare Garofalo, Matteo Ignaccolo, Giuseppe Inturri, Alessandro

- Pluchino, Andrea Rapisarda, and Salvatore Tudisco. Multi agent simulation of pedestrian behavior in closed spatial environments. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, pages 375–380. IEEE, 2009.
- [3] Sean Curtis, Basim Zafar, Adnan Gutub, and Dinesh Manocha. Right of way. *The Visual Computer*, 29(12) :1277–1292, 2013.
- [4] Alessandro de Lima Bicho, Rafael Araújo Rodrigues, Soraia Raupp Musse, Cláudio Rosito Jung, Marcelo Paravisi, and Léo Pini Magalhães. Simulating crowds based on a space colonization algorithm. *Computers & Graphics*, 36(2) :70–79, 2012.
- [5] Cleo Deroo, Angélique Montuwy, Beatrice Degraeve, Jean-Michel Auberlet, Anne-Helene Olivier, and Marie-Axelle Granie. Pedestrian collision avoidance on narrow sidewalk : A meeting between psychology and virtual reality. Technical report, 2019.
- [6] Kay Fitzpatrick, Marcus A Brewer, and Shawn Turner. Another look at pedestrian walking speed. *Transportation research record*, 1982(1) :21–29, 2006.
- [7] Jonathan ME Gabbai, Hujun Yin, WA Wright, and Nigel M Allinson. Self-organization, emergence and multi-agent systems. In *2005 International Conference on Neural Networks and Brain*, volume 3, pages 1824–1863. IEEE, 2005.
- [8] Stephen J Guy, Jatin Chhugani, Changkyu Kim, Nadathur Satish, Ming Lin, Dinesh Manocha, and Pradeep Dubey. Clearpath : highly parallel collision avoidance for multi-agent simulation. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 177–187, 2009.
- [9] T. S. Hall. *The hidden dimension*. Doubleday & Co., 1966.
- [10] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5) :4282, 1995.
- [11] Richard L Knoblauch, Martin T Pietrucha, and Marsha Nitzburg. Field studies of pedestrian walking speed and start-up time. *Transportation research record*, 1538(1) :27–38, 1996.
- [12] Samuel Lemerrier and Jean-Michel Auberlet. Towards more behaviours in crowd simulation. *Computer Animation and Virtual Worlds*, 27(1) :24–34, 2016.
- [13] Samuel Lemerrier, Asja Jelic, Richard Kulpa, Jiale Hua, Jérôme Fehrenbach, Pierre Degond, Cécile Appert-Rolland, Stéphane Donikian, and Julien Petré. Realistic following behaviors for crowd simulation. In *Computer Graphics Forum*, volume 31, pages 489–498. Wiley Online Library, 2012.
- [14] Mehdi Moussaïd, Dirk Helbing, and Guy Theraulaz. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, 108(17) :6884–6888, 2011.
- [15] Sahil Narang, Andrew Best, Sean Curtis, and Dinesh Manocha. Generating pedestrian trajectories consistent with the fundamental diagram based on physiological and psychological factors. *PLoS one*, 10(4), 2015.
- [16] Anjara Nobby Rakotoarivelo, Jean-Michel Auberlet, and Roland Brémond. Heterogeneous pedestrians behaviors through asymmetrical interaction. In *Proceedings of the 32nd International Conference on Computer Animation and Social Agents*, pages 25–28, 2019.
- [17] Armin Seyfried, Andrea Portz, and Andreas Schadschneider. Phase coexistence in congested states of pedestrian dynamics. In *International Conference on Cellular Automata*, pages 496–505. Springer, 2010.
- [18] Antoine Tordeux and Andreas Schadschneider. White and relaxed noises in optimal velocity models for pedestrian flow with stop-and-go waves. *Journal of Physics A : Mathematical and Theoretical*, 49(18) :185101, 2016.
- [19] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Optimal reciprocal collision avoidance for multi-agent navigation. In *Proc. of the IEEE International Conference on Robotics and Automation, Anchorage (AK), USA*, 2010.
- [20] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE, 2008.
- [21] Yao Xiao, Ziyou Gao, Yunchao Qu, and Xingang Li. A pedestrian flow model considering the impact of local density : Voronoi diagram based heuristics approach. *Transportation research part C : emerging technologies*, 68 :566–580, 2016.
- [22] Hwasoo Yeo and Alexander Skabardonis. Understanding stop-and-go traffic in view of asymmetric traffic theory. In *Transportation and Traffic Theory 2009 : Golden Jubilee*, pages 99–115. Springer, 2009.

Politiques de synchronisation dans les systèmes multi-agents distribués parallèles

A. Rousset¹, P. Breugnot², B. Herrmann², C. Lang² et L. Philippe²

¹ Faculté des Sciences, de la Technologie et de Médecine (FSTM),
Université du Luxembourg, Luxembourg

² Institut Femto-ST, Université de Bourgogne-Franche-Comté/CNRS, France

christophe.lang@univ-fcomte.fr

Résumé

Parmi les méthodes de modélisation/simulation, les systèmes multi-agents présentent un intérêt particulier pour simuler les systèmes complexes. Lorsque la taille des modèles croît, le recours aux systèmes multi-agents parallèles est nécessaire mais pose de nombreux problèmes. Dans cet article, nous nous intéressons à l'impact de la synchronisation sur la définition des modèles et leur exécution. Nous mettons en évidence des problématiques de synchronisation à travers des instances de modèles puis nous analysons expérimentalement l'impact des politiques de synchronisation sur des exécutions de grande taille.

Mots-clés

multi-agent simulation, parallélisme, MAS, High Performance Computing, synchronisation

Abstract

Among simulation or modelisation methods, multi-agent systems are interesting candidate to simulate complex systems. As the size of the models increases, the use of parallel multi-agent systems is mandatory but comes with many issues. In this article, we are interested in the impact of synchronization on models design and on their execution. We highlight synchronization problems through model instances then we experimentally analyze the impact of synchronization on large scale testcases.

Keywords

Multi-agent simulation, Parallelism, High Performance Computing, Synchronization

1 Introduction

La simulation numérique est devenue le troisième pilier de la science en tant qu'étape de validation de la théorie, déterminante pour le passage à l'expérimentation. Elle vise à virtualiser le monde réel, à en reproduire les comportements, par exemple pour explorer son évolution dans différentes configurations ou pour comprendre comment le contrôler. Dans les systèmes complexes, plusieurs phénomènes peuvent ainsi être étudiés simultanément mais

les comportements sont souvent trop élaborés et interdépendants pour pouvoir être modélisés par une loi unique. Les systèmes multi-agents sont alors souvent utilisés pour modéliser les comportements dynamiques des entités qui composent le système car ils reposent sur une description algorithmique simple d'agents qui interagissent entre eux. De nombreuses plates-formes [13, 7, 12] proposent un environnement de développement pour de tels modèles.

La qualité d'une simulation dépend bien souvent de la taille et de la précision du modèle. Or l'accroissement de la taille du modèle et de sa précision entraîne, de fait, une augmentation du nombre de calculs réalisés et rend nécessaire le recours à des exécutions parallèles, voire à l'utilisation de moyens de calcul haute performance (HPC : High Performance Computing). Si la simulation sur un seul ordinateur est souvent complexe, l'exécution distribuée parallèle d'une simulation est un vrai enjeu car elle pose de nombreux problèmes comme la distribution de l'environnement, la communication entre les instances parallèles de la plateforme, etc. Il existe des plateformes multi-agents (Parallel and Distributed MAS ou PDMAS) qui prennent en charge tout ou partie de l'exécution distribuée. Plusieurs instances, ou processus, de la plateforme coopèrent pour faciliter la mise en œuvre d'un modèle sur un ensemble d'ordinateurs ou au sein d'un cluster mais la synchronisation reste l'un des points clefs pour l'exécution efficace d'une simulation multi-agents parallèle du fait des nombreux échanges et dépendances temporelles qu'elle induit.

Nous nous intéressons dans cet article aux problèmes posés par la synchronisation au sein de simulations multi-agents distribuées parallèles, en visant plus particulièrement les exécutions à large échelle. Comme les plates-formes multi-agents fonctionnent fréquemment par pas de temps, il est nécessaire de s'interroger sur la manière d'échanger les données entre les processus distribués d'une même simulation au regard de ce mode de fonctionnement.

Les contributions proposées dans cet article sont, d'une part, la définition de politiques de synchronisation qui peuvent être utilisées au sein de simulations multi-agents parallèles et leur mise en évidence à travers des instances de modèles, et d'autre part, l'analyse expérimentale de l'impact des politiques de synchronisation sur des exécutions de grande

taille, jusqu'à 512 cœurs, dans les systèmes multi-agents parallèles et distribués.

L'article est organisé comme suit. Dans la section 2 nous présentons un état de l'art sur les systèmes multi-agents parallèles et la synchronisation. Nous proposons une étude des problématiques de synchronisation dans les systèmes multi-agents parallèles distribués dans la section 3. Puis, dans la section 4, avant de conclure, nous présentons une étude expérimentale de l'impact de différentes politiques de synchronisation sur l'exécution de trois modèles agents.

2 PDMAS et synchronisation

Par nature les agents interagissent entre eux, soit directement en consultant les données d'autres agents, soit indirectement, à travers les modifications réalisées sur l'environnement ou par échange de messages. Dans un PDMAS, les agents du modèle sont distribués entre les instances de la plateforme, ou processus, qui prennent en charge leur animation. De la même manière les données de l'environnement sont distribuées entre les processus. Un agent peut avoir à interagir avec des agents ou utiliser des données de l'environnement situés sur un autre processus. Ces accès nécessitent donc la mise en place d'une synchronisation entre les processus pour maintenir un état cohérent et permettre aux agents d'accéder à la valeur à jour des données avec l'objectif que la simulation parallèle donne le même résultat que la simulation séquentielle.

Les processus étant distribués, la synchronisation des données est réalisée sous la forme de communications. L'accès à ces données est alors plus coûteux qu'un simple accès local et il est important d'implémenter la synchronisation au plus juste, de manière à limiter les surcoûts. Cette implémentation suppose de caractériser les propriétés attendues. Dans la littérature, la caractérisation de la synchronisation de modèles parallèles repose sur le mode de dépendance de données et son respect de la causalité.

2.1 Modes de dépendance de données

On distingue deux modes de gestion de la dépendance de données dans les simulations numériques : (i) le mode *ghost* qui utilise les données calculées au pas de temps précédent (le *ghost*) et (ii) le mode *non ghost* qui n'utilise qu'une seule instance des données, dans laquelle les résultats des calculs sont directement reportés. Le *ghost* n'est accessible qu'en lecture alors que, sans *ghost*, les données sont accessibles en lecture et en écriture et l'ordre d'accès aux données impacte les résultats. Dans le cas des PDMAS, le modèle *ghost* permet de ne diffuser les données modifiées qu'à la fin des pas de temps. À noter que, pour limiter le coût de ces mises à jour, plusieurs PDMAS utilisent des zones de recouvrement, copies locales des données distantes limitées à la zone de perception des agents. Le mode *non ghost* nécessite la mise en place de mécanismes de mise à jour continus (au sein d'un même pas de temps) garantissant que les données utilisées lors d'interactions entre processus différents sont les données les plus à jour. Ces mises à jour doivent alors être synchronisées pour éviter les conflits. Nous analysons en 3.1 l'incidence du mode de dépendance de données sur

la synchronisation dans les modèles multi-agents et nous montrons que le choix du mode est déterminé par le modèle.

2.2 Respect de la causalité

Pour optimiser l'exécution parallèle de simulations, les systèmes à événements discrets autorisent parfois les processus à s'affranchir de la causalité. Deux approches de synchronisation sont ainsi définies, l'approche conservatrice et l'approche optimiste. Dans l'approche conservatrice (ou pessimiste), lorsqu'un processus traite un événement de date T , pour respecter la causalité, il doit être sûr de ne pas utiliser de données qui pourraient être modifiées ultérieurement avec une date $T' < T$. Dans cette approche de synchronisation il est nécessaire de déterminer les événements dont les données sont à jour. Les premiers algorithmes utilisant l'approche conservatrice ont été proposés par Chandy et al. [2]. Cette approche limite l'exploitation du parallélisme d'un modèle.

Contrairement à l'approche conservatrice, la violation de la contrainte de causalité est possible avec l'approche optimiste. Dans cette approche, chaque processus traite les événements dont il a connaissance. Cette connaissance des événements à traiter étant locale, et donc partielle, elle peut impliquer l'omission de certains événements provenant d'autres processus et donc ne pas respecter la causalité [11]. Lorsque le processus reçoit une donnée dont la date est antérieure à la date du processus, un retour en arrière est effectué par des mécanismes de *Rollback* [8] qui impliquent de sauvegarder plusieurs points de récupération par processus. Ainsi Xu et al. [17] définissent le *lookahead* comme étant la durée jusqu'au prochain pas de temps auquel il faudra mettre à jour les données. Ceci laisse la possibilité de continuer un processus tant qu'il n'est pas arrivé à son *lookahead*. À noter que cette approche est difficilement généralisable à tous les modèles et qu'elle n'est pas efficace lorsque le temps d'exécution passé à effectuer des *Rollback* devient très important, un *Rollback* pouvant entraîner une réaction en chaîne de *Rollback*.

Dans le cas des PDMAS, les événements sont planifiés tous les pas de temps. L'utilisation des approches conservatrices ou optimistes peut s'entendre comme le respect strict ou non de la frontière du pas de temps, comme nous l'étudions dans la section 3.

2.3 La synchronisation dans les PDMAS

Dans [9] nous avons proposé une étude des plateformes multi-agents parallèles et distribuées (PDMAS). Parmi les plateformes que nous avons évaluées, seules quatre d'entre elles permettent d'envisager une exécution sur des ressources de grande taille, de type HPC.

D-Mason. La plateforme D-Mason [5] implémente des mécanismes de synchronisation conservatifs et le mode *ghost* de gestion de dépendance des données. Pour réaliser une synchronisation conservatrice, chaque pas de temps est divisé en deux étapes : (1) la communication et synchronisation et (2) l'exécution de la simulation. Il y a donc une barrière de synchronisation à chaque pas de temps. Les agents d'une cellule (partition) c ne peuvent pas exécuter

le pas de temps i tant que les cellules voisines n'ont pas terminé d'exécuter le pas de temps $i - 1$. A la fin d'un pas de temps, chaque cellule envoie aux cellules voisines les informations concernant les agents qui se situent dans la zone de recouvrement ou les agents qui doivent être migrés d'un processus à un autre. Pour le pas de temps i les comportements de la cellule c sont ainsi calculés à partir du *ghost* des cellules voisines.

RepastHPC. Pour gérer le partage de données, la plateforme RepastHPC propose au programmeur de faire une copie, sur les processus distants, des agents susceptibles d'y être utilisés. La synchronisation entre les processus est effectuée uniquement dans quatre cas [4] : (i) lorsqu'un processus a besoin d'une copie d'agents d'un autre processus, pour maintenir la simulation dans un état cohérent, (ii) quand un processus possède un agent copie d'un autre processus, pour mettre à jour les copies de l'agent original, (iii) lorsque les zones de recouvrement d'une grille doivent être mises à jour, à chaque fin de pas de temps, (iv) quand un agent doit être migré d'un processus à un autre. Avec les outils de la plateforme, les programmeurs ont à développer un ensemble de méthodes qui définissent les informations nécessaires devant être synchronisées dans les agents. La plateforme permet d'implémenter les deux modes de dépendances de données car la mise à jour d'un agent copie peut-être soit périodique soit à la consultation de la copie. En couplant ces modes avec les fonctions d'ordonnement de la plateforme, il est possible d'implémenter les politiques conservatives et optimistes mais l'ensemble reste à la charge du développeur avec le risque de surcharger le modèle.

Flame. Avec Flame, tous les échanges entre les agents se font par messages, la synchronisation conservative repose donc sur celle de tableaux de messages [3]. Elle est effectuée en deux étapes : la demande de synchronisation puis l'exécution de la synchronisation. Dans un premier temps, lorsqu'un processus a terminé d'exécuter ses agents, il verrouille son tableau de messages et envoie aux autres processus une demande de synchronisation. Après cette étape, il est encore possible de faire des actions qui ne nécessitent pas l'utilisation du tableau de messages. Lorsque tous les processus ont verrouillé leurs tableaux de messages, une seconde étape d'exécution de la synchronisation est effectuée par échange de messages entre les tableaux. Après ces deux étapes, les tableaux de messages sont débloqués et la simulation se poursuit. Puisqu'il n'y a pas de modification des données entre deux synchronisations des tableaux de messages, la plateforme Flame repose sur un mode *ghost*.

Pandora. Dans la plateforme Pandora la synchronisation est conservative [1] et repose sur une grille 2D. Les données et les agents situés dans les zones de recouvrement sont copiés et envoyés aux cellules voisines à chaque pas de temps. Pour résoudre le problème de la dépendance de données, la simulation est découpée en parties, numérotées de 0 à 3. Au cours d'un pas de temps tous les processus exécutent séquentiellement chacune des parties, dans le même ordre : la partie 0, puis 1, 2 et 3. Une fois l'exécution d'une

partie terminée, les zones de recouvrement sont envoyées aux cellules voisines. De cette façon, il n'y a pas de conflits de cohérence car les parties exécutées en parallèle ne sont pas adjacentes. Cette approche originale réduit les coûts de synchronisation avec un modèle en partie *ghost* et en partie *non ghost*, suivant l'emplacement des données. Cela limite tout de même l'utilisation de Pandora à des modèles à deux dimensions.

Synthèse. Les PDMAS existants utilisent donc principalement une approche conservative quant à la causalité, certains laissant cette gestion au développeur. Ceci paraît justifié car l'approche optimiste convient peu aux systèmes multi-agents qui animent les agents par pas de temps, donc de manière uniforme au sein de processus parallèles à l'inverse des systèmes à événements discrets où les processus ont une répartition temporelle des événements différente entre eux. Il est donc moins intéressant de faire avancer plus vite certains processus et une politique de répartition de la charge peut s'avérer plus efficace.

Les plateformes proposent principalement un mode *ghost*, moins lourd qu'une mise à jour systématique, qui recopie les données vers les autres processus au changement de pas de temps, lorsque tous les agents sont dans un état fixe.

Une analyse de plusieurs types de modèles permet cependant de montrer que tous n'ont pas les mêmes besoins de synchronisation. Nous proposons donc, dans la section suivante, une étude des besoins de synchronisation de modèles multi-agents et différentes politiques pouvant répondre à ces besoins.

3 Impact des synchronisations

La synchronisation est un point clé pour une exécution efficace d'une simulation multi-agents parallèle, du fait des nombreux échanges et dépendances temporelles qu'elle induit. Son impact sur la parallélisation dépend néanmoins du modèle lui-même : dans un modèle où les agents n'interagissent pas, aucune synchronisation n'est nécessaire. L'intérêt du modèle agent repose cependant justement sur la capacité des agents à interagir [6]. Notre objectif est donc d'étudier l'impact de la synchronisation sur les temps d'exécution et l'impact de politiques de synchronisation plus relâchées. En effet, dans les systèmes multi-agents, l'observation d'un phénomène s'effectue en général au niveau macroscopique et non microscopique. De ce point de vue, il est possible que, dans les simulations composées d'un grand nombre d'agents, des synchronisations erronées ou fausses se compensent et limitent ainsi l'impact d'une synchronisation relâchée. Il est donc intéressant de mettre en relation l'erreur possible avec le surcoût dû à la synchronisation. Nous proposons dans la suite différentes politiques de synchronisation et étudions leur impact.

3.1 Quand synchroniser ?

L'objectif est de garantir l'accès à des données à jour et la cohérence des actions, pour respecter les règles du modèle. Plusieurs synchronisations sont donc nécessaires, que la simulation soit en mode *ghost* ou *non ghost* : (i) à la fin

de chaque pas de temps pour permettre le passage au pas suivant et garantir que tous les processus exécutent le même pas de temps, (ii) à la migration d'un agent d'un processus à un autre pour continuer à exécuter ses comportements il est nécessaire que les processus concernés soient dans le même état, (iii) à la mise à jour des zones de recouvrement pour garder la continuité des champs de perception des agents lors de la distribution de l'environnement sur plusieurs processus.

Une exécution en mode *ghost* ne nécessite pas de gérer ces problèmes d'écriture mais elle ne peut conserver que la cohérence des modèles qui ne nécessitent pas d'écriture concurrente, c'est à dire où deux entités ne modifient pas la même donnée au cours d'un pas de temps. En effet, utiliser le mode *ghost* avec des écritures concurrentes peut conduire à une violation des règles du modèle. Par exemple, dans un modèle proie-prédateur, plusieurs prédateurs pourraient manger une même proie au cours d'un pas de temps puisque, suite à une première attaque, la mort de la proie est enregistrée dans la copie de travail et non dans la copie *ghost* qui est utilisée pour connaître l'état du système. C'est seulement au changement de pas de temps que la mort de la proie sera reportée dans la copie *ghost*. À noter que ceci est vrai, même si la simulation n'est pas parallèle.

À l'opposé, dans le mode *non ghost*, les informations sont accessibles en lecture et en écriture. Ceci nécessite alors des mécanismes de synchronisation pour les données détenues par d'autres processus ou situées dans les zones de recouvrement. Dans ce cas, les points de synchronisation précédemment définis ne sont pas suffisants car ils ne permettent pas de gérer les écritures dans les zones de recouvrement. L'accès aux données doit donc être géré pour garantir qu'aucune incohérence (ou biais) n'est injectée dans la simulation et que les informations sont à jour.

Dans les modèles agents, il existe des modèles qui nécessitent que les données soient accessibles uniquement en lecture, d'autres en lecture et en écriture. La synchronisation est donc à considérer au cas par cas. Pour illustrer ceci, nous analysons différents types de modèles dans la suite.

3.2 Analyse de modèles

Pour évaluer l'impact de la synchronisation sur les résultats d'exécution des simulations multi-agents, nous utilisons trois modèles agents (Proie-prédateur, Virus et Flocking) qui nécessitent des niveaux de synchronisation différents pour s'exécuter de manière cohérente.

Il est important de noter que le système modélisé et son implémentation ont un impact important sur la synchronisation qui doit être mise en place pour garantir la qualité des résultats obtenus par la simulation parallèle. Ceci est, en particulier, vrai si le choix d'implémentation utilise un *ghost* ou non. Ainsi, parmi les modèles suivants, nous avons choisi l'approche que nous avons le plus souvent trouvée par rapport à ce choix d'implémentation. Changer ce choix modifierait les contraintes de synchronisation et conduirait à d'autres conclusions.

Le modèle Flocking [14]. Le modèle Flocking simule le vol d'une nuée d'oiseaux afin d'étudier le comportement

collectif. Le modèle est composé d'un seul type d'agent, les oiseaux qui sont localisés dans l'espace et ont une zone de perception réduite. Chaque agent oiseau a trois comportements : (i) la cohésion qui le pousse à se rapprocher des oiseaux proches, (ii) l'alignement qui le pousse à se déplacer dans la même direction que les oiseaux voisins et (iii) la séparation qui le pousse à tourner pour éviter un oiseau qui est trop près. Ces comportements déterminent la nouvelle position de l'oiseau en fonction de la position des oiseaux qui composent son voisinage. Le modèle possède plusieurs paramètres : la taille de l'environnement, la distance maximale qu'un oiseau peut parcourir par pas de temps, la durée d'un pas de temps, les taux de cohésion, d'alignement et de séparation.

Ce modèle fonctionne en mode *ghost*. Les agents calculent leur déplacement en fonction de la position des oiseaux voisins obtenue au pas de temps précédent. La mise à jour des données des zones de recouvrement à chaque pas de temps garantit que chaque agent oiseau dispose des informations correctes pour calculer son déplacement. Il n'y a donc pas de problème de concurrence d'accès sur ces données puisqu'elles sont accédées uniquement en lecture.

Le modèle Virus [15]. Le modèle Virus permet de simuler la transmission et la survie d'un virus dans une population [18]. Il est composé d'un seul type d'agent : les personnes qui sont localisées sur une grille en deux dimensions et n'ont qu'une connaissance partielle de l'environnement dans lequel ils évoluent. Les agents ont cinq comportements : (i) le vieillissement, jusqu'à ce qu'ils meurent, (ii) le déplacement de manière aléatoire sur l'environnement, (iii) l'infection des personnes de leur voisinage, (iv) la récupération qui permet à un agent infecté d'avoir une probabilité de devenir immunisé et (v) la reproduction, pour les personnes non contaminées, qui renouvelle la population. Le modèle possède plusieurs paramètres : la capacité de transport du virus, l'âge maximum, le taux de natalité, le taux de reproduction et le nombre de personnes porteuses du virus à l'initialisation du modèle.

Le modèle Virus fonctionne en mode *non ghost*. Il est donc possible que, dans un même pas de temps, un agent *A*, qui a été infecté, infecte à son tour un agent *B*. Si les deux agents ne sont pas exécutés sur le même processus, il est nécessaire de mettre à jour les données distantes. Pour finir, nous pouvons remarquer qu'un agent ne change pas son état même s'il est infecté plusieurs fois. Cette propriété, que nous appelons écriture idempotente, fait que nous n'avons pas à gérer de concurrence en écriture sur le changement de l'état de l'agent, puisque même si deux agents infectent un même agent l'ordre des deux exécutions n'a pas d'incidence sur le résultat final.

Le modèle proie-prédateur [16]. Le modèle proie-prédateur explore la stabilité des écosystèmes. Le modèle étudié possède trois types d'agents : les loups (prédateurs), les moutons (proies/prédateurs) et l'herbe (proie). Les loups et les moutons se déplacent au hasard dans l'environnement. L'herbe disparaît lorsqu'elle est mangée et repousse après un temps fixé. Chaque étape coûte de l'énergie aux loups et aux

moutons qui, lorsqu'ils n'en ont plus, meurent. L'énergie peut-être reconstituée pour un loup en mangeant un mouton et pour un mouton en mangeant de l'herbe. Pour permettre à la population de perdurer, les loups et les moutons ont une probabilité de se reproduire à chaque pas de temps. Tous les agents sont localisés sur une grille à deux dimensions et ne connaissent que leur zone de perception. A chaque pas de temps, les agents loups et moutons exécutent les quatre comportements suivants dans l'ordre donné : (i) se déplacer aléatoirement sur l'environnement, (ii) se nourrir de proies si elles se situent dans leur champ de perception, (iii) mourir s'ils n'ont plus d'énergie ou ont atteint leur durée de vie, et (iv) se reproduire. Un agent herbe apparaît aléatoirement tous les n pas de temps. Le modèle possède plusieurs paramètres : la taille de l'environnement, le nombre de loups, le nombre de moutons, le taux de natalité, le taux de reproduction, le gain de vie lorsqu'un prédateur mange une proie et le temps de croissance maximal des agents herbe.

Le modèle proie-prédateur fonctionne en mode *non ghost*. Dans la mesure où les prédateurs mangent les proies, ils en changent l'état, ce qui engendre une écriture dans les données de l'agent. Comme pour le modèle virus cela oblige à une synchronisation pendant le pas de temps pour prendre en compte la modification mais il est, en plus, nécessaire de gérer les écritures concurrentes. A cause du parallélisme, plusieurs prédateurs situés dans des processus différents peuvent être tentés de manger une même proie dans la zone de recouvrement. Si tous les prédateurs mangent cette proie (celle qui est dans la zone de recouvrement) alors chacun bénéficiera d'un apport en énergie et donc d'une augmentation de sa durée de vie, ce qui constitue une erreur par rapport au modèle séquentiel. Il est donc nécessaire de synchroniser tous les agents qui souhaiteraient manger une proie pour garantir qu'un seul prédateur la mangera.

Synthèse. L'analyse de ces modèles met en évidence différents besoins en termes de synchronisation. Ces besoins dépendent des caractéristiques du modèle et nous permettent de définir trois types en fonction des interactions entre les agents : (i) les modèles en lecture (L), comme flocking, (ii) les modèles en écriture idempotente (EI), comme le modèle virus, où l'état de l'agent ne change plus après une écriture et (iii) les modèles en écriture concurrente (EC). Il est alors possible d'assister le modélisateur en lui fournissant le choix de la politique de synchronisation qui permet l'implémentation correcte de son modèle tout en limitant l'impact sur les performances. Ainsi nous définissons dans la section suivante plusieurs politiques de synchronisation.

3.3 Politiques de synchronisation

Nous proposons ici plusieurs politiques de synchronisation, issues de l'analyse des modèles précédents, dont nous souhaitons évaluer l'impact sur les résultats et sur le temps d'exécution des simulations, sachant que d'autres politiques pourraient présenter un intérêt pour d'autres modèles. A ces propositions nous ajoutons le cas "sans synchronisation" qui sert de référence.

La politique **aucune synchronisation (NS)** distribue la simulation en n portions sans zone de recouvrement ni écritures

distantes. Les agents peuvent se déplacer d'un processus à un autre en ayant un champ de perception tronqué lorsqu'ils sont proches des limites des processus.

La politique **overlapping zones (OLZ)** ne gère que des zones de recouvrement, qu'elle copie à chaque pas de temps. Les écritures n'y sont pas reportées sur les originaux et sont écrasées au pas de temps suivant par la mise à jours.

La politique **écritures asynchrones (EA)** fait des écritures à distance sans attendre une confirmation ou une valeur de retour. Elle est utilisée lorsqu'un agent modifie une donnée de la zone de recouvrement et que cette écriture doit être prise en compte dans le pas de temps courant mais que l'agent n'attend pas de donnée en retour.

La politique **synchronisation stricte (SS)** gère les zones de recouvrement et les écritures concurrentes pour garantir au maximum la reproduction du cas séquentiel. Chaque demande en écriture est bloquante jusqu'à l'acquittement de sa prise en compte, ainsi la cohérence des données est garantie. Elle est ce qu'il y a de plus strict en termes de synchronisation sans revenir à une exécution séquentielle.

Contrairement à la synchronisation stricte, la **synchronisation stricte décalée (SSD)** s'effectue de manière non-bloquante. Ainsi, lorsqu'un agent effectue une demande de synchronisation, il est mis en attente de réponse jusqu'à la fin du pas de temps afin que l'exécution des autres agents se poursuive, assurant ainsi un meilleur recouvrement calcul-communication.

Pour diriger le choix d'une politique de synchronisation pour un modèle, il est nécessaire de connaître l'incidence qu'aura le choix de cette politique sur le modèle parallélisé. Les spécifications données précédemment permettent de discerner l'impact sémantique et les contraintes garanties. Pour connaître l'incidence de chacune des politiques sur les performances de la simulation, nous présentons dans la suite les mesures de performance réalisées en implémentant ces politiques sur les modèles étudiés. Les modèles choisis sont spatialisés mais ces problématiques s'appliquent de la même manière sur des modèles non-spatialisés puisqu'elles sont liées aux échanges de données entre agents plutôt qu'à la position de ceux-ci.

4 Expérimentations

L'objectif de cette section est de mettre en évidence le lien entre les performances d'exécution d'un modèle et la politique de synchronisation choisie en fonction de la variation d'extensibilité ou de montée en charge. Comme aucune des plateformes vues précédemment n'offre ces politiques de synchronisation, nous les avons implémentées dans des modèles pour les tester.

4.1 Modèles et politiques de synchronisation

Nous utilisons les trois modèles vus précédemment, choisis pour leurs différents besoins de synchronisation. La table 1 donne les politiques de synchronisation utilisées avec chacun des modèles.

Le modèle Flocking est un modèle L. Deux politiques de synchronisation sont donc testées : aucune synchronisation (NS) et utilisation de zones de recouvrement (OLZ). Le

Modèle	Ghost	Type	Politiques
Flocking	Oui	L	NS - OLZ
Virus	Non	EI	EA - OLZ - SSD
PP	Non	EC	OLZ - SSD - SS

TABLE 1 – Politiques de synchronisation utilisés

modèle Virus est un modèle EI. Les politiques de synchronisation utilisées sont donc l'écriture asynchrone (EA), les zones de recouvrement (OLZ) et la synchronisation stricte décalée (SSD). La politique de synchronisation stricte (SS) n'est pas utilisée puisque l'écriture idempotente ne change pas l'état d'un agent contaminé. Il n'est donc pas nécessaire de gérer l'écriture concurrente sur son changement d'état puisque l'ordre d'exécutions de deux écritures n'a pas d'incidence sur le résultat final. Le modèle proie-prédateur est un modèle EC puisqu'une proie ne peut être mangée qu'une seule fois. Les politiques de synchronisation utilisées sont les zones de recouvrement (OLZ), la synchronisation stricte décalée (SSD) et la synchronisation stricte (SS).

4.2 La plateforme de test

L'implémentation des modèles a été effectuée à partir de notre plateforme FPMAS[10], une plateforme multi-agents parallèle qui repose sur la bibliothèque Zoltan pour gérer la distribution de la simulation, le modèle agent étant représenté par un graphe afin de tirer parti des algorithmes de partitionnement parallèles. FPMAS propose également des mécanismes de synchronisation, dont la synchronisation stricte, contrairement aux plateformes précédemment citées et qui n'offrent même pas la possibilité de l'implémenter. Une fois le modèle et ses politiques de synchronisation implémentés, FPMAS peut l'exécuter dans un environnement parallèle adapté au calcul haute performance. La simulation est divisée en N portions et chacune des portions est distribuée sur P processeurs. Les performances dépendent alors du modèle exécuté. A noter que les politiques de synchronisation sont implémentées directement dans les modèles, ce qui peut engendrer quelques différences de performances d'un modèle à l'autre. Pour cette raison nous ne faisons pas la suite que des comparaisons entre politiques par rapport à un modèle donné.

Pour exécuter les simulations, nous avons utilisé le Méso-centre de calcul de Franche-Comté. Le cluster est constitué de nœuds bi-processeurs, avec des processeurs Xeon E5 (8*2 cœurs) cadencés à 2.6 Ghz et 32 Go de mémoire vive. Le cluster possède un total de 1280 cœurs gérés par le système de batch SGE¹. Les nœuds sont interconnectés par un réseau non bloquant QDR InfiniBand² organisé en fat tree. Chaque point des courbes représente une moyenne de 10 exécutions avec 10 graines différentes.

4.3 Impact de l'extensibilité

L'extensibilité des modèles est étudiée en fixant le nombre d'agents de la simulation et en faisant varier le nombre

1. https://en.wikipedia.org/wiki/Oracle_Grid_Engine

2. https://fr.wikipedia.org/wiki/Bus_InfiniBand

de processus sur lesquels elle s'exécute. Nous mesurons l'impact de l'extensibilité à l'aide de deux métriques : le temps d'exécution et le speed-up.

Pour chacun des modèles nous avons calculé un speed-up avec comme référence le temps d'une exécution parallèle sur 16 cœurs car la taille des données des modèles est trop grande pour un seul processus. Le speed-up sur p processeurs est donné par $T(p_{ref})/T(p)$ où $T(p_{ref})$ est le temps d'exécution parallèle sur le nombre de processeurs de référence et $T(p)$ est le temps d'exécution sur p processeurs. Pour les trois modèles l'extensibilité est bonne avec un speed-up de plus ou moins 20 suivant les modèles, alors que le speed-up idéal est de 32. Les résultats obtenus par les politiques sans synchronisation, ou avec moins de synchronisation, sont meilleurs que ceux ayant des synchronisations plus strictes. Pour des raisons de reproductibilité, la même configuration initiale est utilisée pour toutes les exécutions, seule la graine varie d'une exécution à une autre.

Modèle Flocking. L'environnement du modèle Flocking est basé sur un cube 1000^3 . A l'initialisation, 100000 oiseaux sont répartis de manière aléatoire dans l'espace. Les taux de cohésion, de séparation et d'alignement sont fixés à 1 tandis que le taux d'aléa est lui fixé à 1.5. Ces paramètres ont été choisis, afin de ne pas favoriser l'un des trois critères composant les comportements des oiseaux. Seul le taux d'aléa est supérieur aux autres taux dans le but de générer des déplacements plus chaotiques et donc de tester davantage la synchronisation. Chaque simulation est exécutée durant 2000 pas de temps.

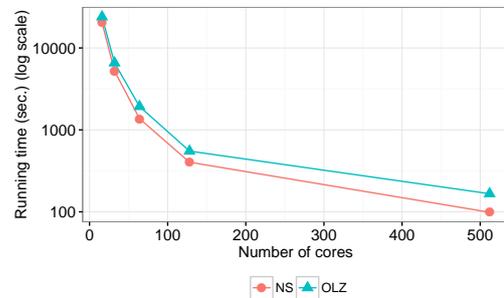


FIGURE 1 – Temps d'exécution du modèle Flocking

La figure 1 donne les temps d'exécution du modèle Flocking pour les politiques de synchronisation NS et OLZ. La différence des temps d'exécution entre les deux politiques de synchronisation est d'environ 15% pour 16 cœurs, 26% pour 128 cœurs, et 65% pour 512 cœurs. Lorsque le nombre de cœurs augmente, plus de messages sont nécessaires pour mettre à jour les zones de recouvrement, ce qui explique cette différence croissante.

Modèle Virus. Le modèle Virus a été exécuté sur une grille de 1000^2 qui représente l'environnement avec une capacité maximale de 500000 personnes. A l'initialisation 9600 personnes sont saines et 640 sont infectées par le virus. Tous les agents sont positionnés de manière aléatoire sur l'environnement. Le taux d'infection est fixé à 0.65 et le

taux de reproduction est fixé à 0.2. Le taux de récupération, c'est à dire le fait qu'une personne infectée devienne immunisée, est fixé à 0.5. Ces valeurs sont issues du modèle Virus de NetLogo. Seule la taille de l'environnement et la capacité maximale ont été adaptées pour obtenir un modèle de grande taille. Pour finir, chaque simulation est exécutée durant 800 pas de temps.

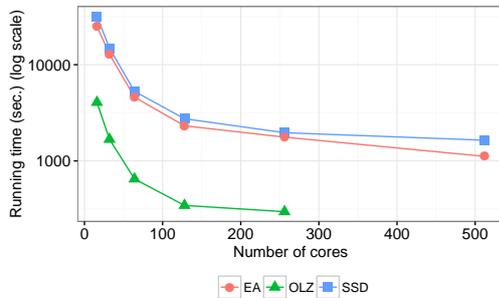


FIGURE 2 – Temps d'exécution du modèle Virus

La figure 2 donne les temps d'exécution du modèle virus pour les synchronisations *OLZ*, *EA* et *SSD*. Pour les courbes *OLZ* il n'y a pas de point pour 512 cœurs car nous n'avons pas eu assez de temps sur le calculateur pour réaliser l'expérimentation. Les résultats obtenus avec le modèle flocking se confirment ici avec une différence de 27% pour 16 cœurs, seulement 15% pour 128 et 47% pour 512 cœurs. Deux raisons expliquent ce surcoût : le traitement additionnel en fin de pas de temps de la synchronisations *SSD* et l'augmentation du nombre de messages nécessaires à la synchronisation due au plus grand nombre de cœurs. Cette figure nous montre également le coût induit par la synchronisation des agents. Pour 16 cœurs, les temps d'exécution de la courbe *OLZ* sont environ 8 fois meilleurs que la courbe *SSD*. Cette différence tend à décroître avec l'augmentation du nombre de cœurs, par exemple, pour 256 cœurs cette différence n'est plus que de 6.6. Pour les courbes *EA* et *SSD* le ratio n'est que de 1.27 pour 16 cœurs, et de 1.4 pour 512 cœurs. Il croît donc avec le nombre de cœurs du fait du nombre plus important de messages.

Modèle proie-prédateur. Le modèle proie-prédateur (PP) utilisé pour les expérimentations est basé sur un environnement grille de 400^2 où 25000 moutons et 17000 loups sont initialement positionnés de manière aléatoire. Le modèle des comportements et l'initialisation de l'énergie des agents loups et moutons est issue du modèle NetLogo. L'énergie gagnée par un mouton lorsqu'il mange de l'herbe est fixée à 5 et à 20 lorsqu'un loup mange un mouton. En ce qui concerne les taux de reproduction, ils sont fixés à 0.5 pour les moutons et à 0.4 pour les loups. Les durées de vie sont respectivement de 4 et 20 pour les moutons et les loups. La croissance de l'herbe est de 8. Pour finir les simulations sont exécutées durant 2000 pas de temps.

La figure 3 présente les temps d'exécution du modèle PP pour les politiques de synchronisation *OLZ*, *SSD* et *SS*. Pour la même raison que précédemment, aucun calcul n'a été

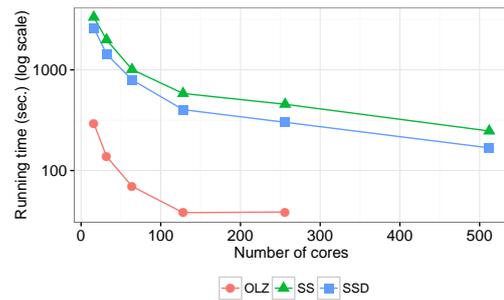


FIGURE 3 – Temps d'exécution du modèle PP

effectué sur 512 cœurs avec *OLZ*. Les points de 16 à 64 cœurs permettent cependant de mettre en évidence, comme pour le modèle virus, le coût important à payer pour avoir des politiques de synchronisations strictes. La courbe *SSD* montre que la relaxation de la synchronisation à la fin du pas de temps de cette politique permet un gain de temps d'exécution par rapport à la synchronisation stricte. Ce gain est dû au fait que le traitement des agents n'est plus bloqué en attente de la réponse à une écriture concurrente : on a donc un meilleur recouvrement calcul-communication. Sur les modèles étudiés, le calcul des speed-up montre une bonne extensibilité et le niveau de synchronisation choisi pour implémenter le modèle ne semble pas induire d'impact. Les simulations avec un grand nombre de cœurs profitent donc bien du parallélisme, ce qui confirme que les systèmes multi-agents peuvent bénéficier d'une parallélisation. Néanmoins, les politiques de synchronisation ont un coût très important (jusqu'à un facteur 8), qui est dû aux communications engendrées. En effet, dans les modèles étudiés, les agents ont un comportement relativement simple qui s'exécute beaucoup plus rapidement qu'une communication.

4.4 Impact de la montée en charge

La montée en charge est réalisée en fixant le nombre de processus et en faisant varier le nombre d'agents. Le modèle proie-prédateur n'est pas présenté dans cette section car il est très difficile d'y faire varier le nombre d'agents puisque la population s'auto-équilibre.

Modèle Flocking. Le jeu de valeurs utilisé pour évaluer la montée en charge du modèle Flocking est le même que précédemment, à la différence que le nombre d'agents oiseaux qui composent la simulation varie de 100000 à 1000000.

La figure 4 présente l'impact de la montée en charge sur 512 cœurs. Les politiques *NS* et *OLZ* supportent bien la charge jusqu'à 500000 agents. Au delà de 500000 agents, les courbes croissent plus rapidement. Sans surprise, la version sans synchronisation supporte mieux la charge que la version avec zone de recouvrement, qui consomme environ un tiers de performance en plus. L'accroissement de la charge à partir de 500000 agents s'explique par un manque d'optimisation lors de la recherche du voisinage des agents. Au lieu de mettre à jour uniquement les agents qui arrivent ou partent du voisinage, un parcours de l'environnement est effectué à chaque pas de temps pour établir le champ de

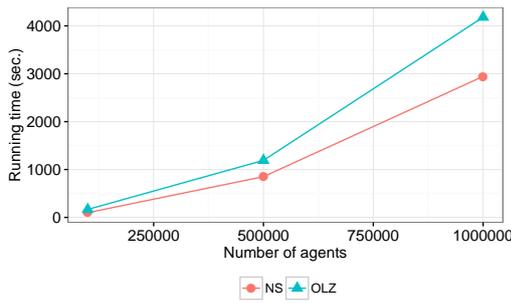


FIGURE 4 – Montée en charge du modèle Flocking

perception de chaque agent.

Modèle Virus. Les paramètres utilisés pour la montée en charge sont les mêmes que précédemment.

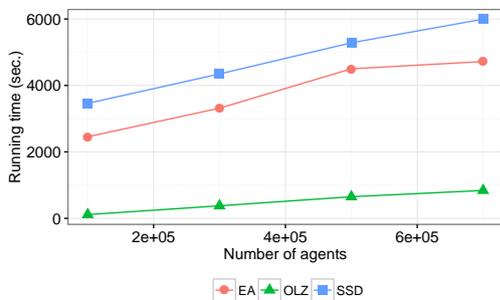


FIGURE 5 – Montée en charge du modèle Virus

La figure 5 montre l'impact de la montée en charge du modèle Virus de 100000 à 700000 agents et 800 pas de temps. Seulement 64 cœurs ont été utilisés pour cette courbe car il n'a pas été possible d'obtenir à nouveau autant de ressources de calcul (512 cœurs) sur le cluster partagé.

La courbe *OLZ* supporte évidemment mieux puisqu'elle ne gère pas les écritures. Pour les autres, on constate que *EA* supporte mieux la charge que *SSD*. La courbe *SSD*, reste linéaire, alors que la courbe *EA* croît très peu de 500000 à 700000 agents. Ceci est dû au surcoût lié à l'accumulation des synchronisations en fin de pas de temps. Le ratio de performance obtenu entre 100k et 700k pour la courbe *EA* est de 1.9, 1.73 pour *SSD* et 7.5 pour *OLZ*.

Les résultats sur la montée en charge confirment que, quelle que soit la politique de synchronisation utilisée, les modèles multi-agents tirent bénéfice d'une parallélisation. Les résultats obtenus avec la politique de synchronisation *OLZ* montrent que cette politique est plus efficace que les politiques plus contraintes. Nous avons donc étudié l'impact des politiques de synchronisation sur les résultats des exécutions, ce que nous présentons dans la suite.

4.5 Impact sur les résultats

Pour étudier l'impact des différentes politiques, nous observons deux résultats : le résultat de la simulation lui-même et le nombre d'interactions incohérentes. Le résultat d'une

simulation est ce qui est attendu par le modélisateur. Nous étudions l'impact d'une exécution plus ou moins synchronisée sur les résultats de l'exécution. Par ailleurs, dans les cas où les écritures dans les zones de recouvrement sont gérées, c'est à dire avec les politiques *EA*, *SS* et *SSD*, des interactions peuvent mettre en évidence une incohérence entre l'information locale et l'information distante. Par exemple pour le modèle proie prédateur, un loup essaie de manger un mouton dans la zone de recouvrement seulement si il est vivant. Par conséquent, un retour négatif pour cette action montre que le mouton était en fait déjà mort dans les données distantes. En conséquence, prendre seulement en compte la zone de recouvrement aurait mené à une action incohérente où au moins deux loups auraient mangé le même mouton. De même, pour le modèle virus, un agent essaie d'en infecter un autre seulement si ce dernier est sain. Ainsi, pour analyser l'impact des interactions sur les résultats des simulations nous comptabilisons le nombre total de demandes de synchronisation, les synchronisations pour lesquelles l'information était cohérente (notées *CO*) et celles pour lesquelles l'information était incohérente (notées *NCO*). Cette analyse n'inclut pas le modèle Flocking car elle ne s'applique pas à un modèle en lecture.

Modèle Virus. La figure 6 présente le détail des demandes de synchronisation pour les synchronisations *EA* et *SSD*.

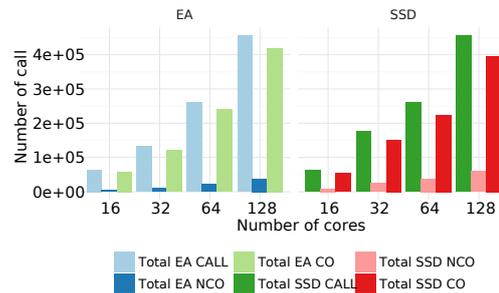
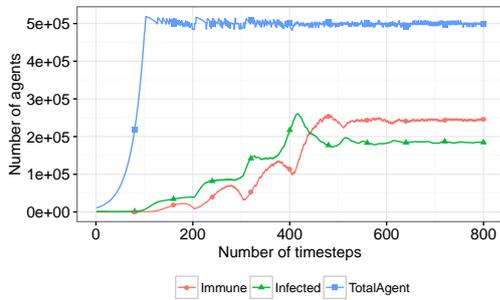


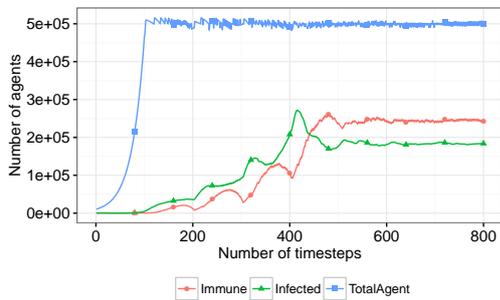
FIGURE 6 – Détail des synchronisations du modèle Virus

Pour les deux politiques de synchronisation le nombre total de demandes de synchronisation croît de manière linéaire avec le nombre de cœurs car, lorsque le nombre de cœurs augmente la simulation est divisée en plus de processus. Il y a donc plus de zones de recouvrement et plus d'interactions potentielles. Le nombre d'interactions *NCO* est très faible pour *EA*. Il croît d'environ 10% pour *SSD* à cause des synchronisations gérées en fin de pas de temps. Les agents qui effectuent une demande de synchronisation sont suspendus et exécutés à la réception de la réponse, en fin de pas de temps. De ce fait, certaines interactions qui étaient potentiellement *CO* au moment de la demande de synchronisation peuvent devenir *NCO* si les agents concernés n'avaient pas encore été exécutés. Il faut donc poser la question de savoir quel est l'impact de ces interactions *NCO* sur les résultats. La figure 7 présente les résultats de l'exécution sur 128 cœurs. Les résultats pour *SSD* (figure 7(a)) et *EA* (7(b)) sont quasi identiques. Les quelques variations s'expliquent par

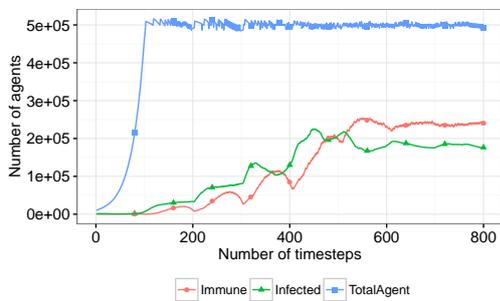
les interactions *NCO* traitées en fin de pas de temps par la politique *SSD*. En revanche, les résultats de *OLZ* (figure 7(c)) présentent de nombreuses différences, malgré une tendance identique.



(a) SSD



(b) EA



(c) OLZ

FIGURE 7 – Résultats d’une exécution du modèle Virus

Modèle proie-prédateur. La figure 8 présente le détail des synchronisations lors de l’exécution du modèle PP. Comme pour le modèle Virus, le nombre de demandes de synchronisations croît de manière linéaire avec le nombre de cœurs. Le nombre d’appels *NCO* reste plus faible pour la politique *SS* que pour la politique *SSD*.

La figure 9 présente les résultats sur 128 cœurs des politiques *SSD*, *SS* et *OLZ*. Le modèle PP est un modèle très sensible : la modification d’un paramètre peut conduire à une instabilité qui se traduit par la mort d’une des espèces. Le changement d’une politique induit ainsi des différences de résultats entre les courbes *SSD* (figure 9(a)) et les courbes *SS* (figure 9(c)). La figure 9(b) qui représente l’exécution sans synchronisation montre des résultats proches de la po-

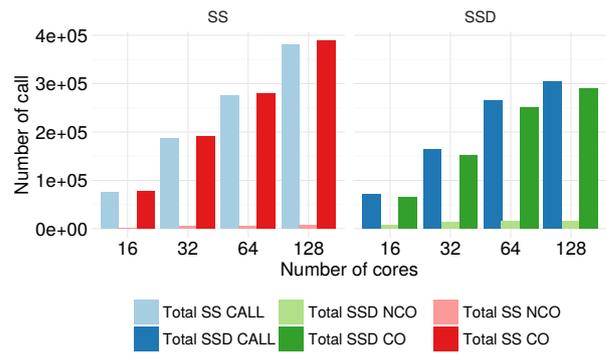
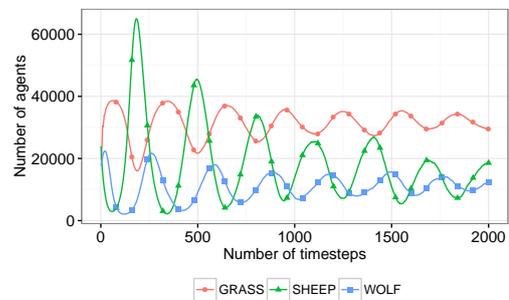
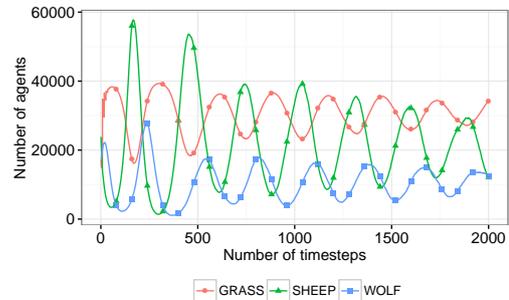


FIGURE 8 – Détail des synchronisations du modèle PP

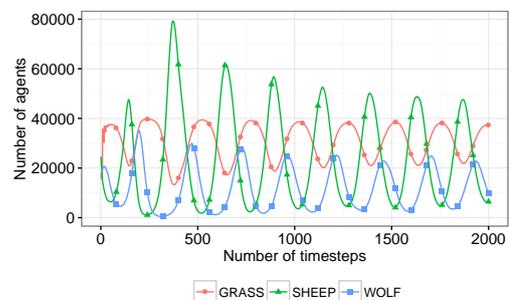
litique *SSD* (figure 9(a)). Nous retrouvons tout de même des courbes cycliques qui tendent à s’équilibrer.



(a) SSD



(b) OLZ



(c) SS

FIGURE 9 – Résultats d’une exécution du modèle PP

Au vu des courbes, on peut s'interroger sur ce qu'il est important d'observer dans les résultats du modèle. Si nous regardons plus en détails ces figures, nous remarquons que le nombre maximal de moutons diffère, environ 60000 pour *OLZ* et *SSD* et environ 80000 pour *SS*. On remarque également que sur la figure 9(c) (synchronisation *SS*) la phase est régulière avec un décalage de l'ordre de $\pi/2$. En revanche, sur les figures 9(b) et 9(a), un décalage de phase s'effectue au cours du temps. Les courbes *OLZ* et *SSD* sont semblables et donnent des résultats erronés.

5 Conclusion

Notre étude sur l'impact des politiques de synchronisation vise à sensibiliser la communauté agent, et plus particulièrement les modélisateurs, aux problèmes liés à la parallélisation d'un modèle. Nous évaluons pour cela différentes politiques de synchronisation en lien avec différents modèles et nous mettons en évidence que, suivant le modèle à implémenter, le niveau de synchronisation a un impact sur performances d'exécution et les résultats. A travers nos expérimentations, nous quantifions le coût de la synchronisation sur les performances de la simulation suivant les modèles et nous montrons son impact sur les résultats. La synchronisation est de fait dépendante des comportements des agents : c'est au modélisateur de prévoir l'exécution du modèle et donc d'adapter la modélisation pour prendre en compte le coût et l'impact de la synchronisation lors de la conception d'un modèle parallèle.

Pour la suite nos travaux s'orientent vers une analyse de nombreux modèles en vue d'identifier les problèmes génériques de synchronisation et de proposer des politiques efficaces, qui pourraient être implémentées dans une plateforme pour simplifier le travail du modélisateur.

Remerciements

Les calculs présentés dans cet article ont été effectués sur le calculateur du Mésocentre de calcul de Franche-Comté

Références

- [1] E. S. Angelotti, E. E. Scalabrin, and B. C. Ávila. Pandora : a multi-agent system using paraconsistent logic. In *Computational Intelligence and Multimedia Applications, ICCIMA*, pages 352–356. IEEE, 2001.
- [2] K. Mani Chandy and Jayadev Misra. Distributed simulation : A case study in design and verification of distributed programs. *IEEE Transactions on software engineering*, (5) :440–452, 1979.
- [3] AL Chin, AD Worth, AC Greenough, S Coakley, M Holcombe, and M Kiran. Flame : An approach to the parallelisation of agent-based applications. *Work*, 501 :63259, 2012.
- [4] Nicholson Collier and Michael North. Parallel agent-based simulation with repast for high performance computing. *SIMULATION*, Nov 2012.
- [5] G. Cordasco, R. Chiara, A. Mancuso, D. Mazzeo, V. Scarano, and C. Spagnuolo. A Framework for Distributing Agent-Based Simulations. In *Euro-Par 2011 : Parallel Processing Workshops*, volume 7155 of *LNCS*, pages 460–470, 2011.
- [6] Jacques Ferber and Jean-François Perrot. *Les systèmes multi-agents : vers une intelligence collective*. Inter-Editions Paris, 1995.
- [7] Olivier Gutknecht and Jacques Ferber. Madkit : A generic multi-agent platform. In *4th intl Conf. on Autonomous agents*, pages 78–79. ACM, 2000.
- [8] David R Jefferson. Virtual time. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 7(3) :404–425, 1985.
- [9] A. Rousset, B. Herrmann, C. Lang, and L. Philippe. A survey on parallel and distributed multi-agent systems for high performance computing simulations. *Computer Science Review*, 22 :27 – 46, nov 2016.
- [10] A. Rousset, B. Herrmann, C. Lang, L. Philippe, and H. Bride. Nested graphs : A model to efficiently distribute multi-agent systems on hpc clusters. *Concurrency and Computation : Practice and Experience*, 30(7) :e4407, 2018.
- [11] B. Samadi. Distributed simulation, algorithms and performance analysis. *PhD Thesis, Computer Science Department, Univ. of California, Los Angeles*, 1985.
- [12] P. Taillandier, DA. Vo, E. Amouroux, and A. Drogoul. GAMA : A Simulation Platform That Integrates Geographical Information Data, Agent-Based Modeling and Multi-scale Control. In *Principles and Practice of Multi-Agent Systems*, volume 7057, pages 242–258. Springer, 2012.
- [13] S. Tisue and U. Wilensky. Netlogo : Design and implementation of a multi-agent modeling environment. In *Proceedings of Agent*, volume 2004, pages 7–9, 2004.
- [14] U. Wilensky. Netlogo flocking model. *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL*, 1998.
- [15] U. Wilensky. Netlogo virus model. *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL*, 1998.
- [16] U. Wilensky. Netlogo wolf sheep predation (docked) model. *Center for connected learning and computer-based modeling, Northwestern Univ., Evanston, IL*, 2005.
- [17] Yadong Xu, Wentong Cai, Heiko Aydt, Michael Lees, and Daniel Zehe. Relaxing synchronization in parallel agent-based road traffic simulation. *ACM Trans. Model. Comput. Simul.*, 27(2) :14 :1–14 :24, May 2017.
- [18] J. A. Yorke, N. Nathanson, G. Pianigiani, and J. Martin. Seasonality and the requirements for perpetuation and eradication of viruses in populations. *American Journal of Epidemiology*, 109(2) :103–123, 1979.

Interactions dans les Communautés Mixtes Humains-Agents : État de l'Art

J. Saunier, A. Pauchet

Normandie Université, INSA Rouen Normandie, LITIS

{prenom.nom}@insa-rouen.fr

Résumé

L'explosion du nombre d'objets connectés, de l'informatique ubiquitaire et des interfaces humain-machine intuitives et naturelles permet une multiplication des systèmes cyber-physiques. Si les interactions avec un seul utilisateur disposent d'une littérature abondante dans le domaine des agents virtuels incarnés, la mise en place de systèmes coopératifs intégrant plusieurs utilisateurs et plusieurs agents virtuels reste difficile. Les problématiques classiques, telles que la gestion des tours de paroles par exemple, doivent être adaptées au contexte multipartie. De nouvelles problématiques, triviales en contexte dyadique comme la détection du destinataire d'un énoncé, émergent également. Dans cet article, nous discutons les défis actuels posés par les communautés mixtes agents-humains tant au niveau des agents que du système multi-agent dans son ensemble.

Mots-clés

Systèmes multi-agents, communautés mixtes, agents virtuels, robots, compagnons artificiels.

Abstract

The growth in the number of connected objects, of ubiquitous computing and of intuitive and natural human-machine interfaces is enabling a growing number of cyber-physical systems. While interactions with a single user have an abundant literature in the field of embodied virtual agents, the implementation of cooperative systems integrating several users and several virtual agents remains difficult. Classical issues, such as the management of turn-taking, for example, must be adapted to the multi-party context. New problems, trivial in a dyadic context such as the detection of the recipient of a statement, also emerge. In this article, we discuss the current challenges posed by mixed agent-human communities both at the level of agents and of the multi-agent system as a whole.

Keywords

Multi-agent systems, mixed communities, virtual agents, robots, artificial companions.

1 Introduction

L'explosion du nombre d'objets connectés et de l'informatique ubiquitaire permet une multiplication des systèmes cyber-physiques dans le contexte de l'Internet des Objets. Cette tendance s'accompagne de la mise à disposition d'interfaces humain-machine intuitives et naturelles dont l'objectif est de permettre aux utilisateurs d'interagir aisément. Nous pouvons par exemple citer Alexa, Siri ou Cortana, qui permettent une interaction en langage naturel [21].

De façon à faciliter l'acceptabilité de ces systèmes et l'engagement des utilisateurs, de nombreux travaux se sont intéressés aux compagnons artificiels, qui regroupent agents conversationnels animés et robots. Leurs principales applications se situent dans les domaines des environnements pour l'apprentissage humain [2, 46], dans lesquels les agents peuvent à la fois peupler l'environnement et servir de tuteurs à l'apprenant, les services à la personne [37] et la santé physique ou mentale [11], ou encore le tourisme [45] pour guider les visiteurs et interagir avec eux.

Si les interactions entre un utilisateur et un agent virtuel ou un robot disposent d'une littérature abondante, la mise en place pratique de systèmes coopératifs intégrant plusieurs utilisateurs et plusieurs compagnons artificiels reste difficile. Les problématiques classiques, telles que la gestion des tours de paroles par exemple, doivent être adaptées au contexte multipartie. De nouvelles problématiques, triviales en contexte dyadique comme la détection du destinataire d'un énoncé, émergent également.

Un point particulièrement saillant pour la mise en place de systèmes en interaction avec des humains est la gestion des émotions. L'avènement de l'informatique affective [39], qui s'intéresse aux systèmes prenant en compte les émotions des utilisateurs et pouvant en « jouer », n'est jamais directement traitée dans un contexte multipartie. Les mécanismes affectifs, comme par exemple la contagion émotionnelle ou l'empathie, sont alors beaucoup plus complexes à mettre en œuvre.

Dans cet article, nous discutons les défis actuels posés par les communautés mixtes agents-humains. Dans la deuxième section, nous recensons les exigences intrinsèques à la mise en place de communautés mixtes et les travaux existants, du point de vue de l'environnement partagé, des agents autonomes, et du système multi-agent dans

son ensemble, de façon à en extraire les critères nécessaires à la mise en place de ces communautés. Puis, nous identifions en section 3 les spécificités pour la communauté en terme de mécanismes multi-agents. En section 4, nous discutons l'acceptabilité de ces systèmes et soulevons quelques questions éthiques. Enfin, nous concluons et présentons des perspectives en section 5.

2 Communautés mixtes

Dans cette section, nous introduisons tout d'abord une définition et des exemples de communautés mixtes, puis nous précisons les conditions de déploiement de communautés mixtes au regard de l'état de l'art.

De façon à mettre en œuvre des communautés mixtes, nous pouvons catégoriser les besoins selon trois niveaux :

- (1) l'environnement partagé de réalité mixte, incluant des éléments réels et virtuels partagés et en interaction via d'éventuels dispositifs de réalité augmentée ou de réalité virtuelle,
- (2) les modèles d'agent, notamment dans leur capacité à prendre en compte les autres entités du système, et
- (3) les mécanismes collectifs.

2.1 Définitions et exemples

Un *système mixte* est un système dans lequel plusieurs humains et agents co-existent et interagissent, directement ou indirectement. Un exemple d'interaction indirecte est l'utilisation d'un système multi-agent pour adapter l'ambiance musicale d'un club en fonction de l'humeur des humains [48], tandis qu'un exemple d'interaction directe est d'interfacer via le langage naturel des agents logiciels aux humains [50].

Une *communauté mixte* est un système mixte dans lequel les agents disposent d'une incarnation. L'*incarnation* d'un agent correspond à une personnification permettant de représenter cet agent auprès des utilisateurs. Par exemple, un agent peut aussi bien être incarné par un personnage virtuel dans un environnement de réalité virtuelle ou projeté sur un écran, ou encore piloter le comportement d'un robot. Une communauté mixte peut ainsi mêler des agents virtuels, des robots, des humains, physiquement présents ou représentés par un avatar ou même par un robot de télé-présence.

De nombreuses configurations de communautés mixtes ont été étudiées dans la littérature. Dans [2], par exemple, l'humain est immergé dans un environnement virtuel, et interagit avec des agents conversationnels animés au sein de cet environnement. À l'inverse, dans [8] c'est l'humain qui se déplace au sein d'un environnement physique pour interagir avec différents agents robotiques et virtuels, situés à des endroits différents.

Les travaux impliquant des utilisateurs humains et des agents incarnés via la *réalité augmentée* sont encore rares dans la littérature. La réalité augmentée permet de superposer à la réalité des éléments numériques, par incrustation d'objets virtuels dans des séquences d'images ou via un dispositif semi transparent. Dans [19], par exemple, les auteurs montrent les éléments à mettre en place pour ce type

d'interactions, principalement liés aux caractéristiques intrinsèques de la réalité augmentée : calage de la scène et de ses éléments, adaptation de la colorimétrie, etc.

2.2 Besoins au niveau environnement

La notion d'environnement est fondamentalement polysémique puisqu'elle recouvre à la fois l'*environnement agent* [64], qui sert de support aux entités artificielles, l'*environnement de déploiement* matériel et logiciel, qui donne leur contexte d'exécution et les services utilisables [61], l'*environnement virtuel*, qui propose une représentation en 3 dimensions d'un monde pouvant être un calque du monde réel ou une alternative, et l'*environnement physique*.

Dans ce cadre, la problématique de l'environnement partagé émerge de l'interaction entre entités présentes dans des espaces différents [48] : la corporalité des agents conversationnels animés se situe par essence dans un environnement virtuel, tandis que les humains et robots se meuvent dans le monde physique. La restitution d'un agent virtuel dans l'environnement réel peut être menée via différents moyens, lesquels limitent leurs possibilités d'action et d'interaction. Par exemple, la restitution en réalité augmentée permet à un avatar d'être positionné dans tout l'espace, tandis que l'utilisation de projecteurs ou écrans restreint les opportunités de co-présence.

Du point de vue de l'architecture servant de support à la communauté mixte, la première exigence est donc la capacité d'intégrer, dans un même système, des entités dont l'incarnation appartient à la fois aux environnements virtuels et physiques. Cette exigence implique que les agents du SMA soient *situés* [1, 49], *i.e.* que l'environnement des agents a un impact facilitant [42, 63] ou contraignant [48] leurs capacités d'action et d'interaction.

Une seconde problématique découle de la nécessité d'introduire des primitives d'interaction entre humains et agents. Les interfaces naturelles, notamment dialogiques, ont montré qu'elles facilitent les interactions [56] au point de s'apparenter à celles que les humains entretiennent entre eux [26]. Cependant, ce type d'interactions implique également un ensemble de problèmes liés à la captation multimodale des utilisateurs humains : transcription automatique [68], détection des expressions faciales [28], *etc.*

De même, une problématique d'architecture de l'environnement multi-agent se pose pour que la communauté mixte puisse être ouverte, *i.e.* permette aux agents et humains d'entrer ou sortir du système, et qu'humains et agents puissent échanger dynamiquement leurs rôles au cours du temps. Cette dernière possibilité indique la nécessité de pouvoir interfacer de façon transparente humains et agents, mais également les différentes incarnations des agents au sein du système.

2.3 Besoins au niveau agent autonome

Sur le second niveau, la littérature montre que selon la théorie de l'esprit [7, 41], l'incarnation des agents induit chez les utilisateurs un ensemble d'attentes sur leurs comportements ainsi que sur leur fonctionnement interne [26].

Autrement dit, l'observation du comportement de l'agent suffit aux humains pour reconstruire un raisonnement, potentiellement erroné, sur ses motivations (voir *e.g.* une décision fondée sur la conservation de ressources interprétée comme une décision fondée sur l'émotion [6]). Par suite, la crédibilité de l'agent, définie de façon minimaliste comme issue d'un comportement cohérent dans le temps et l'espace, dépend de l'application d'un ensemble de règles connues et acceptées de l'utilisateur :

- la perception est locale (l'agent n'est pas omniscient),
- l'agent prend en compte de façon individualisée les autres éléments du système,
- son comportement peut être interprété comme relevant d'une rationalité humaine ¹.

De plus, un certain nombre d'éléments favorisent l'engagement des utilisateurs en interaction avec un agent : la personnalité, l'expression de sentiments, le langage corporel, l'empathie et la prise en compte des relations sociales [36, 55, 62]. Si ces facteurs sont importants dans un cadre dyadique, la présence de plusieurs agents incarnés renforce la nécessité d'individualiser leurs comportements, de façon à permettre de les différencier [16].

Il est également important de noter que cette prise en compte ne doit pas être réalisée uniquement vis-à-vis de l'utilisateur, mais également dans le cadre des relations entre agents, car celles-ci sont observables par l'utilisateur [59]. Ainsi, deux agents communiquant ensemble n'ont pas besoin d'utiliser la voie orale, mais un humain ne percevant pas la transmission d'information entre ceux-ci ne peut comprendre un éventuel changement d'état des interlocuteurs.

2.4 Besoins au niveau système multi-agent

Enfin, concernant le dernier niveau, la mise en œuvre de communautés mixtes implique de modéliser les phénomènes collectifs de façon *native*. En effet, les mécanismes multiparties tels que les interactions multimodales [5, 59, 69] ou la contagion émotionnelle [52] ne sont pas une simple agrégation de relations dyadiques.

Par exemple, d'un point de vue dialogique, les communications de groupe ont par essence des propriétés qui n'existent pas en situation dyadique, telles que la possibilité d'avoir des rôles différents d'émetteur ou de récepteur [23, 27], la gestion des tours de paroles ou encore les comportements de suivi de regard dans l'environnement partagé.

2.5 Synthèse

Les principaux éléments constitutifs des communautés mixtes sont résumés en table 1, selon les différents types d'espaces, d'entités et d'interactions.

Sur la base de ces éléments, nous pouvons en tirer les caractéristiques fondamentales des travaux existants :

¹. La notion de rationalité [31] ne peut être résumée à un raisonnement parfait, de par sa nature (limitée), et de par les multiples influences, parfois contradictoires, qui la composent.

Catégorie	Nom
Type d'espace	Physique Virtuel Mixte (augmenté)
Type d'entité	Humain Agent conversationnel Robot Objet connecté Objet virtuel / agent non incarné
Type d'interaction	Dialogique dyadique Dialogique multipartie Indirecte / stigmergique

TABLE 1 – Éléments d'une communauté mixte

- le système permet-il de gérer un seul utilisateur humain ou plusieurs, un ou plusieurs robots, un ou plusieurs agents incarnés ?
- le système prend-il en compte l'espace physique, virtuel, augmenté ?
- les interactions sont-elles dialogiques ?
- le système gère-t-il des mécanismes multiparties ?
- les comportements sont-ils individualisés ?

3 Mise en regard : les travaux issus des communautés agents autonomes et systèmes multi-agents

Les problématiques montrées précédemment relèvent des domaines des systèmes multi-agents, d'une part, et des compagnons artificiels d'autre part.

Dans notre vision, ces deux domaines possèdent déjà un important corpus de connaissances vis-à-vis des besoins exposés ci-dessus, qui nécessitent cependant d'être adaptées aux spécificités des communautés mixtes.

3.1 Architecture(s)

Nous avons vu dans la section précédente que deux problématiques architecturales de nature très différente se posaient : d'un côté, l'architecture doit permettre la cohabitation d'entités situées dans des environnements distincts, de l'autre elle doit également fournir le support aux interactions entre chacune de ces entités, humaines ou agents.

Du point de vue de l'architecture des SMA, Weyns et al. ont montré comment interfacer le SMA au monde réel via une architecture à trois couches [61, 64]. Ces trois couches sont l'environnement physique, non en tant qu'espace physique mais qu'infrastructure informatique matérielle, la plateforme d'exécution (logicielle), et la couche applicative multi-agent.

Du point de vue des systèmes mixtes, Rinçon et al. [48] proposent une architecture permettant la mise en œuvre de SMA dans un contexte cyber-physique en réifiant les entités dotées d'une présence physique virtuelle et celles n'en disposant pas au sein d'un environnement virtuel informé. Nous pouvons également citer des travaux tels que

[50] dont l'objectif est d'utiliser le dialogue naturel pour interfacier le système multi-agent, fournisseur de services, aux humains souhaitant les utiliser ; ainsi que [25] qui utilise la notion d'émotions artificielles pour la résilience des systèmes cyber-physiques.

Ces deux familles de travaux s'inscrivent dans des perspectives semblables puisqu'elles mettent en lumière la façon d'intégrer des systèmes multi-agents pour supporter l'activité humaine. Cependant, les agents ne sont pas incarnés dans ces propositions, ce qui laisse ouverte la question de l'interfaçage supplémentaire de la modalité incarnée des agents dans les communautés mixtes, indépendamment du type d'incarnation.

Ceci est également illustré par les travaux dans lesquels tous les agents sont incarnés dans un environnement virtuel, par exemple dans [59], où l'environnement physique réel n'est pas pris en compte.

De façon à résoudre cette difficulté, il est nécessaire d'intégrer une séparation entre l'incarnation (robot, agent virtuel) et le raisonnement, de façon à ce que cette seconde partie puisse être conçue de façon générique. Une approche à base de composants telle que [53], a vocation à permettre la réutilisabilité d'une partie des modules de raisonnement quels que soient les capteurs et effecteurs. Nous pouvons également citer les notions de *smart body* [57] et de *shell* [18] qui séparent la partie raisonnement de celle liée à l'incarnation.

Cette séparation, qui concerne la conception de l'agent, peut également être reflétée dans l'environnement agent pour envisager la présence dans l'environnement partagé de façon transparente quel que soit le type d'entité de la communauté mixte (humain, agent virtuel ou robot). La gestion de cette co-présence peut être envisagée sous forme de *corps logiciels* [40, 51] indépendants de la représentation incarnée. Le rôle principal de ces artefacts [44] est d'interfacier dans l'environnement logique les entités entre elles, par la médiation de l'observabilité des entités.

De façon à gérer la cohabitation entre entités virtuelles et physiques, et fournir des services intelligents aux utilisateurs, des travaux récents [43, 32, 70] utilisent le concept de *monde-miroir* [17]. Un monde-miroir est un environnement virtuel qui est déterminé par le monde réel : chaque entité qui peut être observée et sur laquelle il est possible d'agir dans le monde physique possède un alter-égo dans le monde numérique, que les agents peuvent observer et sur lesquels ils peuvent agir. Dans [32], les auteurs proposent un modèle en 3 phases, où le monde réel est tout d'abord virtualisé, puis ce modèle est augmenté par le biais d'interactions entre humains et agents dans l'environnement virtuel, avant d'utiliser le résultat en réalité augmentée. À contre-pied de cette approche, certains travaux [70] inversent la logique en envisageant une réalité augmentée inversée, où c'est le monde virtuel qui est enrichi d'éléments du monde réel.

Nous montrons en figure 3.1 une façon d'articuler ces différents éléments. Dans le monde réel cohabitent humains et robots. Les agents conversationnels animés ont également une représentation dans le monde physique, de façon à in-

teragir avec les humains, mais ils sont situés dans un monde virtuel. La mise en relation des deux dimensions constitutives des communautés mixtes implique de réifier les deux environnements dans un système symbolique commun, lequel constitue le monde logique. Chaque entité du système possède une interface, qui sert à représenter son état et médier ses interactions, dans ce monde logique et plus particulièrement dans l'environnement agent. Ces interfaces s'inscrivent ainsi dans la notion de corps logiciel évoquée plus haut.

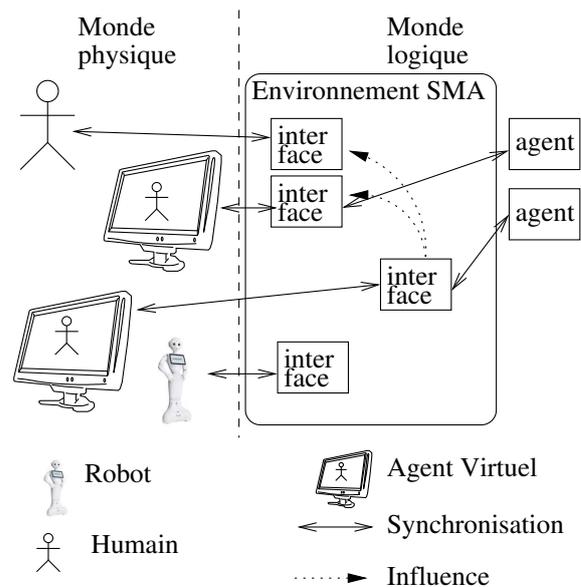


FIGURE 1 – Vue d'ensemble

Il est à noter que les humains doivent également posséder une représentation dans le monde logique, de façon à gérer leur observabilité de la même façon que les autres entités de la communauté. La génération et le suivi de cette représentation s'appuie sur les capteurs de l'environnement par exemple via une approche multi-agent utilisant la notion d'*artefacts* [9, 47]. Une synchronisation est donc réalisée entre les éléments du monde physique/virtuel et leurs contreparties du monde logique.

Via leur interface, les agents peuvent exercer une influence sur celles des autres éléments du monde logique. Nous utilisons le terme d'influence au sens de [34], *i.e.* que les actions des agents n'ont pas nécessairement d'effet, ou l'effet escompté, sur les autres entités du système.

Dans ce système, chaque incarnation agent virtuel et robot peut être pilotée par un ou plusieurs agent(s) autonome(s) via cette interface dont l'objectif est d'uniformiser l'ensemble des entités de la communauté virtuelle afin de pouvoir bénéficier des mécanismes génériques du système.

3.2 Modèles d'interaction multipartie

Les travaux dans les domaines des agents virtuels ou robotique n'ont que peu abordé le cas où un nombre n d'agents interagit avec un nombre m d'humains, avec n et m strictement supérieurs à 1, ce qui implique un cer-

tain nombre d’adaptations [58]. La différence la plus fondamentale entre interactions dyadiques et multiparties tient au rôle des participants au cours du dialogue : les participants peuvent être destinataires ou non des énoncés, individuellement ou en groupe [27], ce qui implique des attentes différentes vis-à-vis du déroulement du dialogue. Dans ce contexte, les différents mécanismes du dialogue, tels que la prise de tour de parole, la détection du ou des destinataires des énoncés ou la gestion du regard sont par exemple peu étudiés et lorsqu’ils le sont les moyens mis en œuvre sont souvent minimalistes, comme en témoigne le faible nombre de corpus d’interactions multiparties [15, 22, 33]. Plusieurs travaux étudient des situations où un humain interagit avec plusieurs compagnons, comme [8]. Cependant, les seules interactions entre compagnons artificiels sont scriptées, il s’agit donc d’une suite d’interactions dyadiques et non d’une interaction multipartie simultanée. D’autres travaux étudient le cas où un robot ou agent interagit avec plusieurs humains. Si le robot n’est pas explicitement prévu pour gérer ce type de situation, le résultat est soit incohérent, soit une succession d’interactions dyadiques [4].

Deux approches principales ont été proposées dans la littérature pour résoudre certaines tâches telles que la détection des destinataires : la première est fondée sur l’application de règles [59], tandis que la seconde exploite l’apprentissage automatique [30]. Notons que des travaux récents mentionnent explicitement la notion de systèmes multi-agents pour représenter les situations de dialogues multiparties avec plusieurs agents virtuels [24].

3.3 Mécanismes multiparties en environnements physique et réel

De façon à maintenir la cohérence entre les comportements des agents incarnés et ce qui se passe dans le monde physique, l’environnement multi-agent doit refléter ce dernier tant dans le maintien d’un état du monde pertinent que dans les mécanismes et dynamiques qui régissent celui-ci.

Deux catégories de mécanismes multiparties entrent dans ce cadre : 1. les mécanismes simulant les règles de l’environnement réel pour les agents de façon à obtenir des comportements homogènes, et 2. ceux impliquant directement à la fois des humains et des agents selon des modalités spécifiques.

Pour illustrer la première catégorie, nous pouvons citer la problématique des communications multiparties. Par nature, les communications humaines s’effectuent à la cantonade, autrement dit la perception des signaux dépend uniquement soit de la proximité par rapport à l’émetteur et de la force du signal, soit de l’utilisation d’une médiation.

Dans un contexte de réalité mixte, l’application de ces règles aux agents grâce à l’environnement multi-agent permet d’assurer la localité des échanges. Plus généralement, tous les mécanismes de “bas niveau” (perception, action) doivent simuler, dans leurs contraintes, l’environnement partagé, par exemple par des mécanismes de perception active [65].

Un second aspect à prendre en considération est l’obser-

vabilité des interactions. Si les agents incarnés n’ont pas besoin d’échanger de messages oraux pour communiquer, un changement de comportement dont l’origine n’est pas observable du point de vue des utilisateurs humains crée un problème d’appréhension du système et d’explicitabilité des comportements. De la même façon, pour le travail d’équipe, Lopez *et al.* [29] montre que le dialogue (énoncé des états mentaux des agents au fur et à mesure du déroulé de la tâche) permet aux utilisateurs et agents de mieux coopérer. Il est donc nécessaire de représenter tout ou partie des interactions, y compris purement entre agents, dans le monde physique.

La seconde catégorie implique directement agents et humains, par exemple dans la contagion émotionnelle. Hatfield *et al.* [20] ont montré que la contagion émotionnelle se situe à un niveau de conscience significativement plus bas que l’empathie, grâce à des processus automatiques, *i.e.* non contrôlés. Ces processus automatiques sont à la fois perceptifs et actifs, notamment via les comportements d’imitation, qui font partie des normes sociales du dialogue. Dans ce cas, il convient de considérer différemment la nature des interfaces selon la nature de l’entité considérée. Le tableau 2 montre ainsi les différents cas liés au caractère situé des interactions.

source \ cible	Humain	Agent
Humain	Env. physique Trace	Env. physique Interface
Agent	Incarnation	Env. Logique incarnation

TABLE 2 – Modalités d’interaction

Les utilisateurs interagissent dans l’environnement physique, mais leur activité doit être reflétée sous forme de trace même lorsqu’elle ne concerne pas directement les agents, de façon à leur permettre d’avoir conscience du contexte de la situation. Lorsqu’ils interagissent avec des agents, l’interface qui les représente dans l’environnement logique est modifiée en conséquence. A l’inverse, un agent interagissant avec un humain passe nécessairement par le biais de son incarnation, tandis que les interactions entre agents se situent dans l’environnement logique, éventuellement reflété via leurs représentations du monde physique.

La mise en place de mécanismes transparents, quelles que soient les entités en jeu, implique ainsi trois nécessités : la capacité à capter le monde réel pour le représenter dans l’environnement logique, la capacité de simuler ces mécanismes et la capacité à les représenter au sein de l’environnement physique.

3.4 Synthèse

La table 3 récapitule les différents travaux évoqués dans cet état de l’art en fonction des critères donnés en section 2.5. Il est à noter que les systèmes permettant à plusieurs humains d’interagir sont en général ouverts, *i.e.* permettent à des participants d’entrer ou sortir de la communauté.

Référence(s)	Entités	Espace(s)	Interactions dialogiques	Mécanismes multiparties	Comportement individualisé
[2, 24, 29, 59]	1 humain, 1+ agents virtuels	virtuel	oui	oui	oui
[3, 50]	1+ humains, 1+ agents non incarnés	physique	oui	oui	non
[48]	1+ humains, 1+ agents non incarnés	physique	oui	non	non
[4]	1+ humains, 1 robot	physique	oui	non	non
[8]	1 humain, 1+ agents virtuels, 1+ robots	physique	oui	non	oui
[9]	1 agent non incarné par humain	physique, augmenté	textuelles	non	non
[12, 38]	1 humain, 1 agent virtuel	virtuel	oui	non	oui
[15]	3 humains, 1 robot	physique	oui	oui	non
[18]	1 humain, 1 agent avec 2 incarnations	physique et virtuel	non	non	non
[19]	1 humain, 1 agent virtuel	augmenté	oui	non	oui
[32, 43, 70]	1 humain, 1+ agents virtuels	physique, virtuel, augmenté	possibles	oui	non
[69]	1+ humains et 1+ agents virtuels ou 1+ robots	physique, virtuel	oui	oui	oui

TABLE 3 – Tableau récapitulatif de l'état de l'art. Un nombre fixe d'humains ou agents dénote un système fermé, tandis que 1+ dénote la possibilité d'en intégrer plusieurs.

4 Éthique et acceptabilité

De façon à concevoir des communautés mixtes responsables d'un point de vue éthique, il est nécessaire de distinguer trois angles [13] :

l'éthique dans la conception (*ethics in design*), qui s'intéresse à la prise en compte de critères éthiques liés au fonctionnement et à la finalité du système ;

l'éthique pour la conception (*ethics for design*), qui couvre les acteurs et les pratiques de développement ;

l'éthique par conception (*ethics by design*), qui étudie le comportement du système lui-même, de façon explicite (le système est capable de raisonner sur des concepts éthiques et de les prendre en compte dans ses décisions) ou implicite (les règles de comportement ont été conçues pour être éthiques).

Dans cette section, nous nous intéressons principalement au premier angle, l'éthique dans la conception, pour discuter les implications à la fois des compagnons artificiels eux-mêmes et des moyens mis en œuvre pour ces communautés humains-agents. Cette intégration de la dimension éthique dès la conception se décline selon 4 critères [14] : a) l'utilité, qui se rapporte à la finalité du système et son adéquation aux besoins des utilisateurs ; b) l'utilisabilité, qui correspond à l'efficacité du système à achever son but ; c) l'acceptabilité, qui est l'adéquation avec les pratiques et le système de valeurs des utilisateurs ; et enfin d) l'appropriabilité, qui représente la possibilité d'intégrer un dispositif dans le fonctionnement d'un humain.

En fonction de leur tâche (et donc de leur rôle assigné) et

de leur public, il existe des différences notables d'acceptabilité des compagnons artificiels. D'un côté du spectre, les agents sont parfois considérés comme plus acceptables que des humains mis dans la même position, car les utilisateurs les supposent moins aptes à les juger [12]. Dans le projet NARECA, dans lequel un agent interactif narre une histoire à un enfant, même des versions dégradées des avatars (sans prosodie ou sans expression faciale) n'empêchent pas les enfants d'être engagés dans l'interaction. Dans le projet MoCA [8], où les robots et agents virtuels accompagnent des enfants dans différentes activités, les enfants semblent moins réticents que les adultes à nouer une relation et trouver utiles les compagnons. À l'inverse, les adultes acceptent plus facilement l'idée d'utiliser ces compagnons pour des tâches quotidiennes (ménagères, ludiques) que pour leur potentiel amical ou pour s'occuper des enfants [10].

Pour les personnes âgées [66], si des critères individuels tels que la personnalité des utilisateurs ont un impact sur l'acceptabilité des compagnons artificiels, certains critères de conception jouent un rôle important : communication naturelle et capacité à répondre aux besoins émotionnels, psychologiques et sociaux de façon personnalisée. Un second aspect est lié à l'appropriabilité : pour intégrer un compagnon dans son fonctionnement quotidien, se ressent le besoin de pouvoir avoir confiance dans celui-ci, et de pouvoir le contrôler.

Ces éléments montrent l'importance de la sécurité des données et de leur usage. Concernant la privacité des données, de nombreuses pratiques douteuses d'un point de vue éthique de la part des entreprises entament la confiance des

utilisateurs. Les assistants vocaux captent en permanence les données audio des utilisateurs, et peuvent par erreur ou par choix [21] transmettre des enregistrements non sollicités. Le même type de problème a été détecté avec des objets connectés fournissant des données d'usage à leur constructeur sans en informer les clients, y compris dans le cas de données sensibles, par exemple dans le cas de vibromasseurs transmettant en temps réel leur état, associé aux données personnelles des utilisateurs [67]. Plus prosaïquement, les agents conversationnels utilisent souvent des API externes de type GoogleSpeech pour la transcription automatique, ce qui implique que les données de conversation ne sont plus maîtrisées par le concepteur. Pour répondre à ces difficultés, une approche de protection des données par conception semble la plus pertinente [60], via des mesures techniques (par exemple localité des informations) et de contrôle (informer et donner la possibilité aux utilisateurs de choisir quelles données sont utilisées, et pour quel usage).

La dimension de contrôle passe également par l'explicitabilité des comportements [35, 14] : comme nous l'avons vu, les humains ont la capacité de reconstruire un raisonnement à partir de comportements observés ; mais d'une part cette reconstruction peut être erronée, et d'autre part certaines décisions sont complexes et nécessitent d'explicitier le processus, de façon à ce que l'être humain le comprenne et éventuellement agisse sur le système pour le maîtriser.

Nous pouvons également noter que les recherches sur les compagnons robotiques utilisent souvent des mises en situation de type magicien d'Oz (e.g. [38]), dans lesquelles les humains interagissent avec un agent qu'ils croient autonome alors que celui-ci est piloté par un expérimentateur. Des travaux comme SimSensei [12], où les utilisateurs se confient à un agent autonome en pensant ne pas être jugé par un autre humain alors que le protocole expérimental est contradictoire avec cette attente, doivent être soumis à une analyse éthique.

Enfin, l'utilisation de compagnons robotiques pose également un certain nombre de problèmes dans leur dimension applicative [54], c'est à dire en terme d'utilité : le risque des applications de type santé ou soin est un remplacement ou une réduction du contact humain, une infantilisation des utilisateurs, l'absence de contrôle et de confidentialité. Si la notion de communauté mixte ne présente pas de difficultés éthiques spécifiques, leur mise en œuvre invite, de par la présence d'êtres humains en leur sein, à prendre en compte dès la conception les aspects de sécurité et confidentialité des données, ainsi que les conséquences des applications considérées.

5 Conclusion

Dans cet article, nous avons souligné les différentes problématiques liées à la mise en place de communautés mixtes, incluant plusieurs humains et agents incarnés, et esquissé quelques pistes de réflexion pour la mise en œuvre de mécanismes multi-agents dans ces communautés.

Au delà du besoin de repenser les interactions traditionnel-

lement dyadiques, issues des travaux sur les compagnons artificiels et agents virtuels, vers un contexte multipartite et ouvert, la mise en place de tels environnements peut s'appuyer sur un ensemble d'outils tels que les artefacts [44]. Une spécification importante de ces systèmes est la contrainte du temps réel, ce qui implique de considérer également les problèmes classiques de conception logicielle (efficacité, distribution...).

Un autre aspect important est celui de l'acceptabilité de telles communautés mixtes : d'un côté, la présence d'humains implique que le système et son fonctionnement soient explicables et donc transparents ; de l'autre il nécessite une capacité de contrôle des données sensibles ainsi qu'une conception respectueuse de la vie privée. Au delà de l'acceptabilité des communautés mixtes, une étude de l'éthique par conception, prenant en compte la dimension éthique des comportements des agents de façon explicite ou implicite serait à mener.

Enfin, si cet article s'est focalisé sur les communautés mixtes, il est nécessaire d'envisager d'autres cas dans lesquels tous les agents ne sont pas incarnés [3]. Une réflexion doit alors être menée sur les modes d'interaction entre agents non incarnés, agents incarnés et utilisateurs qui puissent être intégrés à un même système.

Références

- [1] P. Agre and D. Chapman. Pengi : An implementation of a theory of activity. volume Vol. 87, pages 268–272, 01 1987.
- [2] M. Barange, J. Saunier, and A. Pauchet. Pedagogical Agents as Team Members : Impact of Proactive and Pedagogical Behavior on the User. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 791–800, Unknown, Unknown Region, 2017.
- [3] H. Billhardt, V. Julián, J. M. Corchado, and A. Fernández. An architecture proposal for human-agent societies. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 344–357. Springer, 2014.
- [4] D. Bohus, C. W. Saw, and E. Horvitz. Directions robot : in-the-wild experiences and lessons learned. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 637–644, 2014.
- [5] H. Branigan. Perspectives on multi-party dialogue. *Research on Language & Computation*, 4 (2-3) :153–177, 2006.
- [6] S. Campano, N. Sabouret, E. De Sevin, and V. Corruble. An evaluation of the cor-e computational model for affective behaviors. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 745–752, 2013.
- [7] P. Carruthers and P. K. Smith. *Theories of theories of mind*. Cambridge University Press, 1996.
- [8] M. Courgeon, C. Hoareau, and D. Duhaut. Interaction with artificial companions : Presentation of an exploratory study. In *International Conference on Social Robotics*, pages 611–620. Springer, 2016.
- [9] A. Croatti, S. Montagna, and A. Ricci. A personal medical digital assistant agent for supporting human operators in emergency scenarios. In *Agents and multi-agent systems for health care*, pages 59–75. Springer, 2017.

- [10] K. Dautenhahn, S. Woods, C. Kaouri, M. L. Walters, K. L. Koay, and I. Werry. What is a robot companion-friend, assistant or butler? In *2005 IEEE/RSJ international conference on intelligent robots and systems*, pages 1192–1197. IEEE, 2005.
- [11] D. David, S.-A. Matu, and O. A. David. Robot-based psychotherapy : Concepts development, state of the art, and new directions. *International Journal of Cognitive Therapy*, 7(2) :192–210, 2014.
- [12] D. DeVault, R. Artstein, G. Benn, T. Dey, E. Fast, A. Gainer, K. Georgila, J. Gratch, A. Hartholt, M. Lhommet, et al. Simsensei kiosk : A virtual human interviewer for healthcare decision support. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1061–1068, 2014.
- [13] V. Dignum. *Responsible Artificial Intelligence : How to Develop and Use AI in a Responsible Way*. Springer International Publishing, 2019.
- [14] M. Fréjus and D. Lahoual. Conception d'interactions éthiques et durables entre l'humain et les systèmes d'intelligence artificielle. le cas de l'expérience vécue des usagers de l'ia vocale. *Revue d'intelligence artificielle*, 32, 12 2018.
- [15] K. Funakoshi and M. Komuro. Toward mindful machines : A wizard-of-oz dialogue study. In *2019 IEEE International Conference on Humanized Computing and Communication (HCC)*, pages 94–99. IEEE, 2019.
- [16] D. Gatica-Perez. Automatic nonverbal analysis of social interaction in small groups : A review. *Image and vision computing*, 27(12) :1775–1787, 2009.
- [17] D. Gelernter. *Mirror worlds : or : the day software puts the universe in a shoebox... How it will happen and what it will mean*. Oxford University Press, 1993.
- [18] P. F. Gomes, E. M. Segura, H. Cramer, T. Paiva, A. Paiva, and L. E. Holmquist. Vipleo and phypleo : Artificial pet with two embodiments. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, pages 1–8, 2011.
- [19] A. Hartholt, S. Mozgai, E. Fast, M. Liewer, A. Reilly, W. Whitcup, and A. Rizzo. Virtual humans in augmented reality : A first step towards real-world embedded virtual roleplayers. In *Proceedings of the 7th International Conference on Human-Agent Interaction, HAI '19*, page 205–207, New York, NY, USA, 2019. Association for Computing Machinery.
- [20] E. Hatfield, J. Cacioppo, and R. Rapson. *Emotional contagion*. Cambridge Univ Pr, 1994.
- [21] M. B. Hoy. Alexa, siri, cortana, and more : an introduction to voice assistants. *Medical reference services quarterly*, 37(1) :81–88, 2018.
- [22] N. Jovanovic, R. o. d. Akker, and A. Nijholt. A corpus for studying addressing behaviour in multi-party dialogues. *LREC'06*, 40(1) :5–23, 2006.
- [23] K. Kamali, X. Fan, and J. Yen. Towards a theory for multiparty proactive communication in agent teams. *International Journal of Cooperative Information Systems*, 16(02) :271–298, 2007.
- [24] R. B. Kantharaju, A. Pease, D. Reidsma, C. Pelachaud, M. Snaith, M. Bruijnes, R. Klaassen, T. Beinema, G. Huijing, D. Simonetti, et al. Integrating argumentation with social conversation between multiple virtual coaches. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, pages 203–205, 2019.
- [25] E. Kouicem, C. Raievsky, and M. Occello. Towards a cyber-physical systems resilience approach based on artificial emotions and multi-agent systems. In *The 12th International Conference on Agents and Artificial Intelligence*, 2019.
- [26] N. C. Krämer, A. von der Pütten, and S. Eimler. Human-agent and human-robot interaction theory : similarities to and differences from human-human interaction. In *Human-computer interaction : The agency perspective*, pages 215–240. Springer, 2012.
- [27] S. Kumar, M. J. Huber, D. R. McGee, P. R. Cohen, and H. J. Levesque. Semantics of agent communication languages for group interaction. In *AAAI/IAAI*, pages 42–47, 2000.
- [28] A. T. Lopes, E. de Aguiar, A. F. De Souza, and T. Oliveira-Santos. Facial expression recognition with convolutional neural networks : coping with few data and the training sample order. *Pattern Recognition*, 61 :610–628, 2017.
- [29] T. Lopez, P. Chevallier, V. Gouranton, P. Evrard, F. Nouviale, M. Barange, R. Bouville, and B. Arnaldi. Collaborative virtual training with physical and communicative autonomous agents. *Computer Animation and Virtual Worlds*, 25(3-4) :485–493, 2014.
- [30] U. Malik, M. Barange, N. Ghannad, J. Saunier, and A. Pauchet. A generic machine learning based approach for addressee detection in multiparty interaction. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents, IVA '19*, page 119–126, New York, NY, USA, 2019. Association for Computing Machinery.
- [31] K. I. Manktelow and D. E. Over. *Rationality : Psychological and philosophical perspectives*. Taylor & Frances/Routledge, 1993.
- [32] S. Mariani, A. Croatti, A. Ricci, A. Prati, and G. Vizzari. Vitalise : Virtual to augmented loop in smart environments. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2114–2116, 2019.
- [33] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, et al. The ami meeting corpus. In *Proc. of the 5th International Conference on Methods and Techniques in Behavioral Research*, volume 88, page 100, 2005.
- [34] F. Michel. Le modèle irm4s, de l'utilisation des notions d'influence et de réaction pour la simulation de systèmes multi-agents. *Revue d'Intelligence Artificielle*, 21(5-6) :757–779, 2007.
- [35] S. T. Mueller, R. R. Hoffman, W. Clancey, A. Emrey, and G. Klein. Explanation in human-ai systems : A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable ai. Technical report, Florida Institute for Human and Machine Cognition Pensacola United States, 2019.
- [36] R. Niewiadomski, M. Ochs, and C. Pelachaud. Expressions of empathy in ecas. In *IVA '08 : Proceedings of the 8th international conference on Intelligent Virtual Agents*, pages 37–44, Berlin, Heidelberg, 2008. Springer-Verlag.

- [37] I. Olaronke, O. Oluwaseun, and I. Rhoda. State of the art : a study of human-robot interaction in healthcare. *International Journal of Information Engineering and Electronic Business*, 9(3) :43, 2017.
- [38] A. Pauchet, O. Şerban, M. Ruinet, A. Richard, É. Chanoni, and M. Barange. Interactive narration with a child : Avatar versus human in video-conference. In *International Conference on Intelligent Virtual Agents*, pages 343–346. Springer, 2017.
- [39] R. Picard. Affective computing. 1995.
- [40] E. Platon, N. Sabouret, and S. Honiden. Tag interactions in multiagent systems : Environment support. In *Proceedings of Environment for Multi-Agent Systems, Workshop held at the Fifth Joint Conference in Autonomous Agents and Multi-Agent Systems*, volume 4389 of *Lecture Notes in Artificial Intelligence*, pages 106–123. Springer Verlag, 2007.
- [41] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4) :515–526, 1978.
- [42] A. Ricci, A. Omicini, M. Viroli, L. Gardelli, and E. Oliva. Cognitive stigmergy : Towards a framework based on agents and artifacts. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *Proceedings of Environment for Multi-Agent Systems, Workshop held at the fifth AAMAS conference*, volume 4389 of *LNAI*, pages 124–140. Springer Verlag, 2007.
- [43] A. Ricci, M. Piunti, L. Tummolini, and C. Castelfranchi. The mirror world : Preparing for mixed-reality living. *IEEE Pervasive Computing*, 14(2) :60–63, 2015.
- [44] A. Ricci, M. Piunti, M. Viroli, and A. Omicini. Environment programming in cartago. In *Multi-Agent Programming* :, pages 259–288. Springer, 2009.
- [45] D. Richards. Agent-based museum and tour guides : applying the state of the art. In *Proceedings of The 8th Australasian Conference on Interactive Entertainment : Playing the System*, pages 1–9, 2012.
- [46] D. Richards and V. Dignum. Supporting and challenging learners through pedagogical agents : Addressing ethical issues through designing for values. *British Journal of Educational Technology*, 50(6) :2885–2901, 2019.
- [47] J. Rincon, J.-L. Poza-Lujan, V. Julian, J.-L. Posadas-Yagüe, and C. Carrascosa. Extending mam5 meta-model and jacobiv e framework to integrate smart devices from real environments. *PLoS one*, 11(2), 2016.
- [48] J. A. Rincon, V. Julian, and C. Carrascosa. Developing an emotional-based application for human-agent societies. *Soft Computing*, 20(11) :4217–4228, 2016.
- [49] S. Rosenschein. Formal theories of knowledge in ai and robotics. *New Generation Comput.*, 3 :345–357, 12 1985.
- [50] A. Sanchis, V. Julián, J. M. Corchado, H. Billhardt, and C. Carrascosa. Using natural interfaces for human-agent immersion. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 358–367. Springer, 2014.
- [51] J. Saunier, C. Carrascosa, S. Galland, and P. S. Kanmeugne. Agent bodies : An interface between agent and environment. In *Agent Environments for Multi-Agent Systems IV*, pages 25–40. Springer, 2015.
- [52] J. Saunier and H. Jones. Mixed agent/social dynamics for emotion computation. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 645–652. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [53] O. Şerban and A. Pauchet. Agentslang : A fast and reliable platform for distributed interactive systems. In *2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 35–42. IEEE, 2013.
- [54] A. Sharkey and N. Sharkey. Granny and the robots : ethical issues in robot care for the elderly. *Ethics and information technology*, 14(1) :27–40, 2012.
- [55] M. Soleymani, D. Garcia, B. Jou, B. Schuller, S.-F. Chang, and M. Pantic. A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65 :3 – 14, 2017. Multimodal Sentiment Analysis and Mining in the Wild Image and Vision Computing.
- [56] W. R. Swartout, J. Gratch, R. W. Hill Jr, E. Hovy, S. Marsella, J. Rickel, and D. Traum. Toward virtual humans. *AI Magazine*, 27(2) :96–96, 2006.
- [57] M. Thiebaut, S. Marsella, A. Marshall, and M. Kallmann. Smartbody : Behavior realization for embodied conversational agents. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 151–158, 2008.
- [58] D. Traum. Issues in multiparty dialogues. In F. Dignum, editor, *Advances in Agent Communication*, pages 201–211. Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [59] D. Traum and J. Rickel. Embodied agents for multi-party dialogue in immersive virtual worlds. In *AAMAS '02 : Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 766–773. ACM Press, 2002.
- [60] M. Veale, R. Binns, and J. Ausloos. When data protection by design and data subject rights clash. *International Data Privacy Law*, 8(2) :105–123, 2018.
- [61] M. Viroli, T. Holvoet, A. Ricci, K. Schelfhout, and F. Zambonelli. Infrastructures for the environment of multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1) :49–60, 2007.
- [62] P. Wagner, Z. Malisz, and S. Kopp. Gesture and speech in interaction : An overview. *Speech Communication*, 57 :209 – 232, 2014.
- [63] D. Weyns and T. Holvoet. Architectural design of a situated multiagent system for controlling automatic guided vehicles. *International Journal of Agent-Oriented Software Engineering*, 2(1) :90–128, 2008.
- [64] D. Weyns, A. Omicini, and J. Odell. Environment as a first-class abstraction in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1) :5–30, Feb. 2007. Special Issue on Environments for Multi-agent Systems.
- [65] D. Weyns, E. Steegmans, and T. Holvoet. Towards active perception in situated multi-agent systems. *Special Issue of Journal on Applied Artificial Intelligence*, 18 (9-10) :867–883, 2004.
- [66] S. Whelan, K. Murphy, E. Barrett, C. Krusche, A. Santorelli, and D. Casey. Factors affecting the acceptability of social robots by older adults including people with dementia or cognitive impairment : a literature review. *International Journal of Social Robotics*, 10(5) :643–668, 2018.

- [67] M. Wynn, K. Tillotson, R. Kao, A. Calderon, A. Murillo, J. Camargo, R. Mantilla, B. Rangel, A. A. Cardenas, and S. Rueda. Sexual intimacy in the age of smart devices : Are we practicing safe iot? In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, pages 25–30, 2017.
- [68] D. Yu and L. Deng. *Automatic Speech Recognition*. Springer, 2016.
- [69] Z. Yumak and N. Magnenat-Thalmann. Multimodal and multi-party social interactions. In *Context Aware Human-Robot and Human-Agent Interaction*, pages 275–298. Springer, 2016.
- [70] Z. Zhang, D. Weng, H. Jiang, Y. Liu, and Y. Wang. Inverse augmented reality : A virtual agent's perspective. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 154–157, 2018.

Émergence de comportements collectifs basée sur l'apprentissage progressif individuel

François Suro
suro@lirmm.fr

Jacques Ferber
ferber@lirmm.fr

Tiberiu Stratulat
stratulat@lirmm.fr

Fabien Michel
fmichel@lirmm.fr

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Résumé

Dans la perspective d'un développement ouvert et continu il est crucial qu'un comportement final souhaité à un instant donné puisse être un élément pour la création future de comportements plus complexes, dont le but ne peut être anticipé. L'acquisition de nouveaux comportements est déclenchée par l'apparition de nouveaux besoins et problèmes résultants de l'interaction avec l'environnement, mais aussi à travers l'interaction avec d'autres agents.

Dans cet article nous soumettons l'architecture développementale MIND à un problème de coordination multi-agents. A partir de comportements établis individuellement, nous ferons apprendre à un groupe d'agents, au moyen d'un curriculum établi par un instructeur humain, un comportement collectif, ainsi que l'utilisation alternante de comportements individuels et collectifs afin d'accomplir un but commun.

Mots-clés : Robotique développementale, Apprentissage agent, Architecture modulaire, Architecture hiérarchique, Apprentissage par Curriculum, Coordination Multi-Agents

Abstract

In the perspective of open ended and continuous development it is crucial that a final desired behaviour at a given moment can be the starting point for the future creation of more complex behaviours, the purpose of which cannot be anticipated. The acquisition of new behaviours is triggered by the rise of new needs and problems resulting from interaction with the environment, but also through interaction with other agents.

In this paper we subject the MIND developmental architecture to a problem of multi-agents coordination. Starting from individually established behaviors, we will teach a group of agents, through a curriculum established by a human instructor, a collective behavior, as well as the alternating use of individual and collective behaviors in order to accomplish a common goal.

Keywords: Developmental Robotics, Agent Learning, Modular Architecture, Hierarchical Architecture, Curriculum-Based Learning, Multi-Agents Coordination

1 Introduction

Inspirée par la théorie de Jean Piaget [29][30] sur le développement cognitif chez l'homme, la recherche sur la robotique développementale [21] vise l'émergence d'une intelligence artificielle générale au moyen d'un processus progressif qui accumule des compétences d'une complexité croissante.

Ce processus dynamique de schémas de

coordination, depuis les schémas sensorimoteurs jusqu'aux opérations de niveau abstrait, est depuis longtemps compris comme une structure hiérarchique [23]. L'organisation, l'émergence et la régulation d'une telle structure, avec la robustesse nécessaire pour un développement à long terme, reste une question fondamentale du domaine de la robotique développementale [27].

Dans nos travaux antérieurs nous avons proposé une solution à cette question d'organisation sous la forme de l'architecture MIND [35]. MIND (Modular Influence Network Design) est une architecture qui organise des modules de compétences (*skills*) en un réseau contrôlé par un mécanisme d'influence. MIND permet d'ajouter de nouveaux modules de compétences dont le rôle est de coordonner des modules plus simples pour obtenir un comportement complexe. Nous avons montré que grâce à cette architecture, un agent individuel est capable de transposer un curriculum établi par un instructeur en une hiérarchie de comportements correspondants.

Dans l'introduction au premier Workshop international sur la robotique épigénétique [38], un précurseur du domaine général de la robotique développementale, les auteurs soulignent l'importance des interactions sociales parmi les interactions menant au développement cognitif. Qu'il s'agisse du développement personnel, par instruction ou imitation, ou bien du développement d'une organisation sociale à travers les rôles individuels.

Dans la perspective de développer MIND comme architecture supportant le développement d'agents intelligents, nous souhaitons intégrer pleinement l'aspect social de l'apprentissage et des comportements résultants, que ce soit dans la coordination d'essaim de robots (swarm robotics) ou la communication entre agents artificiels (ou biologiques).

Dans cet article, nous expérimentons avec l'architecture MIND dans un contexte de coordination multi-agents. En nous basant sur une hiérarchie apprise individuellement, nous apprenons à un groupe d'agents de nouveaux comportements collectifs se basant sur des

comportements individuels. Nous montrons que le principe de conception modulaire de MIND fournit une grande flexibilité dans la coordination de comportements individuels et collectifs.

Après un bref état de l'art à la section 2 nous présenterons l'architecture MIND dans la section 3, puis nous décrirons dans la section 4 le contexte expérimental et dans la section 5 les résultats de la tâche de fourragement multi-agents que nous souhaitons apprendre. En conclusion nous discuterons des futurs travaux multi-agents que nous réaliserons à l'aide de MIND.

2 Contexte

Apprentissage et agents Contrairement à d'autres domaines axés sur l'apprentissage, la robotique développementale vise à créer des systèmes artificiels avec des compétences qui vont au-delà de l'apprentissage d'une seule tâche. *"La recherche d'un système d'apprentissage multitâche flexible, autonome et ouvert est, par essence, une ré-instanciation particulière de la recherche de longue date sur l'IA à usage général (general purpose AI)"* [21]. La robotique développementale, inspirée par l'être humain, vise à l'émergence de l'IA par un apprentissage progressif et continu sous les contraintes des agents incarnés.

La robotique développementale est un domaine de recherche hautement interdisciplinaire, s'inspirant de domaines autres que l'informatique et la robotique, telles que les neurosciences et la psychologie. Mais au sein même de l'informatique, la robotique développementale s'appuie sur de nombreuses disciplines telles que les algorithmes d'apprentissage, structures de données, exploration, communication entre systèmes, traitement du signal, etc.

Apprentissage par Curriculum Bien qu'utilisé implicitement depuis les années 90, l'apprentissage par Curriculum [2] s'est vraiment établi grâce aux avancées dans les domaines du reinforcement learning (RL) et du transfer learning [36]. Cette méthode d'apprentissage progressive très étudiée à ce jour dans le domaine du machine learning vient compléter les techniques de RL traditionnelles et permet de résoudre et d'accélérer l'apprentissage de problèmes autrement impossibles. En subdivisant une tâche d'apprentissage en sous-tâches différentes mais complémentaires (Source tasks), les différents signaux de Feedback (ou récompense) sont séparés et chaque sous-tâche est beaucoup plus simple à apprendre. Cette technique a été appliquée avec succès à des problèmes de robotique et de jeux vidéo [25]. D'autre part l'utilisation d'un comportement simple comme point de départ pour

apprendre un comportement complexe aide également dans l'aspect exploratoire de l'apprentissage, en se concentrant sur la complexité supplémentaire du nouvel environnement associé à la tâche.

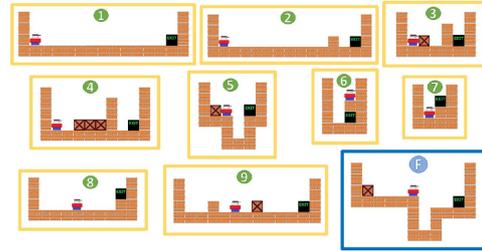


Fig. 1 – La compétence finale F (encadré en bleu) est apprise en transférant toutes les compétences apprises précédemment, telles qu'atteindre la sortie(1), sauter sur un bloc(2), pousser un bloc(3) ... (extrait de [14])

La figure 1 montre un comportement complexe final mémorisé par un approximateur de fonction unique grâce à l'apprentissage par transfert des différentes tâches sources [28].

Bien que ces méthodes permettent d'apprendre par des moyens progressifs, leur portée est limitée à la tâche spécifique à accomplir. Ces travaux ne couvrent pas l'apprentissage continu, ouvert et cumulatif d'une variété de tâches, un défi au cœur de la robotique développementale qui se caractérise par l'acquisition progressive de compétences et de connaissances [27].

Coordination de compétences Afin de faciliter l'apprentissage de compétences diverses, se basant potentiellement sur des compétences de base communes, il serait intéressant de pouvoir assimiler le Curriculum sous forme de modules de compétences indépendants. La principale difficulté réside dans le fait de devoir coordonner des modules dont les sorties sont concurrentes (par exemple : plusieurs compétences affectant le déplacement d'un agent).

Des travaux antérieurs dans le domaine des systèmes de commande de robots tels que l'architecture de subsomption [3] offrent des solutions pour alterner l'utilisation de différentes compétences selon le contexte. Cette méthode de composition séquentielle et exclusive de compétences est la plus répandue, depuis le "world of blocks" de Minsky [23] jusqu'aux systèmes de contrôles avancés des architectures cognitives (SOAR [19], CLAIRON, ICARUS [20]), en passant par des méthodes telles que le Skill Chaining [18] qui utilisent une séquence d'actions très simple dont la transition est déclenchée par l'apparition d'un contexte particulier.

Une autre possibilité de composition est la composition simultanée, ou somme vectorielle.

Les travaux sur les Boids [32] constituent un des premiers exemples de cette méthode de coordination entre des comportements concurrents. Les Boids sont des oiseaux virtuels conçus pour expérimenter le comportement de troupeau (*flocking*). Ils doivent ainsi satisfaire trois contraintes : la cohésion (rester proche du groupe), la séparation (garder une distance minimale avec les autres individus) et l'alignement (se diriger dans la même direction que le groupe). Trois compétences motrices indépendantes sont en charge de satisfaire chacune de ces contraintes, et la compétence globale devient, comme le dit Reynolds, un problème "d'arbitration de comportements indépendants".

"Chaque comportement dit : "si J'étais aux commandes, J'accélérerais dans cette direction." [...] C'est au module de navigation du cerveau du Boid de collecter toutes les demandes d'accélération pertinentes et de déterminer ensuite une seule accélération désirable du point de vue du comportement. Il doit combiner, prioriser et arbitrer entre des impulsions potentiellement conflictuelles." [32]

Ce mécanisme est employé dans des architectures comme AuRA [1] ou Sat-Alt [33], ou encore récemment sur des problèmes de locomotion complexe pour un robot humanoïde à 54 dimensions [16]. On retrouve également ce principe dans le "Social Force Model" décrivant le comportement de piétons [17].

Organisation hiérarchique Un autre aspect important de l'accumulation de compétences indépendantes sous forme de modules est l'établissement d'une hiérarchie de contrôle entre ces modules.

L'architecture ICARUS [6] est une architecture téléoréactive [24, 26], utilisant une organisation hiérarchique de deux manières différentes. L'évaluation de l'état actuel se fait en évaluant une hiérarchie de percepts/concepts de bas en haut, réunissant des percepts en concepts de bas niveau, puis ces concepts de bas niveaux en concepts de haut niveau. L'exécution se fait en parcourant une hiérarchie de buts de haut en bas, le but de plus haut niveau est activé par les concepts correspondants et cherche à réaliser ses sous buts. Les sous buts possèdent eux mêmes des sous buts qui sont choisis en fonction de l'état de la hiérarchie de percepts/concepts. Une des particularités d'ICARUS est d'évaluer la prochaine action non seulement en fonction de l'état de la hiérarchie de percepts/concepts, mais aussi de sa position courante dans la hiérarchie de buts/sous-but. Cette particularité donne à ICARUS une approche équilibrée entre un système purement réactif et un engagement fort dans les plans de haut niveau (*commitment*).

Une série de travaux qui réunissent une grande partie des éléments de la robotique développementale présentés jusqu'ici sont les travaux sur les techniques de Robot Shaping de Dorigo et Colombetti [9, 10]. Ici, les comportements s'apprennent par Curriculum et sont représentés comme des compétences indépendantes. Ces compétences sont ensuite combinées pour obtenir des comportements de plus haut niveau. L'accent est mis sur le processus d'apprentissage en relation avec la structure représentant la connaissance. Plusieurs architectures sont discutées, des hiérarchies monolithiques aux hiérarchies multi-niveaux. Des méthodes d'apprentissage et de récompense adaptées aux agents artificiels sont également proposées.

Afin de répondre aux besoins spécifiques de la robotique développementale nous avons proposé l'architecture MIND [35]. MIND (Modular Influence Network Design) est une architecture de contrôle pour agent artificiel conçue pour apprendre des comportements à partir d'un Curriculum établi par un instructeur humain. MIND encapsule les comportements dans des modules et les combine en une hiérarchie reflétant la nature modulaire et hiérarchique des tâches complexes, et permet la préservation et la réutilisation des compétences acquises. Les résultats présentés dans l'article [35], sur une tâche de collecte d'objets avec évitement d'obstacles, ont montré que l'architecture MIND a su transformer le Curriculum en une hiérarchie de skills avec à la fois coordination et exclusion mutuelle de skills. MIND a été en mesure de traiter les tâches sensori-motrices de bas niveau ainsi que les opérations complexes de coordination de skills, sans fournir aucune information a priori sur la nature de la tâche. Nous avons montré les avantages de cette conception modulaire, par exemple pour l'identification et le réentraînement spécifique d'un aspect du comportement sous-optimal. Enfin nous avons montré que cette modularité est essentielle dans le cadre d'un développement continu où les futures compétences à apprendre ne peuvent pas être anticipées, et bénéficieront d'une large bibliothèque de compétences de base disponibles pour être combinées.

Développement et interactions Un point sur lequel s'appuie la robotique développementale est l'idée d'incarnation, l'affirmation selon laquelle le fait d'avoir un corps qui sert de médiateur entre la perception et le comportement joue un rôle intégral dans l'émergence de la cognition humaine [22]. Cette incarnation soumet l'entité apprenante aux contraintes

physiques élémentaires de l'environnement (sa réalité) et pousse à l'élaboration de méthodes de résolution fondées sur une approche locale. La condition de cette entité est donc celle d'un agent [11].

Dans les premiers travaux se revendiquant de la robotique développementale, encore appelée robotique épigénétique, l'importance des interactions sociales est mise en avant comme faisant partie des interactions menant au développement cognitif [38]. L'étude des Systèmes Multi-Agents a montré les bénéfices d'une vision orientée agent et des interactions sociales dans un système. Les travaux sur la modélisation de sociétés d'insectes [7, 31] ont montré que la coordination d'agents réactifs très simples peut aboutir à des comportements complexes, capables de résoudre des problèmes bien au-delà de la simple somme des capacités individuelles de chaque agent. En complexifiant l'échange de simple signaux en un échange de messages il est possible de passer d'une simple coordination réactive vers une collaboration mettant en oeuvre des rôles différents au sein d'un groupe, la mise en oeuvre de tels systèmes est étudiée dans le cadre de la collaboration dans une flottille d'AUV [5].

L'intérêt de se pencher sur la question de l'interaction sociale pour un agent développemental va au-delà du gain de productivité en rapport à une tâche à accomplir. Pour qu'un agent en développement puisse sortir de la phase expérimentale et entrer dans le monde réel, il devra être capable d'interagir socialement, à la fois avec d'autres agents du même type et avec une population hétérogène d'agents, y compris biologiques.

L'interaction sociale est étroitement liée avec les représentations internes, dans une notion de soi par exemple, mais aussi dans la communication des intentions de résoudre les problèmes collectifs qui pourraient nécessiter l'acquisition de symboles de manière émergente afin de former un proto-langage [21, 37]. Les interactions sociales jouent également un rôle important dans le processus d'apprentissage, avec des algorithmes exploitant des aspects sociaux tels que l'imitation, ou l'interaction avec un instructeur humain par une forme de communication beaucoup plus naturelle que des algorithmes de programmation.

3 Modular Influence Network Design

MIND (Modular Influence Network Design) est une architecture hiérarchique de modules capables de coordonner des comportements (*Skills*) distincts pour accomplir une tâche com-

plexe. Cette architecture permet d'accumuler continuellement de nouvelles compétences et d'utiliser la coordination des compétences antérieures pour maîtriser rapidement de nouvelles tâches d'une complexité croissante. Dans cette section nous reprenons les principes de base de l'architecture MIND [35].

3.1 Base skill, complex skill, et influence

Considérons un agent dont les informations sensorielles et les commandes motrices sont représentées comme un vecteur de réels, normalisés entre 0 et 1. On peut créer un module qui encapsule une fonction $f(x)$ qui prend en entrée le vecteur $V_I = [I_1, I_2, \dots, I_k]$ et fournit en sortie le vecteur $V_O = [O_1, O_2, \dots, O_2, \dots, O_j]$. Cette fonction peut être implémentée comme une procédure de programmation, ou peut être un approximateur de fonction, un réseau neuronal, ou tout autre type de fonction qui associe deux vecteurs de nombres réels. Nous appellerons un tel module un *skill*, et le module dont le vecteur de sortie est utilisé directement comme commande moteur un *base skill*.

$$V_O = f(V_I) \quad (1)$$

Eq.1 : le vecteur d'entrée V_I est fourni à la fonction interne $f()$ du skill pour produire le vecteur de sortie V_O .

Il est possible de créer un unique base skill qui utilise toutes les entrées sensorielles et toutes les sorties motrices de l'agent pour apprendre à accomplir une tâche, même relativement complexe, chaque leçon du Curriculum étant mémorisée dans la même structure unique. Ce skill unique agrège toutes les différentes expériences dans sa propre fonction interne, sans qu'il soit possible de différencier ce qui a été enseigné.

Au lieu d'accomplir une tâche complexe au moyen d'un seul skill, nous accomplirons de nombreuses sous-tâches, potentiellement concurrentes, avec des base skills distincts. Chaque base skill n'associera que les entrées et sorties nécessaires à l'exécution de la tâche qui lui est associée. Pour accomplir la tâche complexe, nous allons ensuite créer un *complex skill* qui va coordonner plusieurs base skills, que nous appelons ses *subskills* (Fig.2). Un *complex skill* coordonne ses *subskills* en envoyant un signal appelé *influence* qui détermine le poids qu'un subskill aura sur l'action résultante, ce qui revient à déléguer à une ou plusieurs sous-capacités la résolution de la tâche courante.

Un Complex skill, comme n'importe quel skill, encapsule une fonction qui prend en entrée un vecteur de valeurs issues des capteurs V_I et

sort un vecteur de nombres réels V_O , dont la sortie est dirigée vers ses subskills. Ce vecteur de sortie est appelé le vecteur d'influence $V_{Infl} = [Infl_1, Infl_2, \dots, Infl_i]$, et ses éléments $Infl_x$ sont appelés *influences*.

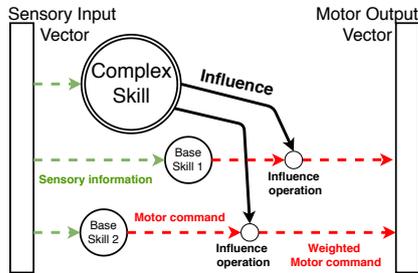


Fig. 2 – Un *complex skill* influençant deux *base skills*

Un *complex skill* peut avoir d'autres *complex skills* comme subskills, créant des hiérarchies de skills. Au sommet de la hiérarchie se trouve le *Master skill* dont la seule particularité est de recevoir une influence constante de 1.0, une impulsion permettant de lancer le mécanisme de calcul.

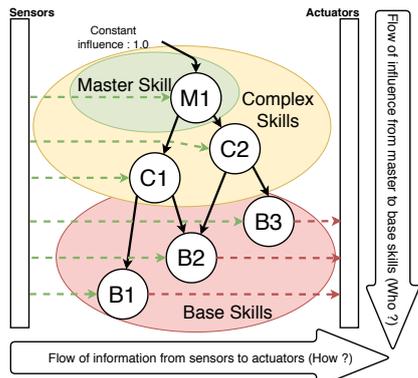


Fig. 3 – Une hiérarchie de skills, le *master skill* influence des *complex skills* qui à leur tour influencent des *base skills*

Cette hiérarchie de skills forme un graphe orienté acyclique (Fig.3). L'influence s'exerce le long d'un axe vertical, du *master skill* jusqu'aux *base skills*, et détermine qui est responsable de l'action résultante. L'information des capteurs et les commandes des moteurs sont envoyées par les *base skills* vers les actionneurs formant un flux d'information horizontal. Ce flux détermine comment l'action résultante va être exécutée.

3.2 Utilisation de l'influence pour déterminer les commandes moteurs

A partir du *master skill*, chaque *complex skill* calcule son vecteur de sortie V_O et multiplie chaque élément par la somme des influences reçues, formant le vecteur d'influence V_{Infl} . Le skill envoie alors chaque élément $Infl_x$ de V_{Infl} au subskill correspondant (Fig.4).

$$V_{Infl} = V_O * \sum_{x=1}^i Infl_x \quad (2)$$

Eq.2 : V_{Infl} le vecteur d'influence vers les subskills, $V_O = f(V_I)$ le vecteur de sortie de la fonction interne du skill, et $\sum_{x=1}^i Infl_x$ la somme de toutes les influences reçues par le skill (également noté $\Sigma Infl$).

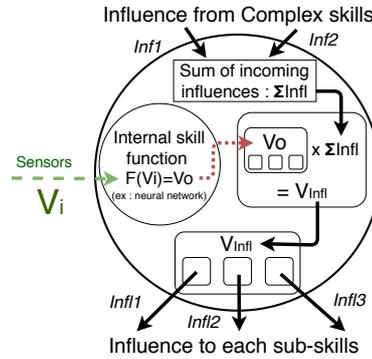


Fig. 4 – Architecture interne d'un skill

Un *base skill*, comme un *complex skill*, calcule son vecteur de sortie et multiplie chaque élément par la somme des influences qu'il a reçues (Eq.2), formant le vecteur de commandes moteur $V_{Com} = [Com_1, Com_2, \dots, Com_j]$. Le *base skill* envoie alors chaque élément Com_x du vecteur de commandes moteur au module moteur correspondant, ainsi que la somme des influences reçues par lui-même $\Sigma Infl$.

Chaque module moteur calcule ensuite la commande correspondant à son actionneur à partir des commandes reçues par les skills concurrents au moyen d'une somme pondérée normalisée :

$$M = \frac{\sum_{x=1}^l Com_x}{\sum_{x=1}^i \Sigma Infl_x} \quad (3)$$

Eq.3 : M la commande de moteur résultante, x l'index d'un des l *base skills* qui envoie une commande de moteur, Com_x la commande de moteur pondérée pour ce module moteur depuis le *base skill* x , $\Sigma Infl_x$ la somme des influences du *base skill* x .

4 Éléments de l'expérimentation

L'environnement, les éléments de l'expérimentation et les techniques d'apprentissage utilisés sont les mêmes que ceux présentés dans l'article précédent [35]. Nous rappelons ici brièvement leurs caractéristiques essentielles et présentons les quelques fonctionnalités ajoutées.

Le robot utilisé dans notre simulation est composé de deux moteurs, un pour chaque roue, et d'une pince pour saisir l'objet cible. Il dispose également de 18 capteurs, détecteurs d'obstacles et d'orientation des objets à collecter.

Il peut également générer une onde sinusoïdale qui peut être utilisée à la manière d'un capteur (par exemple, pour résoudre des situations de deadlock réactif).

Le robot est également équipé d'un émetteur de signal et d'un récepteur correspondant capable d'indiquer la présence, l'orientation et la distance d'un signal émis par un autre agent. Enfin le robot est capable de connaître la distance et l'orientation du robot le plus proche.

Les fonctions internes des skills sont principalement des perceptrons multicouche, dont la couche d'entrée correspond au vecteur d'entrée du skill et la couche de sortie à son vecteur de sortie. Selon le skill le nombre de couches cachées varie entre 2 et 3. Les fonctions de transfert des couches du réseau neuronal peuvent être sigmoïde ou ReLU. La couche de sortie utilise une fonction de transfert linéaire fixée entre 0 et 1 (clamp). Certains skills peuvent utiliser une topologie différente, par exemple un réseau neural convolutionnel pour traiter un grand nombre d'entrées, ce qui réduit le temps de convergence de l'algorithme d'apprentissage.

Enfin, certains skills triviaux utilisent une procédure programmée comme fonction interne, la complexité de la tâche ne justifiant pas un apprentissage.

L'algorithme d'apprentissage est un algorithme génétique simple. Les poids des connexions du réseau neuronal sont ordonnés dans un vecteur, qui correspond au génome de l'individu du point de vue de l'algorithme génétique. Le génome sera croisé et muté en fonction du résultat de la fonction de fitness (d'adaptation) que l'environnement fournit en retour. L'algorithme génétique se prête bien à ce genre de tâches, l'évaluation ayant lieu à la fin du cycle de vie de l'individu simplifie les problèmes de récompense différée et ne nécessite pas de set d'entraînement comme dans le cas d'un algorithme de rétropropagation. L'inconvénient de l'algorithme génétique est son coût élevé en ressources dans l'évaluation de chaque génome. Toutefois, l'évaluation est totalement indépendante et parallélisable et permet donc l'utilisation de solutions de calcul haute performance.

Notre implantation de l'algorithme génétique permet d'initialiser la population soit de manière aléatoire, soit en fournissant des individus déjà entraînés. Ceci permet de reprendre l'entraînement à partir d'un point donné pour optimiser un comportement, et compléter la technique d'optimisation dans le contexte final présenté dans l'article précédent (Scenario 2 [35]).

5 Fourragement multi-agents

Le problème des robots fourrageurs est un problème bien connu dans le domaine de l'intelligence artificielle et des systèmes multi-agents, que ce soit du point de vue de la modélisation de sociétés d'insectes [8] ou du développement de robots autonomes pour l'exploration spatiale [4]. Ce problème met en jeu des questions de résolution distribuée de problèmes et de communication inter-agents. Nous souhaitons apprendre une tâche de fourragement multi-agents en nous basant sur une hiérarchie MIND préexistante qui réalise la collecte d'un objet dans un environnement contenant des obstacles [35]. Nous souhaitons montrer qu'il est possible d'étendre les capacités d'un agent MIND, dans le sens développemental du terme, à de nouvelles tâches nécessitant un comportement réactif de coordination multi-agents. Ce comportement de fourragement multi-agents proche d'un comportement d'insecte reste dans un domaine purement réactif et ne requiert que l'échange de signaux simples.

5.1 Protocole

Un groupe de quatre agents est placé dans l'environnement. Chaque agent possède un capteur donnant l'orientation de l'objet, avec une portée de perception limitée, un émetteur de signaux et un récepteur donnant l'orientation du signal le plus proche. Les agents ont également un capteur qui donne l'orientation de l'agent le plus proche.

L'environnement contient des obstacles à éviter et une zone de dépôt pour les objets à collecter. Les objets sont placés proches les uns des autres dans l'environnement, en groupes de 8.

Les agents ne peuvent transporter qu'un seul objet à la fois et leur émetteur de signaux a une portée bien supérieure à celle de leur capteur d'orientation d'objet.

Étant donné que les agents ne possèdent aucune information sur leur environnement, le comportement optimal attendu consiste à chercher les objets à collecter là où aucun autre agent n'est présent afin d'obtenir la plus grande zone observée. Quand un agent trouve le groupe d'objets, il envoie un signal pour demander aux autres agents de venir.

Tous les agents de la simulation utilisent la même hiérarchie MIND, chacun possédant une instance traitant les valeurs de ses propres capteurs. La réponse globale durant l'évaluation du groupe d'agents utilisant le même génome pour la fonction interne à entraîner constitue le score de ce génome.

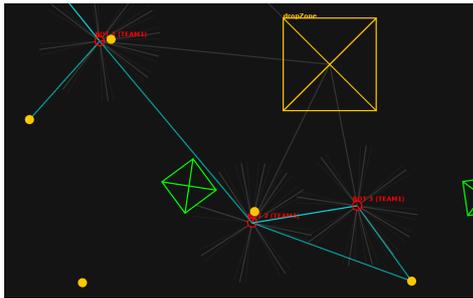


Fig. 5 – l’environnement de simulation : en jaune la zone de dépôt et les objets à collecter, en vert les obstacles et en rouge les agents. Les différentes lignes représentent les informations des capteurs et des signaux.

5.2 Hiérarchie

La hiérarchie finale que nous avons établie est montrée dans la figure 6. Les flèches vertes représentent les entrées, les rouges les sorties et les noires les relations d’influence. Les skills en pointillé utilisent une fonction interne programmée et les skills en bleu sont les skills qui nécessitent un apprentissage collectif.

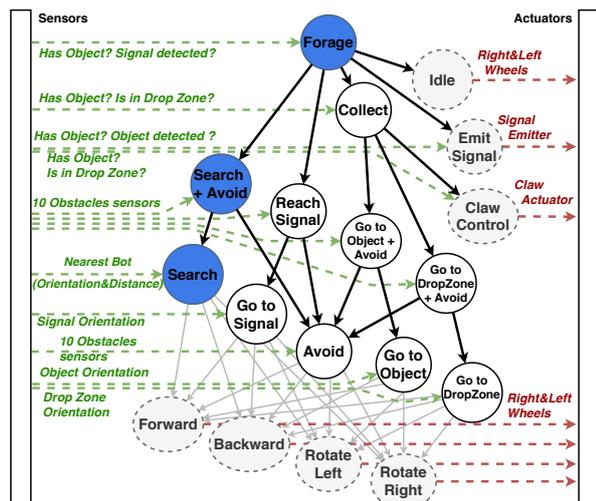


Fig. 6 – L’architecture MIND pour la tâche de fourrage

Hiérarchie initiale Pour accomplir la tâche de fourrage, nous nous basons sur la hiérarchie *Collect* préexistante [35]. Cette sous hiérarchie a été apprise individuellement, la tâche de collecte étant accomplie par un seul agent.

Collect se base sur 2 complex skills *GoToObject+Avoid* et *GoToDropZone+Avoid* qui, respectivement, cherche un objet en évitant les obstacles et retourne à la zone de dépôt en évitant les obstacles. Ces deux skills sont eux même basés sur 3 skills : *GoToObject*, *GoToDropZone* et *Avoid* qui, respectivement, dirige l’agent vers un objet, dirige l’agent vers la zone de dépôt, et permet à l’agent d’éviter les obstacles.

Un autre skill présent est *Reach Signal* qui combine *Avoid* et *GoToSignal* et permet à l’agent d’atteindre un signal émis par un autre agent en évitant les obstacles. Bien que ce skill mette en oeuvre les signaux, il n’a pas été appris dans un contexte multi agent, l’environnement d’apprentissage utilisé est capable de simuler l’émission d’un signal.

Skills programmés Certains skills ne sont pas appris mais programmés, leur fonction étant triviale et ne justifiant pas la mise en place d’un curriculum. C’est le cas de *Emit Signal* qui émet un signal dès qu’un objet est présent et *Claw Control* qui saisit l’objet quand il est à portée et le lâche quand l’agent est présent dans la zone de dépôt. Certain skills sont aveugles, c’est à dire qu’ils ne prennent aucune entrée capteur et fournissent une sortie constante : *Forward*, *Backward*, *Rotate Left*, *Rotate Right* et *Idle*.

Comportement multi-agents *Search+Avoid* et son subskill *Search* sont les skills réellement dépendants d’un fonctionnement multi-agents, et ne peuvent être appris que dans un contexte multi-agents. Un agent seul ne dispose d’aucune information qui puisse l’aider à établir un parcours de recherche efficace, d’aucune mémoire des lieux déjà visités ou d’un système de phéromones/marqueurs à déposer dans l’environnement. Il ne connaît pas la forme ou les dimensions de l’environnement et ne peut pas établir, par exemple, un parcours de balayage. Toutefois, en tant que groupe, les agents peuvent tirer parti du fait qu’ils ont plusieurs fois le champ de perception d’un agent individuel et utiliser les informations sur la position de chacun pour se coordonner en une formation qui couvrira la plus grande zone possible.

Master skill Enfin *Forage* coordonne *Search+Avoid* pour trouver les objets, *Emit Signal* pour appeler les autres agents, *Reach Signal* pour rejoindre un agent qui aurait trouvé les objets avant lui, et *Collect* pour collecter les objets trouvés. Comme l’agent ne possède pas de mémoire et de système de coordonnées pour décrire la position du groupe d’objet, *Forage* a la possibilité d’utiliser le skill *Idle* qui immobilise l’agent et permet ainsi d’attendre que d’autres agents arrivent. L’agent agit ainsi comme une balise marquant la position [15].

5.3 Résultats

Cette section décrit le comportement d’un agent sous la forme d’une succession d’étapes.

Les figures 7 et 8 montrent l’état de la hiérarchie MIND pour l’agent aux étapes 3 et 5. En rouge les actionneurs (placés proches des skills correspondants) et en bleu les 4 capteurs

fournis au skill **Forage** : *NearFriendlyActive* présence d'un autre agent à portée de détection, *ListenerAAActive* présence d'un signal à portée de détection, *ACTOBJA* présence d'un objet cible est à portée de détection, *SENSOBJ* indique le transport d'un objet. La hiérarchie correspond à celle présentée dans la figure 3, les skills sont en gris, en dessous du nom de chaque skill apparaît la somme de l'influence reçue par le skill. Enfin, les liens d'influence sont représentés par un dégradé allant de noir pour la valeur 0 au vert pour la valeur 1, le trait central représente la commande du skill seul et le trait extérieur représente l'influence effectivement transmise (voir Fig.4 et Eq.2).

Étape 1 : L'agent ne perçoit pas d'objets, ne reçoit pas de signal. Le master skill *Forage* active *Search+Avoid* avec une intensité maximale (une très faible valeur relative et donnée à *ReachSignal* mais ceci n'a aucun impact sur le comportement).

Étape 2 : L'agent perçoit un objet, le signal est activé, le skill *Collect* est activé en majorité, *Search+Avoid* reste partiellement actif mais ne perturbe pas le comportement de *Collect*. Dans cette phase *Collect* active la branche *GoToObject* qui permet d'atteindre l'objet.

Étape 3 : L'agent a ramassé l'objet, le capteur correspondant est actif. Dans cette phase *Collect* active la branche *GoToDropZone* qui permet de rejoindre la zone de dépôt. *Forage* active complètement *Collect* et désactive *Search+Avoid*. À cette étape les autres agents sont attirés par le signal émis.

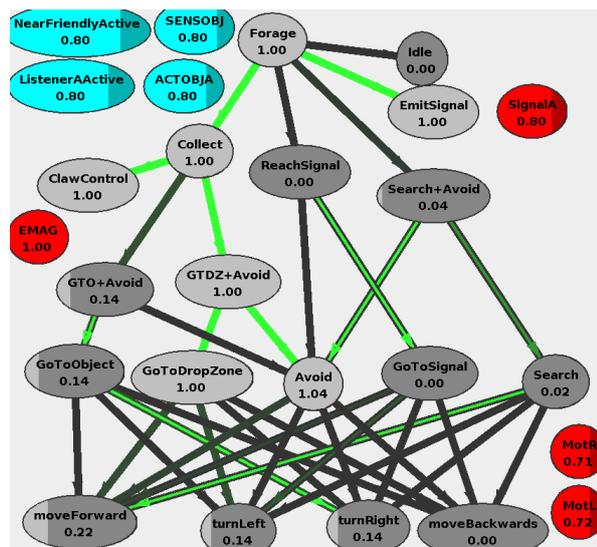


Fig. 7 – Étape 3

Étape 4 : L'agent en route pour la zone de dépôt, quitte le champ de perception des objets, son

signal s'éteint. Le changement des valeurs des capteurs influe sur la hiérarchie sans pour autant perturber le comportement de l'agent. Bien que notre agent ait cessé d'émettre le signal, un autre agent est maintenant à portée de perception des objets et émet le signal pour attirer les autres agents ne transportant pas encore d'objets.

Étape 5 : L'agent a déposé l'objet. *Collect* est désactivé, *Search+Avoid* et *ReachSignal* sont activés. Même si *ReachSignal* est inférieur à *Search+Avoid*, on peut voir (fig.8) que l'influence transmise résulte dans l'activation majoritaire du comportement requis par *ReachSignal* (i.e : *turnRight* : 0.33 contre *turnLeft* : 0.14).

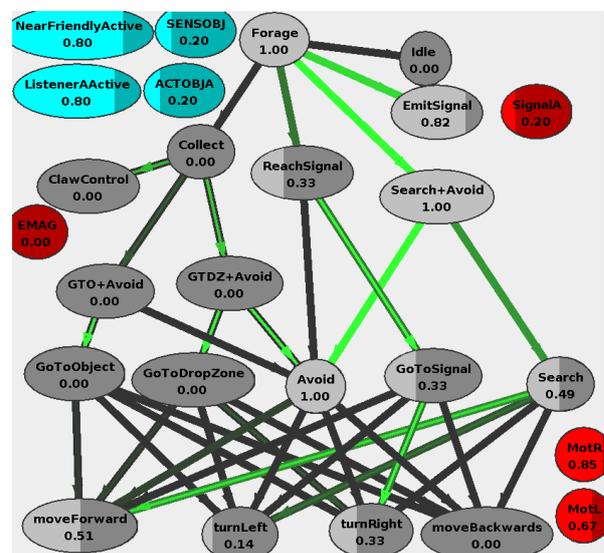


Fig. 8 – Étape 5

Étape 6 et 7 : Le virage en direction de la source du signal est effectué, l'agent se déplace en ligne droite vers celui-ci. Une fois que l'agent est à nouveau à portée de perception d'un objet cible, le processus de collecte est répété.

L'élaboration d'une hiérarchie MIND pour un fonctionnement multi-agents en coordination ne présente ni contraintes ni difficultés particulières, autres que de respecter le point de vue et la démarche agent. La coordination se fait localement, mettant en oeuvre les solutions les plus simples en se coordonnant avec les agents les plus proches. En outre, l'algorithme génétique est extrêmement flexible vis à vis de la composition du skill fourni. Par exemple, au cours d'une expérience différente sur la tâche *Collect*, une erreur dans les paramètres a rendu l'agent incapable de saisir l'objet. Exploitant le moteur physique de la simulation, l'agent s'est contenté de pousser l'objet dans la zone de dépôt. On peut voir dans notre cas le comportement de

l'agent à l'étape 5 ou encore le fait que le skill *Idle* permettant à l'agent d'attendre que les autres agents arrivent n'a pas été utilisé.

La grande difficulté, comme l'ont remarqué Dorigo et Colombetti [9][10], réside dans l'élaboration de la fonction de fitness. Régler l'équilibre entre plusieurs sources de récompenses, la valeur à attribuer à chacune est un processus qui nécessite souvent plusieurs essais pour observer (et interpréter) l'impact de chaque réglage. Bien que dans notre cas la fonction de fitness elle-même ne soit qu'une simple récompense pour chaque objet collecté, ce travail s'est trouvé reporté sur l'équilibrage des paramètres de l'environnement et la compréhension de leurs interactions avec la fonction de fitness. Pour reprendre l'exemple du skill *Idle*, après avoir observé le comportement des agents, nous nous sommes aperçus que l'agent qui trouve le groupe d'objets en premier a rarement le temps de quitter la zone de perception avant que les autres agents arrivent, et donc il n'est pas nécessaire de les attendre (c'est même un gain de temps et donc une chance de collecter plus d'objets dans le temps imparti). Ici la fonction de fitness n'est pas directement en cause mais simplement la taille de l'environnement ne donne pas l'occasion aux agents d'être assez éloignés pour avoir besoin du skill *Idle*. D'autre part, la taille de l'environnement, la dispersion du groupe d'objets et même la densité des obstacles dans l'environnement impactent l'apprentissage et l'utilisation d'une recherche d'objet coordonnée au lieu d'un simple parcours aléatoire de l'environnement.

6 Conclusion

Nous avons vu que l'architecture MIND est capable d'accomplir une tâche de coordination pour un groupe homogène d'agents réactifs. Il a été possible de combiner des comportements acquis de manière individuelle avec de nouveaux comportements strictement collectifs, et ce sans changement majeur dans les techniques d'apprentissage employées. Le seul impact sur le coût en ressources de calcul vient de la gestion des signaux dans l'environnement simulé. L'aspect modulaire de MIND a encore un fois montré son avantage puisqu'il a été possible d'ajouter non seulement de nouveaux comportements, mais aussi de nouveaux capteurs et actionneurs sans aucune modification à la hiérarchie d'origine. Pris individuellement, le comportement *Collect* est toujours capable de fonctionner de manière autonome.

Comme nous avons traité de l'apprentissage d'un groupe homogène d'agents se coordonnant pour atteindre un but commun, tous les agents

du groupe utilisent le même génome durant l'évaluation d'une population. L'intérêt réside dans le fait que d'une part, à chaque instant le même génome est évalué autant de fois que le nombre d'agents présents, et d'autre part le même génome est évalué de part et d'autre d'une interaction. Ce dernier aspect simplifie l'évaluation car il ne s'agit plus de savoir si chaque agent a bien effectué son rôle dans l'interaction, ou même quels sont les rôles et la forme de cette interaction, mais simplement de savoir si un résultat globalement positif a eu lieu. Dans le cas d'une évaluation d'une interaction entre des génomes différents, un cas d'échec amène soit à punir les deux, soit devoir déterminer lequel des deux a causé l'échec. Il est possible d'imaginer une tâche compétitive ou une évaluation concurrente est au contraire plus simple à établir, par exemple un combat avec élimination.

Cette question de l'hétérogénéité, plus particulièrement dans les capacités et les motivations des agents est un point important pour notre projet d'agent développemental. Serait-il possible d'obtenir une coordination dans un groupe hétérogène d'agents réactifs? Dans le cadre, par exemple, d'une évaluation concurrente de génomes de différentes populations dans un contexte proie-prédateur. Ou encore conduire à l'émergence d'une relation symbiotique dans un contexte de coopération où l'hétérogénéité induit un biais initial vers le développement d'un rôle spécifique.

Cette première approche de coordination multi agent s'est basée sur des agents uniquement réactifs. Nos récents travaux sur l'architecture MIND nous ont permis d'intégrer un système de mémoire simple, les variables, que nous utilisons pour stocker des représentations internes. Ces représentations sont apprises de manière émergente, en utilisant le même système d'apprentissage par curriculum que pour les skills. En utilisant ce système nous sommes parvenus à apprendre à notre agent à compter et à retenir le nombre d'étapes accomplies dans une séquence d'actions afin de choisir le bon comportement à exécuter.

En utilisant ce système de mémoire nous envisageons d'apprendre un comportement multi agent de coopération, au moyen d'une variable représentant le rôle de l'agent, comme une première approche d'un problème d'organisation AGR [12]. La relation entre la variable représentant le rôle et les skills de la hiérarchie MIND serait apprise au moyen d'un curriculum. La valeur de cette variable évoluerait ensuite au cours d'une

simulation pour obtenir l'émergence d'une organisation par rôle au sein d'un groupe d'agents homogènes, d'une manière similaire aux travaux des simulations MetaCiv [13, 34], où une spécialisation sociale est apprise par renforcement.

Références

- [1] R. C. Arkin and T. Balch. Aura : Principles and practice in review. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3) :175–189, 1997.
- [2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48. ACM, 2009.
- [3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1) :14–23, 1986.
- [4] R. A. Brooks, P. Maes, M. J. Mataric, and G. More. Lunar base construction robots. In *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, pages 389–392. IEEE, 1990.
- [5] N. Carlési. *Coopération entre véhicules sous-marins autonomes : une approche organisationnelle réactive multi-agent*. Theses, Université de Montpellier 2, 2013.
- [6] D. Choi and P. Langley. Evolution of the icarus cognitive architecture. *Cognitive Systems Research*, 48 :25–38, 2018.
- [7] J.-L. Deneubourg and S. Goss. Collective patterns and decision-making. *Ethology Ecology & Evolution*, 1(4) :295–311, 1989.
- [8] J.-L. Deneubourg, S. Goss, and al. The dynamics of collective sorting robot-like ants and ant-like robots. In *1st international conference on simulation of adaptive behavior on From animals to animats*, pages 356–363, 1991.
- [9] M. Dorigo and M. Colombetti. Robot shaping : Developing autonomous agents through learning. *Artificial intelligence*, 71(2) :321–370, 1994.
- [10] M. Dorigo and M. Colombetti. *Robot shaping : an experiment in behavior engineering*. MIT press, 1998.
- [11] J. Ferber. Les systèmes multi-agents : vers une intelligence collective. *InterEditions, Paris*, 322, 1995.
- [12] J. Ferber, O. Gutknecht, F. Michel, et al. Agent/group/roles : Simulating with organizations. In *ABS'03 : Agent Based Simulation*, 2003.
- [13] J. Ferber, J. Nigon, G. Maille, T. Seguin, S. Holtz, and T. Stratulat. De masq à metaciv : un cadre générique pour modéliser des sociétés humaines dans une approche transdisciplinaire, 2014.
- [14] F. Foglino, C. C. Christakou, and M. Leonetti. An optimization framework for task sequencing in curriculum learning. In *Joint IEEE 9th International Conference ICDL-EpiRob*, pages 207–214. IEEE, 2019.
- [15] S. Goss and J.-L. Deneubourg. Harvesting by a group of robots. In *Proceedings of the First European Conference on Artificial Life*, pages 195–204, 1992.
- [16] N. Heess, G. Wayne, Y. Tassa, T. P. Lillicrap, M. A. Riedmiller, and D. Silver. Learning and transfer of modulated locomotor controllers. *CoRR*, abs/1610.05182, 2016.
- [17] D. Helbing and A. Johansson. Pedestrian, crowd, and evacuation dynamics. *arXiv preprint arXiv :1309.1609*, 2013.
- [18] G. Konidaris and A. G. Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in neural information processing systems*, pages 1015–1023, 2009.
- [19] J. E. Laird. Extending the soar cognitive architecture. *Frontiers in Artificial Intelligence and Applications*, 171 :224, 2008.
- [20] P. Langley and D. Choi. A unified cognitive architecture for physical agents. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1469. Menlo Park, CA ; Cambridge, MA ; London ; AAAI Press ; MIT Press ; 1999, 2006.
- [21] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini. Developmental robotics : a survey. *Connection science*, 15(4) :151–190, 2003.
- [22] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini. Developmental robotics : a survey. *Connection Science*, 15(4) :151–190, 2003.
- [23] M. Minsky. *Society of mind*. Simon and Schuster, 1988.
- [24] J. L. Morales, P. Sánchez, and D. Alonso. A systematic literature review of the teleo-reactive paradigm. *Artificial Intelligence Review*, 42(4) :945–964, 2014.
- [25] S. Narvekar, J. Sinapov, M. Leonetti, and P. Stone. Source task creation for curriculum learning. In *International Conference on Autonomous Agents & Multiagent Systems*, pages 566–574. International Foundation for Autonomous Agents and Multiagent Systems, 2016.
- [26] N. Nilsson. Teleo-reactive programs for agent control. *Journal of artificial intelligence research*, 1 :139–158, 1993.
- [27] P.-Y. Oudeyer. Developmental robotics. In *Encyclopedia of the Sciences of Learning*, pages 969–972. Springer, 2012.
- [28] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10) :1345–1359, 2010.
- [29] J. Piaget. *The construction of reality in the child*. Basic Books, New York, 1954.
- [30] J. Piaget and E. Duckworth. Genetic epistemology. *American Behavioral Scientist*, 13(3), 1970.
- [31] M. Resnick. Learning about life. *Artificial life*, 1(1_2) :229–241, 1993.
- [32] C. W. Reynolds. Flocks, herds and schools : A distributed behavioral model. In *ACM SIGGRAPH computer graphics*, volume 21, pages 25–34. ACM, 1987.
- [33] O. Simonin and J. Ferber. Modeling self satisfaction and altruism to handle action selection and reactive cooperation. In *Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior*, volume 2, pages 314–323, 2000.
- [34] F. Suro, J. Ferber, and T. Stratulat. Coglogo : une implémentation de metaciv pour netlogo. In *JFSMA*, pages 166–167. PFIA, 2019.
- [35] F. Suro, J. Ferber, T. Stratulat, and F. Michel. Une représentation hiérarchique de comportements agents pour l'apprentissage progressif et continu. In *JFSMA*, pages 67–76. PFIA, 2019.
- [36] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains : A survey. *Journal of Machine Learning Research*, 10(Jul), 2009.
- [37] P. Varshavskaya. Behavior-based early language development on a humanoid robot. Technical report, MIT AI Lab, 2002.
- [38] J. Zlatev and C. Balkenius. Introduction : Why epigenetic robotics? 2001.

Intégration de la Gestion du Contexte et des Dispositifs IoT dans des Maisons Intelligentes basées sur des Systèmes Multi-Agents

Iago Felipe Trentin^{1,2}, Fano Ramparany¹, Olivier Boissier²

¹ Orange Labs, Meylan, FR 38240

² Univ Lyon, MINES Saint-Étienne, CNRS Lab Hubert Curien UMR 5516, FR 42023

iagofelipe.trentin@orange.com / iago.felipe.trentin@gmail.com

Résumé

Avec les technologies IoT, la domotique s'oriente vers la maison intelligente. Les utilisateurs ont ainsi accès à des services à valeur ajoutée offrant un accès intégré aux dispositifs déployés au sein de la maison.

Outre le fait que ces applications sont centrées sur l'utilisateur, la coordination décentralisée des décisions et la flexibilité sont des défis importants. L'utilisation d'une approche multi-agent permet de relever ces défis. Cependant, l'intégration d'une telle approche avec les solutions de l'IoT et de la gestion de contexte n'est pas un problème facile. Nous présentons une approche pour cette intégration dans un système multi-agent afin que les agents puissent adapter les services centrés sur l'utilisateur à l'évolution du contexte et des dispositifs déployés.

Nous présentons la manière dont nous avons intégré ces technologies dans une architecture multi-agent qui a été conçue pour développer des services à valeur ajoutée qui adaptent leur fonctionnement à l'environnement.

Mots-clés

Sensibilité au contexte, IoT, système multi-agent, intégration.

Abstract

With the raise of IoT technologies, home automation is shifting to smart home. Users are empowered with added value services offering integrated access to home appliances.

Besides the user centrality of such applications, decentralized decision coordination and flexibility are important challenges. Using a multi-agent approach helps addressing these challenges. However, integrating such approach with IoT and Context Management solutions is not an easy problem that may hinder flexibility and adaptation for instance. In this paper, we present an approach for integrating such technologies in a multi-agent system so that agents can adapt user-centric services to the evolution of the context and devices deployed in the home environment. We present how we integrated such technologies in a multi-agent architecture engineered to develop added value services that adapt their functioning to the environment.

Keywords

Context-aware, IoT, Multi-Agent System, Integration.

1 Introduction

La satisfaction et le confort de ses habitants sont au cœur des exigences d'une maison intelligente. Par exemple, si un utilisateur reçoit un appel alors qu'il regarde un film, il s'attend à ce que le service s'adapte de telle sorte que le téléphone se déclenche et permette à l'utilisateur de répondre à l'appel. En parallèle, le service "regarder un film" exécuté par la télévision doit s'adapter, faire une pause ou activer les sous-titres, afin de gérer le conflit soulevé par l'action entreprise par le téléphone tout en démarrant l'exécution de son service pour traiter l'appel.

Comme on peut le voir, les services doivent s'adapter en temps utile aux besoins des utilisateurs ou à l'évolution du contexte. Plus généralement, une maison intelligente doit être sensible au *contexte*, c'est-à-dire l'ensemble des informations obtenues par les dispositifs intelligents existants, et *adapter* ses services en *cours d'exécution* afin, par exemple, de permettre à l'utilisateur de répondre à l'appel pendant qu'il regarde un film. Idéalement, la maison intelligente devrait réduire les interventions directes des humains, en étant *autonome* dans la poursuite d'objectifs *centrés sur l'utilisateur* pour satisfaire les besoins de ceux-ci. En outre, la télévision et le téléphone doivent interagir et se *coordonner* afin de réaliser les adaptations décrites. Pour l'instant, les solutions de maison intelligente conçues ont des difficultés à satisfaire ces exigences.

Composée de services distribués, avec de multiples objets coordonnés les uns avec les autres, une maison intelligente conçue selon une approche multi-agent est intéressante car elle structure l'application en termes d'agents autonomes coordonnés les uns avec les autres pour mettre en œuvre les services ciblés centrés sur l'utilisateur. Toutefois, pour se connecter et s'adapter à l'évolution de l'environnement, ces applications doivent interagir avec un écosystème de dispositifs intelligents. Elles doivent être capables également de tirer profit des données structurées et de la représentation constituant le contexte domestique auquel tous les services doivent s'adapter.

Le défi que nous relevons est l'intégration des dispositifs de l'internet des objets (IoT, de l'anglais "Internet of Things") et des outils de gestion du contexte dans une architecture de maison intelligente basée sur un système multi-agent afin

que les agents puissent adapter les services à l'évolution des besoins et de l'environnement des utilisateurs.

Le document est organisé comme suit : nous motivons d'abord notre travail en présentant dans la section 2 les principales exigences des applications d'une maison intelligente basées sur un système multi-agent en lien avec l'intégration du contexte et de dispositifs IoT. La section 3 montre l'architecture multi-agent que nous proposons pour répondre à ces exigences. Les sections 4 et 5 décrivent ensuite respectivement comment le contexte et les dispositifs IoT ont été intégrés dans cette architecture. La section 6 présente les leçons tirées du processus d'ingénierie. Enfin, la section 7 compare notre approche aux travaux existants.

2 Exigences

Plusieurs solutions de maisons intelligentes existent actuellement [1, 2, 3]. Leur analyse soulève un ensemble d'exigences qui doivent être prises en compte lors du développement de ces applications.

Comme pour les industries manufacturières [4, 5], les maisons intelligentes automatisent leurs processus et sont de plus en plus soutenues par des solutions IoT à cet effet. Ceci nécessite souvent une planification adaptative en fonction du contexte au moment de l'exécution. Toutefois, ces processus diffèrent de ceux de l'industrie car ils ne sont pas prévisibles par conception. Les maisons intelligentes n'effectuent généralement pas d'actions répétitives et contrôlées, et doivent également tenir compte des utilisateurs humains. Par conséquent, en plus d'intégrer des appareils hétérogènes et de composer leurs services pour répondre aux buts des utilisateurs, les maisons intelligentes doivent être sensibles au contexte, ce qui signifie qu'elles doivent adapter ces services composites aux évolutions de la situation pour satisfaire au mieux les utilisateurs. Le présent document se concentre sur ces deux défis en se focalisant sur les problématiques d'intégration de services hétérogènes et de la gestion du contexte. Résoudre ces problématiques est essentiel pour mettre en place dans le futur des stratégies évoluées de coopération et de négociation multi-agents au sein de ces environnements.

2.1 Intégration et Composition de Services Hétérogènes

Les maisons intelligentes actuelles sont de plus en plus complexes, équipées d'appareils omniprésents supportés par des dispositifs hétérogènes fournis par différents vendeurs [6]. Outre l'augmentation du nombre d'appareils, les messages, les formats de données, les protocoles de communication et les architectures continuent d'évoluer. Face à une telle hétérogénéité, plusieurs efforts de normalisation portent sur les niveaux syntaxique et sémantique, tels que ETSI SmartM2M SAREF [7], et W3C Web of Things (WoT) [8].

Certaines solutions permettent la découverte de dispositifs et de leurs services associés, en les connectant au moyen de l'infrastructure Web existante [9]. Toutefois, le problème d'intégration va au-delà de la découverte et de la

connexion. Il concerne également leur composition en services à valeur ajoutée, en s'appuyant sur ceux qui sont disponibles [10, 11].

Les solutions existantes pour intégrer différents dispositifs IoT, sont principalement basées sur des compositions ("mashups") IoT manuelles¹[12, 9], conçues pour permettre la synergie entre des services, protocoles et formats hétérogènes. Même si les mashups ad-hoc présentent un intérêt, ils sont souvent temporaires, spécifiques et non évolutifs [12]. Ces solutions ne présentent donc pas de raisonnement ni de comportement adaptatif, et exécutent plutôt des tâches préalablement programmées par des utilisateurs humains.

Ainsi, les maisons intelligentes ont besoin de mettre en place différents services s'appuyant sur des objets connectés et intégrés au sein de la maison d'une part, et, répondant, d'autre part, aux buts délégués par les utilisateurs. Ces derniers sont ainsi libérés de porter une attention constante au système.

2.2 Intégration du Contexte Dynamique de la Maison

Afin d'adapter leur exécution à l'environnement, y compris aux humains, les applications de la maison intelligente doivent avoir accès à des informations contextuelles actualisées. Le contexte disponible pour les services dans une maison intelligente est notamment alimenté par l'ensemble des informations obtenues et stockées par les dispositifs intelligents existants. Les applications de la maison intelligente peuvent avoir besoin de différentes sources d'information pour exécuter correctement leurs actions. Par exemple la télévision peut avoir besoin d'accéder aux données du capteur du canapé pour savoir si un utilisateur est prêt à regarder la télévision. Cela illustre le besoin d'interopérabilité et de coopération entre les dispositifs et services. La gestion du contexte peut ainsi devenir lourde et consommer des ressources de chacun des dispositifs, surtout lorsqu'il faut partager ce contexte entre plusieurs appareils. Pour faire face à la complexité et au coût de cette gestion, de nombreuses solutions utilisent des systèmes de gestion de contexte. Afin d'accroître l'interopérabilité entre les différentes sources de contexte, de nombreuses solutions utilisent des ontologies pour la représentation du contexte [13, 14].

Les services de maison intelligente fournis par les dispositifs doivent avoir accès au contexte afin de prendre des décisions et de répondre aux besoins des utilisateurs. L'utilisation d'ontologies comme langage contextuel de haut niveau est une abstraction utile pour la communication des services, et crée un format de requête commun permettant aux services d'accéder au contexte fourni par différents gestionnaires de contexte.

3 MASH Architecture

Cette section développe l'analyse précédente pour présenter l'architecture de MASH (sigle de l'anglais "Multi-

1. Comme par exemple <https://nodered.org> et <https://ifttt.com>

Agent based Smart Home"), un système connectée à divers appareils intelligents, dirigé par les buts des utilisateurs et capable d'adapter son comportement à l'évolution du contexte d'une maison intelligente.

La Figure 1 montre les couches de cette architecture : système multi-agent ; Home'In² ; et l'environnement physique, c'est-à-dire les appareils déployés dans le monde réel. La couche multi-agent est composée d'agents autonomes qui contrôlent les appareils de la maison afin de fournir aux utilisateurs des services à valeur ajoutée. Leur architecture sera présentée ci-dessous. Suivant l'approche de programmation orientée multi-agent de [15], les agents ont un accès uniforme aux ressources et dispositifs de l'environnement via un ensemble d'entités non autonomes, appelées artefacts [16]. Elles encapsulent l'accès et le contrôle aux services et fonctions des appareils déployés dans l'environnement physique, tels que, par exemple, la télévision, le téléphone et le canapé. Des artefacts spécialisés offrent l'accès et le contrôle à Home'In.

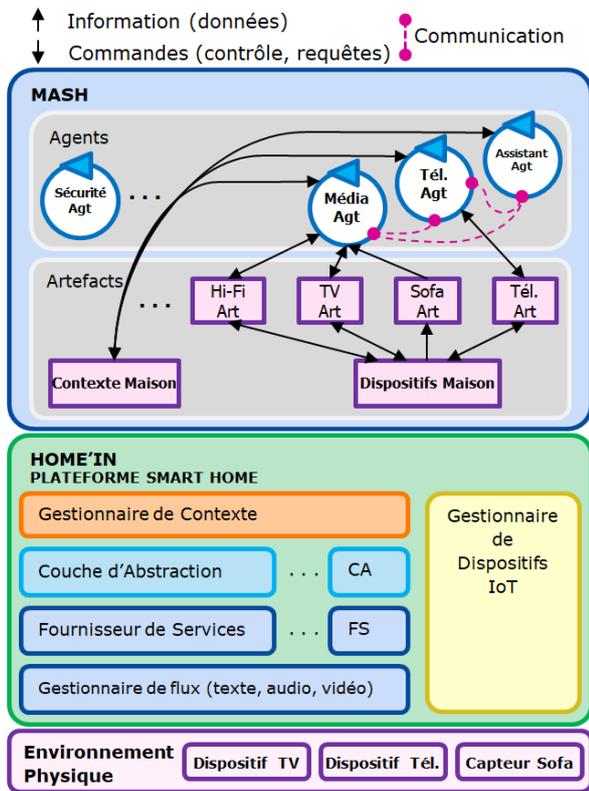


FIGURE 1 – Architecture Globale de MASH

La Fig. 1 représente une vue de haut niveau de la couche Home'In. Le *Gestionnaire de Contexte* reçoit des informations du monde physique via les dispositifs de l'environnement physique qui envoient en permanence des données à des *fournisseurs de services* dédiés de Home'In. Différentes *couches d'abstraction* transforment le format et le contenu de ces données, puis les stockent dans le *Gestionnaire de Contexte*.

2. La plateforme maison intelligente Home'In est ouverte, basée sur des technologies *open-source* et développée par l'entreprise Orange S.A.

naire de Contexte sous forme de triplets RDF. Le contexte est mis à jour en temps réel et ne gère pas d'historique. Le *Gestionnaire de Dispositifs IoT* fournit une interface directe pour contacter les appareils intelligents situés dans l'environnement physique du domicile et connectés à Home'In. Il constitue une façade cachant les protocoles et les formats hétérogènes de chaque dispositif. Les composants de Home'In sont inter-connectés par un mécanisme de publication/souscription via un bus MQTT. Grâce à lui, le *Gestionnaire de Contexte* peut écouter les données des couches d'abstraction et également celles du *Gestionnaire de Dispositifs IoT*, et stocker les données utiles afin de représenter le contexte domestique. La couche multi-agent peut ainsi envoyer des commandes aux dispositifs ou récupérer des informations qu'ils détectent dans l'environnement physique via l'artefact *Dispositifs Maison* qui interagit avec le *Gestionnaire de Dispositifs IoT* au travers du bus MQTT. Elle a également accès aux informations contextuelles via l'artefact *Contexte Maison* qui interagit avec le *Gestionnaire de Contexte* par une API REST.

L'approche de programmation orientée multi-agent [15] retenue permet de concevoir MASH comme un système modulaire, structurant la gestion intelligente des services en agents et les accès partagés aux ressources logicielles et physiques en artefacts. L'hétérogénéité des dispositifs est cachée aux agents en fournissant un ensemble uniforme d'actions et de propriétés observables par des artefacts qui les encapsulent. Les agents de la couche multi-agent sont ainsi indépendants de la plateforme Home'In. Les artefacts sont les seuls points d'ancrage à la plateforme ou à des dispositifs physiques en cas de connexion directe. Ces dispositifs peuvent changer (par exemple, apparition de nouveaux dispositifs, modification des caractéristiques des dispositifs actuels) sans impact sur les agents, tant que l'interface d'utilisation des artefacts reste la même. Par exemple, dans le cas d'une nouvelle télévision, la structure interne du TvArt qui le gère doit être adaptée. Le code des agents restera le même puisque les observations et les modalités d'actions de l'agent sur l'artefact TvArt restent les mêmes. Notons aussi que de nouveaux agents avec des capacités de raisonnement sur l'interface d'utilisation des artefacts peuvent être ajoutés sans impact sur ces artefacts.

Les agents de MASH fournissent des capacités de décision de haut niveau. Ils peuvent être des agents de "service" (centrés sur la prestation de services) ou "personnels" (centrés sur l'utilisateur), c'est-à-dire qu'ils agissent sur les besoins/demandes de l'utilisateur et automatisent les actions en respectant les préférences de l'utilisateur. Les agents de "service", par exemple MédiaAgt et TélAgt, s'occupent de l'exécution des dispositifs pour la réalisation de services à valeur ajoutée au nom des utilisateurs. Les agents "personnels" s'occupent des interactions entre l'utilisateur et le système. Ils jouent un rôle de majordome visant à automatiser les actions satisfaisant les buts de l'utilisateur en respectant ses préférences. Ils sont ainsi différents des "assistants personnels" tels que proposés dans [17] dont le rôle est d'aider l'utilisateur quand il n'arrive pas à trouver une solution par soi-même. Au delà des préférences

utilisateurs, les agents “personnels” gèrent aussi leurs valeurs [18] (e.g. : sécurité, famille, respect) et les activités routinières concernant l’usage des différents services. Ces agents “personnels” sont spécialisés pour aider le système à respecter, par exemple, l’envie de l’utilisateur de préserver l’environnement, ou de ne pas interrompre un moment en famille, ou de tout simplement afficher un film qui lui plaît quand l’utilisateur est sur le canapé.

Sur la base de [19] et [6], une version simplifiée de l’agent est présentée dans la Fig. 2. Cet agent a un comportement dirigé par les buts, c’est-à-dire qu’il agit pour atteindre les objectifs qu’il s’est fixé, tout en réagissant et en s’adaptant au monde. La fonction “Action” utilise la bibliothèque de méthodes prêtes à l’emploi, aussi appelées plans procéduraux (“Savoir-faire”) pour percevoir et agir dans l’environnement afin d’atteindre ou de maintenir des états contextuels. Une seconde bibliothèque de connaissances (“Savoir-quoi”) possède les modèles de ces méthodes pour créer des plans (séquence de méthodes) afin d’atteindre les buts. La fonction “Planification” utilise cette bibliothèque et les “Buts” définis pour raisonner au sein de l’agent avant l’exécution et créer ainsi un ensemble de plans. Ceux-ci seront utilisés par l’agent au moment de l’exécution pour atteindre ses buts.

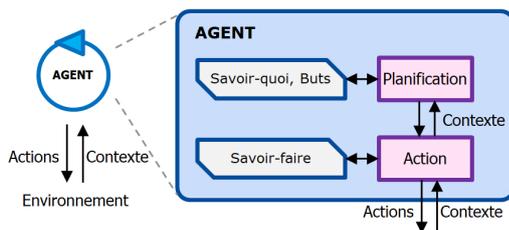


FIGURE 2 – Architecture simplifiée des Agents de MASH

Le système multi-agent a été conçu pour répondre aux exigences susmentionnées. Il possède des artefacts qui permettent d’accéder aux services disponibles. Les artefacts fournissent des informations aux agents par l’intermédiaire de *propriétés observables*, et l’accès aux actionneurs par l’intermédiaire d’*opérations*. La description de ces caractéristiques définit ce que nous appelons l’interface d’utilisation des artefacts. Ils produisent également des *signaux* en cas de succès/échec des opérations ou d’informations transitoires produites pendant l’exécution. Les artefacts peuvent, par exemple, effectuer des calculs lourds, ou informer sur les conditions météorologiques, ainsi qu’exposer les *opérations* à des objets de contrôle comme allumer la télévision ou répondre à des appels. Nous faisons la différence entre les artefacts qui servent de mandataires aux appareils connectés (appelés Artefacts Mandataires Domestiques, en anglais Home Proxy Artifacts) et les artefacts fournissant un accès de haut niveau aux composants de Home’In (appelés Artefacts Utilitaires Domestiques, en anglais Home Utilities Artifacts). Les artefacts SofaArt, TvArt et TélArt sont des exemples d’Artefacts

Mandataires Domestiques (cf. Fig. 1). Le SofaArt ne fournit aucune opération et informe uniquement de la présence de l’utilisateur (s’il est assis dessus). TélArt et TvArt représentent le téléphone et la télévision de l’utilisateur (ce sont des interfaces pour exécuter des opérations et récupérer des données). Les artefacts “Contexte Maison” et “Dispositifs Maison” sont des exemples d’Artefacts Utilitaires Domestiques. Ils n’ont pas de contrepartie directe dans le monde réel. Le premier se connecte au Gestionnaire de Contexte de Home’In, et le second au Gestionnaire des Dispositifs IoT. Leur fonction est de créer une interface d’utilisation facile pour les agents, en cachant les spécificités de Home’In.

4 Intégration du Contexte dans MASH

Dans l’architecture décrite en Fig. 1, les agents de MASH accèdent au contexte en utilisant directement l’artefact *Contexte Maison* ou indirectement l’artefact *Dispositifs Maison* via les artefacts mandataires domestiques. Alors que l’artefact *Contexte Maison* fournit un *accès général* aux informations contextuelles de la maison intelligente en récupérant les données du gestionnaire de contexte, les artefacts mandataires domestiques via l’artefact *Dispositifs Maison* fournissent un *accès direct* aux informations contextuelles détectées et construites par les dispositifs intelligents eux-mêmes.

L’artefact *Dispositifs Maison* fournit des informations contextuelles en se connectant au Gestionnaire des Dispositifs IoT, grâce auquel il peut accéder aux données de l’appareil pour lequel il agit en tant que mandataire. Il s’agit par exemple de données de capteurs (par exemple, la présence de l’utilisateur) et de croyances spécifiques (par exemple, la quantité de places dans le canapé). Le choix d’accéder à partir de l’appareil local plutôt que par le gestionnaire de contexte est lié aux limitations du Gestionnaire de Contexte de Home’In. Pour illustrer cela, considérons les connexions agents-artefacts de la Fig. 1 : puisque le gestionnaire de contexte ne stocke pas l’heure actuelle du film regardé, et étant donné que TélArt en a besoin (pour choisir de sonner ou d’attendre la fin du film), il peut le demander directement (heure actuelle du film) à TvArt. Les descriptions des artefacts et les opérations accessibles aux agents sont décrites dans la section 6.

L’artefact *Contexte Maison* se connecte au gestionnaire de Contexte de Home’In qui stocke et combine un ensemble plus large de données. Cette connexion se réalise par le biais de l’API REST de Home’In. Comme indiqué précédemment, le gestionnaire de contexte stocke un graphe dont les nœuds et les relations sont conformes à des ontologies spécifiques, comme par exemple “Friend of a Friend” (FOAF) [20] et “Semantic Sensor Network” (SSN) [21]. Il fournit ainsi des données contextuelles représentées dans un langage commun et une sémantique partagée par les agents de MASH. Ces ontologies décrivent des objets et des relations, représentant des concepts importants du contexte de la maison. Par exemple, FOAF décrit le nom

de l'utilisateur ainsi que toutes les personnes qu'il connaît, telles que la famille et les collègues de travail, et l'ontologie SSN décrit le capteur de canapé et ses observations.

Les opérations de l'artefact *Contexte Maison* sont utilisées par un agent pour enrichir ses croyances à partir du graphe de contexte du Gestionnaire de Contexte de Home'In avec le contexte utile pour son raisonnement. Etant des opérations d'un artefact, elles apparaissent dans le corps des méthodes de l'agent et sont exécutées lorsque ces méthodes ont été sélectionnées par l'agent. Ces opérations possèdent trois paramètres : une variable, une requête définissant une formule littérale utilisée pour trouver des instances de la variable dans le graphe de contexte, et la réponse avec les instances trouvées. L'ensemble d'opérations est le suivant :

- *isTrue* : renvoie un booléen s'il y a des nœuds dans le graphe de contexte correspondant à la requête
- *count* : renvoie le nombre de nœuds dans le graphe de contexte correspondant à la requête
- *findOne* : renvoie le premier nœud correspondant à la variable d'interrogation dans le graphe de contexte
- *findAll* : renvoie une liste de toutes les instances correspondant aux variables de la requête dans le graphe de contexte

Par exemple, pour obtenir les valeurs du statut de toute entité étant un *humanUser*, l'agent construira la requête *findAll(X, status(humanUser, X), Reply)* correspondant à la recherche des instances de la variable *X* satisfaisant la formule littérale *status(humanUser, X)* dans le graphe de contexte. La variable *Reply* pourra ainsi être unifiée avec [*watchingMovie, sitting*] correspondant à l'état actuel du graphe de contexte.

Enfin, l'interface de cet artefact expose également l'opération *subscribeBeliefs* pour s'abonner aux mises à jour du graphe de contexte. Contrairement aux opérations précédentes, elle n'a pas de retour explicite : elle enregistre la requête passée en paramètre et récupère continuellement tous les résultats du graphe de contexte, mettant à jour les propriétés observables correspondantes dans l'artefact. Les agents qui observent cet artefact recevront ainsi automatiquement des mises à jour sur les changements de contexte correspondant à la requête. Au travers de cette opération, l'artefact agit ainsi comme un filtre sur les mises à jour du graphe de contexte, ne donnant aux agents que ce à quoi ils se sont abonnés. Par exemple, nous pouvons considérer l'opération *subscribeBeliefs(status(movie, X))*, qui souscrit aux mises à jour concernant la valeur de l'état d'un film (par exemple : terminé, en cours); *X* est une variable se référant à l'état. La propriété observable dans l'artefact a le format *status(movie, X)*. En conséquence, l'agent a la croyance correspondant à la propriété observable de l'artefact, et est mis à jour dès que l'artefact l'est.

5 Intégration du Contrôle de Dispositifs dans MASH

Comme nous l'avons vu dans la section 3, un agent est une entité dirigée par des buts, capable de percevoir son envi-

ronnement et d'agir sur celui-ci et de contrôler les appareils de la maison. Les agents peuvent utiliser les artefacts (par exemple TvArt, TélArt) à cette fin. Dans notre scénario, les opérations de ces artefacts contrôleront leurs homologues du monde réel. Un avantage de l'architecture présentée est que les agents de la couche MASH n'ont pas besoin de connaître les spécificités de bas niveau des appareils IoT (ex : les protocoles), puisque les artefacts de MASH permettent d'interagir de manière uniforme via les opérations offertes.

Par exemple, lorsque l'appareil téléphonique commence à sonner, MédiaAgt peut choisir de mettre en sourdine le film que l'utilisateur regarde et d'activer ses sous-titres. Pour le faire, le MédiaAgt active les opérations *stopSpeaker* et *activateSubtitles* du TvArt. L'artefact *Dispositifs Maison* transmet ces commandes à l'aide du bus MQTT au *Gestionnaire de Dispositifs IoT* de Home'In, qui contactera alors finalement le véritable appareil de télévision qui effectuera les opérations de bas niveau.

Le *Gestionnaire de Dispositifs IoT* est une façade cachant les protocoles et formats hétérogènes de chaque appareil, connectant Home'In à l'ensemble des appareils intelligents. Par exemple, TvArt dispose de l'opération *activateSubtitles*, qui sera traduite dans le message JSON :

```
{ "mashup": "myTV.activateSubtitles()" }
```

Ce message sera envoyé à Home'In par l'artefact en utilisant son lien avec l'artefact *Dispositifs Maison*, qui construit le message ci-dessous et l'envoie au *topique MQTT "/home/iot"*, topique réservé au composant *Gestionnaire de Dispositifs IoT* :

```
{
  "id": "message 01",
  "timestamp": 876545678000,
  "source": "mediaAgt",
  "data": {
    "mashup": "tv.activateSubtitles()"
  }
}
```

Le *Gestionnaire de Dispositifs IoT* utilisera le protocole du dispositif pour envoyer la commande au dispositif réel. Les opérations des appareils disponibles et les topiques MQTT, ainsi que le format des messages MQTT, sont spécifiés par les développeurs Home'In.

Par conséquent, l'artefact *Dispositifs Maison*, associé au *Gestionnaire de Dispositifs IoT*, crée une interface uniforme et stable qui peut être utilisée par les agents pour interagir avec les dispositifs disponibles dans l'environnement domestique.

6 Implémentation et Expériences

La couche multi-agent de MASH a été implémentée en utilisant JaCaMo, une plateforme de programmation orientée multi-agent [15]. Les agents ont été programmés en s'appuyant sur le langage de programmation agent de cette plateforme. Les méthodes (i.e. plans procéduraux) des agents de MASH, qui ont une architecture BDI (Belief-Desire-Intention) [22], sont traduites en "plans Jason". Les *Arte-*

facts Utilitaires Domestiques et *Artefacts Mandataires Domestiques* sont codés comme des artefacts CArAgO de la plateforme JaCaMo.

Le Gestionnaire de Contexte de Home'In utilise un triplestore RDF pour stocker le graphe représentant le contexte de la maison. Il est exposé par une API REST avec un point d'accès SPARQL, permettant à l'artefact *Contexte Maison* de l'interroger. Le Gestionnaire de Dispositifs IoT utilise plusieurs protocoles pour se connecter à des objets externes. Dans Home'In, tous les composants utilisent un bus MQTT pour communiquer.

L'artefact *Contexte Maison* est initialisé avec l'API REST; il possède les 5 opérations vues dans la section 4; il convertit entre les formats SPARQL, XML et Jason; il connaît les ontologies, listant ses préfixes et ses URI, et les relations entre les ontologies; et il possède 2 ensembles de données, un avec la liste des requêtes pour récupérer les croyances souscrites, et un avec les valeurs des croyances actuelles (pour créer les propriétés observables des artefacts).

L'artefact *Dispositifs Maison* est initialisé avec le point d'accès du bus MQTT, le topique réservé au *Gestionnaire de Dispositifs IoT*, et une liste de dispositifs qu'il doit écouter. Il attendra les mises à jour de ces dispositifs et les transmettra ensuite aux *Artefacts Mandataires Domestiques* correspondants.

6.1 Cas d'Étude

Pour illustrer le système MASH, nous considérons le cas d'étude suivant, impliquant deux utilisateurs (David et Iaco), limitant les services à l'utilisation de trois objets connectés et intelligents (une télévision, un téléphone et un canapé). Le déroulement du scénario du cas d'étude est le suivant :

1. David, à l'aide de son téléphone, demande à la maison de regarder un film; prenant en compte les habitudes de David, ce film sera diffusé sur la télévision du salon;
2. En arrivant dans son salon, David s'assoie sur le canapé, qui l'identifie; ceci déclenche la mise en place du service télévision et affichage du film;
3. A l'arrivée de Iaco sur le même canapé, celui-ci le reconnaît également, mais rien ne change sur l'affichage du film vu que Iaco ne fait aucune demande;
4. Après quelques minutes, David reçoit un appel de son chef; ce message est renvoyé à la messagerie respectant ainsi la préférence de David stipulant qu'un moment de loisir est plus important que d'accepter la subordination liée au travail;
5. Un moment plus tard, un appel de la mère de David est reçu; respectant à nouveau les préférences de David stipulant que la famille est plus importante que le loisir, le téléphone sonne;
6. David sort du salon; le canapé détecte que David n'est plus présent;
7. La télévision, prenant en compte les préférences de Iaco, lui propose de lire un livre;

8. Iaco répond positivement; la télévision s'éteint;
9. À la fin de son appel avec sa mère, David revient au salon et s'assoie dans le canapé; avec l'accord de Iaco il demande à la télévision de s'allumer;
10. Ayant reconnu la présence de David, la télévision propose de reprendre la diffusion de l'émission à partir du moment où David était sorti de la pièce;
11. David ayant répondu positivement, la télévision reprend la diffusion de l'émission.

Ce cas d'étude vise à explorer la couche multi-agent de MASH, et plus particulièrement la personnalisation des services à ses utilisateurs, en prenant compte leurs préférences dans une situation multi-utilisateur.

6.2 Schéma de Déroulement du Scénario

Pour illustrer la manière dont les composants de gestion du contexte et des dispositifs IoT sont utilisés dans le système MASH, considérons la première et la deuxième étape du scénario présenté ci-dessus, quand l'utilisateur demande l'affichage d'un film.

Cette étape met en situation les agents MédiaAgt, TélAgt et AssistantAgt, et les artefacts SofaArt, TvArt et TélArt. L'agent MédiaAgt utilise TvArt pour contrôler la diffusion du contenu médiatique sur la télévision et utilise SofaArt pour obtenir des informations sur la présence de l'utilisateur dans le canapé. L'agent TélAgt, de son côté, utilise TélArt pour contrôler les opérations d'appel du téléphone. Les trois agents communiquent à l'aide de messages. Ils connaissent tous les services (par exemple, diffusion de contenu médiatique, appel téléphonique, détection de personnes) proposés par les autres agents.

Les premières étapes du scénario, correspondant à la diffusion du film demandé par l'utilisateur via son téléphone, sont présentées sur la Fig. 3, et détaillées ci-dessous :

1. David, utilisant son téléphone connecté à Home'In, demande à regarder "A.I. movie", message qui est envoyé au Gestionnaire de Dispositifs IoT de Home'In. Il est envoyé comme un message JSON par le bus MQTT au Gestionnaire de Dispositifs IoT. Les messages échangés sont (voir Fig. 3) :
(1.1) "Bonjour, diffuse le film "A.I. movie" !"
(1.2) achieve : watch "A.I. movie"
2. Le Gestionnaire de Dispositifs IoT reçoit le message, le traduit dans un format compréhensible par tous les services du domicile et l'envoie sur le bus MQTT. L'artefact *Dispositifs Maison* est à l'écoute du topique MQTT réservé au Gestionnaire de Dispositifs IoT, et donc capte cette nouvelle demande. Le message échangé est (voir Fig. 3) :
(2) {id:1, source:iot, data:watch("A.I. movie")}
3. Le TélArt reçoit la demande sous forme de signal de *Dispositifs Maison*, et émet le même signal à tous les agents qui écoutent TélArt (voir Fig. 3) :
(3.1) «signal» requestMovie("A.I. movie")
(3.2) «broadcast» requestMovie("A.I. movie")

```
requestMovie("A.I. movie")
```

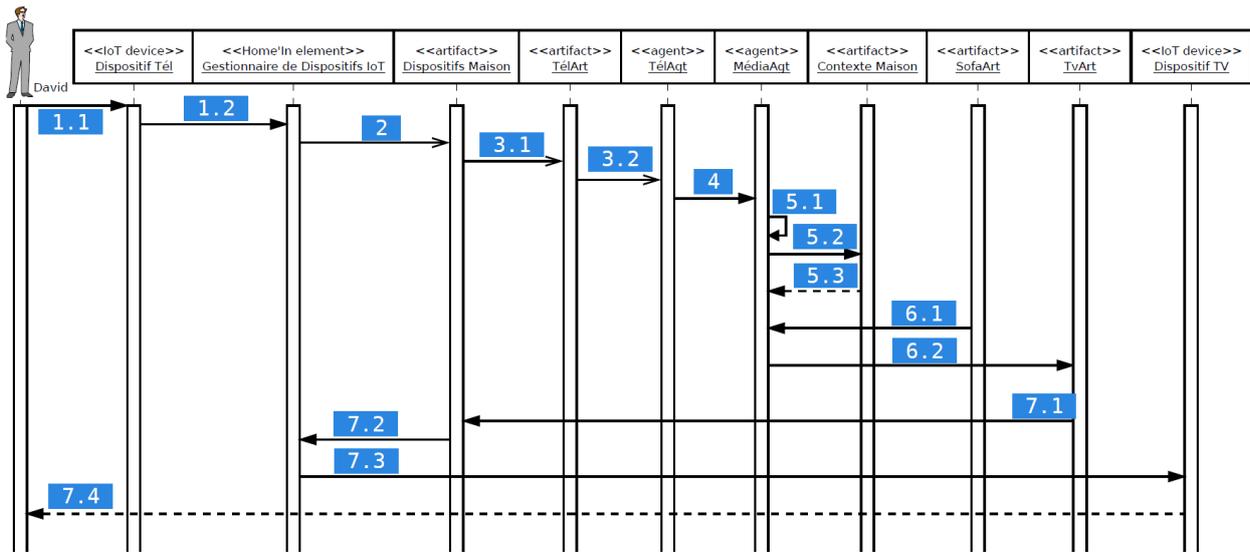


FIGURE 3 – Déroulement du Scénario

4. L'agent TélAgt reçoit le signal et l'envoie (voir ci-dessous) sous forme de message à l'agent MédiaAgt pour qu'il le traite, puisqu'il est lié aux services médiatiques. Le message échangé est (voir Fig. 3) : (4) requestMovie("A.I. movie")

```
.send(mediaAgt, tell,
      requestMovie("A.I. movie"))
```

5. MédiaAgt le reçoit, et décide de créer un but pour observer un film : !watchMovie("David", "A.I. movie"), qui est exécuté par la méthode

```
+!watchMovie(User, Movie) <-
  findAll(X, near(television, X),
          Reply);
!waitNearCondition(User, Reply);
!playMedia(Movie);
.wait({
  +finishedPlayingMedia(Movie)}).
```

La première action (findAll) interroge tous les objets proches de la télévision (le résultat est stocké dans la variable Reply) en générant une requête SPARQL (illustrée ci-dessous) au Gestionnaire de Contexte en utilisant l'artefact Contexte Maison; la requête est une expression dans laquelle "<home:tv>" est le sujet, "<is:near>" le prédicat, et "?x" l'objet (qui dans ce cas est une variable, i.e. l'information que nous voulons découvrir)

```
SELECT ?x
WHERE <home:tv> <is:near> ?x
```

Le résultat est "sofa". L'action suivante (!waitNearCondition) crée un sous-but, qui correspond à attendre que l'utilisateur soit près d'un des objets renvoyés par la requête, puis l'exécution continue. Les messages échangés sont (voir Fig. 3) :

```
(5.1) +!watchMovie(David, "A.I. movie")
(5.2) SELECT ?x WHERE <home:tv> <is:near> ?x
(5.3) x = sofa
```

6. Lorsque le capteur du canapé met à jour la propriété observable de SofaArt avec la présence positive de l'utilisateur, le média qui se concentre sur le SofaArt l'obtient. Son sous-but !waitNearCondition est atteint, ainsi MédiaAgt démarre le sous-but pour jouer le film, qui utilisera les opérations de TvArt. Les messages échangés sont (voir Fig. 3) :

```
(6.1) <home:tv> <is:near> <user:David>
(6.2) play("A.I. movie")
```

7. TvArt utilise Dispositifs Maison pour envoyer un message MQTT au topique "/home/iot" du Gestionnaire de Dispositifs IoT, qui envoie ensuite la commande au dispositif TV (voir Fig. 3) :

```
(7.1) play("A.I. movie")
(7.2) {id:2, source:tv, data:play("A.I. movie")}
(7.3) video : "A.I. movie", audio : "A.I. movie"
(7.4) Show "A.I. movie"
```

Comme nous l'avons vu, le système effectue de nombreuses actions après avoir reçu la demande d'un film de la part de l'utilisateur. Ces actions sont transparentes pour l'utilisateur, qui voit seulement le résultat final : le film diffusé sur la télévision. Ce faisant, le système doit donner à l'utilisateur une réponse en temps utile.

Le scénario montre l'intégration technologique des dispositifs IoT et la gestion du contexte. Comme ceux utilisés dans Home'In, par exemple RDF, SPARQL, MQTT, JSON et XML, et ceux dans MASH, par exemple JaCaMo, Jason (langage AgentSpeak), CArtAgO (Java). Il montre également deux aspects traités dans MASH : la connaissance du contexte (par exemple, le système attend la présence de l'utilisateur pour lancer le film), et le comportement dirigé par les buts (par exemple, la création d'un but "regarder un film" et son exécution).

7 Travaux Connexes

Les travaux susmentionnés présentent des différences importantes par rapport aux mashups IoT. Ces dernières créent souvent des services ad hoc pour des besoins spécifiques [12, 10]. Notre solution de maison intelligente basée sur une approche multi-agent, met en oeuvre un ensemble de services gérés par des agents qui utilisent le raisonnement et l'adaptation en fonction du contexte. Sa conception modulaire permet d'étendre les fonctionnalités et les services par l'incorporation de nouveaux agents, ou de nouveaux artefacts en fonction des dispositifs introduits.

La solution décrite dans cet article s'intéresse également à l'hétérogénéité du système. Elle identifie le Web des Objets (WoT), une technologie permettant aux objets de se connecter et de communiquer en utilisant l'infrastructure web existante [9] réduisant l'écart entre les entités hétérogènes. Notre approche qui utilise REST et MQTT pour la communication dans un système fermé pourrait bénéficier des solutions WoT, tirant ainsi parti des formats et des canaux de communication unifiés, ainsi que du raisonnement des agents. Le système multi-agent MASH vise à présenter un comportement adaptatif intelligent au-delà de la définition de règles et de contraintes comme dans [23], mais en intégrant une capacité d'adaptation en fonction du contexte des plans et des actions des dispositifs. Un tel comportement adaptatif peut être obtenu par une composition flexible des applications utilisant les services des dispositifs disponibles, comme dans [24], qui se distingue de MASH qui a un comportement coopératif plus décentralisé où les dispositifs ne sont pas seulement des appareils mais sont contrôlés par des agents intelligents et coordonnés.

Des travaux récents [1] ont identifiés des besoins des systèmes multi-agents pour la maison connectée, et notre système MASH répond à plusieurs capacités demandées :

- sensibilité au contexte : MASH a un composant dédié pour récupérer le contexte et informer les agents de ses mises à jour,
- adaptation : le système adapte ses services dès qu'il y a des changements de contexte,

Les autres capacités décrites dans cet article telles que "ne pas déranger les utilisateurs", "planifier" et "expliquer" n'ont pas été décrites dans cet article et seront l'objet de développements ultérieurs. Grâce au comportement autonome des agents, les interactions non essentielles avec les utilisateurs pourront être évitées ; chacun des agents a une couche de raisonnement dédiée à la planification.

Enfin, l'intégration de MASH avec la plateforme Home'In présente la solution *Gestionnaire de Contexte* sous forme d'un graphe de triplets centralisé pour gérer le contexte commun partagé entre les entités et les appareils de la maison intelligente, ce qui constitue une approche différente par rapport à [25]. Ce dernier utilise un système multi-agent pour effectuer la gestion du contexte, alors que le premier utilise le graphe de triplets plutôt comme un artefact fournissant des services de gestion du contexte.

8 Conclusions

Cet article s'est concentré sur l'intégration du contexte et des appareils IoT présents dans une maison intelligente au sein d'une application de maison intelligente. Nous avons proposé une architecture multi-agent permettant au système de la maison intelligente, qui interagit directement avec les humains, de fournir des services basés sur les appareils IoT disponibles et de s'adapter aux environnements dynamiques et aux besoins des utilisateurs.

La prise en compte du contexte a été abordée par l'intégration d'un système de gestion du contexte basé sur les technologies du web sémantique et les ontologies. L'utilisation des ontologies comme format commun pour l'échange de croyances entre les agents et les services pourrait aider les futures versions à faire face aux problèmes d'interopérabilité. La version envisagée a mis en oeuvre des mécanismes permettant la demande de données contextuelles à la demande et par des méthodes d'abonnement. Les spécificités de ces mécanismes ont été masquées aux agents grâce à la couche d'abstraction d'artefacts spécialisés.

Le comportement dirigé par les buts a été abordé à deux niveaux : le développement d'un modèle architectural d'agent qui adapte les actions en respectant les objectifs de l'utilisateur, et l'intégration de dispositifs hétérogènes utilisant des composants spécialisés, qui exposent les actions disponibles pour permettre la poursuite des objectifs en agissant dans le monde.

Nous avons mis en oeuvre cette architecture et illustré la prise en compte du contexte et le comportement dirigé par les buts dans un scénario de maison intelligente : un utilisateur qui regarde un film reçoit un appel. La conception de notre solution est générique pour être appliquée à d'autres situations impliquant l'adaptation d'agents dirigés par les buts et la gestion de données contextuelles. Nous avons étudié telles exigences dans un environnement industriel, avec l'intégration d'un système multi-agent (MASH) et d'une plateforme de maison intelligente (Home'In) actuellement développée par Orange S.A., ce qui nous a permis d'explorer le fonctionnement de MASH sur une plateforme existante. Ceci n'a cependant pas remis en cause l'indépendance de la couche multi-agent de MASH vis-à-vis d'autres plateformes vu que seulement les artefacts doivent être adaptés si la couche Home'In était remplacée. Une limitation identifiée est que le Gestionnaire de Dispositifs IoT effectue une attente active ("polling" en anglais) afin de mettre à jour ses croyances toutes les secondes, même si aucun changement n'est intervenu. Dans les versions futures il pourra utiliser le protocole MQTT, c'est-à-dire qu'un client MQTT pourra écouter des topiques spécifiques, en poussant les modifications de l'artefact uniquement lorsqu'elles se produisent.

Les travaux futurs porteront sur la gestion du comportement des agents. L'agent présenté a montré un comportement adaptatif en utilisant les méthodes disponibles pour effectuer des actions, qui peuvent être étendues pour prendre en compte les valeurs humaines de l'utilisateur, permettant au processus de prise de décision d'utiliser les

préférences humaines sur la façon d'atteindre leurs buts. En outre, l'ajout ou le retrait d'agents et de dispositifs existants peut nécessiter que le système découvre de nouvelles méthodes et de nouveaux services, ainsi que la continuité de la disponibilité des anciens.

Une telle adaptation nécessitera le recours à des entités spécialisées, et pourra se fonder sur les solutions actuelles proposées dans Home'In et d'autres solutions Orange S.A., telles que Thing'In, qui dispose d'un graphe avec les descriptions et les relations des objets, centralisant les informations à long terme et fournissant des éléments de base pour la découverte de services. Orange S.A. travaille actuellement sur un cas d'étude basé sur Home'In, dans lequel un groupe d'étudiants a ses appartements équipés d'objets connectés IoT. Les données de ce cas d'étude sont utilisées pour améliorer la qualité des services et produits offerts par Orange S.A. Il est envisagé que, dans les versions futures de ce cas d'étude, MASH sera intégré à Home'In pour explorer la possibilité de personnalisation des services à ses utilisateurs, en prenant compte par exemple de ses préférences, habitudes et valeurs humaines.

Références

- [1] D. N. MEKURIA, P. SERNANI, N. FALCIONELLI et A. F. DRAGONI, "Reasoning in Multi-agent Based Smart Homes : A Systematic Literature Review", en, in *Ambient Assisted Living*, A. LEONE, A. CAROPPO, G. RESCIO, G. DIRACO et P. SICILIANO, éd., t. 544, Cham : Springer International Publishing, 2019, p. 161-179, ISBN : 978-3-030-05920-0 978-3-030-05921-7. DOI : 10.1007/978-3-030-05921-7_13. (visité le 19/04/2020).
- [2] M. R. ALAM, M. B. I. REAZ et M. A. M. ALI, "A Review of Smart Homes Past, Present, and Future", en, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, t. 42, n° 6, p. 1190-1203, nov. 2012, ISSN : 1094-6977, 1558-2442. DOI : 10.1109/TSMCC.2012.2189204.
- [3] R. J. ROBLES et T.-h. KIM, "Review : Context Aware Tools for Smart Home Development", en, *International Journal of Smart Home*, t. 4, n° 1, p. 12, jan. 2010.
- [4] S. MAYER, D. PLANGGER, F. MICHAHELLES et S. ROTHFUSS, "UberManufacturing : A Goal-Driven Collaborative Industrial Manufacturing Marketplace", in *Proceedings of the 6th International Conference on the Internet of Things, IOT 2016, Stuttgart, Germany, November 7-9, 2016*, S. SCHNEEGASS, A. SCHMIDT, F. MICHAHELLES, M. KRITZLER, A. ILIC et K. KUNZE, éd., ACM, 2016, p. 111-119. DOI : 10/gfxpgz.
- [5] A. CIORTEA, S. MAYER et F. MICHAHELLES, "Repurposing Manufacturing Lines on the Fly with Multi-agent Systems for the Web of Things", in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, E. ANDRÉ, S. KOENIG, M. DASTANI et G. SUKTHANKAR, éd., International Foundation for Autonomous Agents et Multiagent Systems Richland, SC, USA / ACM, 2018, p. 813-822.
- [6] I. TRENTIN, O. BOISSIER et F. RAMPARANY, "Insights about user-centric contextual online adaptation of coordinated multi-agent systems in smart homes", en, *Rencontres des Jeunes Chercheurs en Intelligence Artificielle 2019, Jul 2019, Toulouse, France*,
- [7] M. LEFRANÇOIS, "Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Patterns", en, *SEMAN-TiCS 2017 workshops proceedings : SIS-IoT September 11-14, 2017, Amsterdam, Netherlands*, p. 8, 2017.
- [8] M. KOVATSCH, R. MATSUKURA, M. LAGALLY, T. KAWAGUCHI, K. TOUMURA et K. KAJIMOTO, *WEB OF THINGS AT W3C*, en, mai 2019.
- [9] M. BLACKSTOCK et R. LEA, "IoT Mashups with the WoTKit", *2012 3rd IEEE International Conference on the Internet of Things*, 2012. DOI : 10.1109/IOT.2012.6402318.
- [10] M. KOVATSCH, Y. N. HASSAN et S. MAYER, "Practical semantics for the Internet of Things : Physical states, device mashups, and open questions", in *5th International Conference on the Internet of Things, IOT 2015, Seoul, South Korea, 26-28 October, 2015*, IEEE, 2015, p. 54-61. DOI : 10/gfxpgw.
- [11] F. LÉCUÉ, "Composition de Services Web : Une Approche basée Liens Sémantiques. (Web Service Composition : Semantic Links based Approach)", PhD Thesis, École nationale supérieure des mines de Saint-Étienne, France, 2008.
- [12] D. GUINARD, V. TRIFA, T. PHAM et O. LIECHTI, "Towards physical mashups in the Web of Things", en, in *2009 Sixth International Conference on Networked Sensing Systems (INSS)*, Pittsburgh, PA, USA : IEEE, juin 2009, p. 1-4, ISBN : 978-1-4244-6313-8. DOI : 10/bts5dq. adresse : http://ieeexplore.ieee.org/document/5409925/.
- [13] C. PERERA, A. B. ZASLAVSKY, P. CHRISTEN et D. GEORGAKOPOULOS, "Context Aware Computing for The Internet of Things : A Survey", *IEEE Communications Surveys and Tutorials*, t. 16, n° 1, p. 414-454, 2014. DOI : 10/x9w.
- [14] J. EUZENAT, J. PIERSON et F. RAMPARANY, "Dynamic context management for pervasive applications", *Knowledge Eng. Review*, t. 23, n° 1, p. 21-49, 2008. DOI : 10/fg2bbs.

- [15] O. BOISSIER, R. H. BORDINI, J. F. HÜBNER, A. RICCI et A. SANTI, “Multi-agent oriented programming with JaCaMo”, *Sci. Comput. Program.*, t. 78, n° 6, p. 747-761, 2013. DOI : 10/b9crfb.
- [16] A. OMICINI, A. RICCI et M. VIROLI, “Artifacts in the A&A meta-model for multi-agent systems”, en, *Autonomous Agents and Multi-Agent Systems*, t. 17, n° 3, p. 432-456, déc. 2008, ISSN : 1387-2532, 1573-7454. DOI : 10.1007/s10458-008-9053-x.
- [17] E. PLATON, N. SABOURET et S. HONIDEN, “T-compound : An Agent-Specific Design Pattern”, *Proceedings of 3rd international workshop on Agent Oriented Methodologies at OOPSLA 2004*, 2004.
- [18] S. H. SCHWARTZ, “Universals in the Content and Structure of Values : Theoretical Advances and Empirical Tests in 20 Countries”, en, in *Advances in Experimental Social Psychology*, t. 25, Elsevier, 1992, p. 1-65, ISBN : 978-0-12-015225-4. DOI : 10.1016/S0065-2601(08)60281-6.
- [19] M. GHALLAB, D. S. NAU et P. TRAVERSO, *Automated Planning and Acting*. Cambridge University Press, 2016, ISBN : 978-1-107-03727-4.
- [20] D. BRICKLEY et L. MILLER, *FOAF Vocabulary Specification*, en, jan. 2014.
- [21] W3C, *Semantic Sensor Network Ontology*, en, déc. 2017. adresse : <https://www.w3.org/TR/vocab-ssn/>.
- [22] R. H. BORDINI, J. F. HÜBNER et M. WOOLDRIDGE, *Programming Multi-Agent Systems in AgentSpeak using Jason*. 2007.
- [23] P. RUST, G. PICARD et F. RAMPARANY, “Using Message-Passing DCOP Algorithms to Solve Energy-Efficient Smart Environment Configuration Problems”, in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, S. KAMBHAMPATI, éd., IJCAI/AAAI Press, 2016, p. 468-474.
- [24] M. VALLÉE, “Un intergiciel multi-agent pour la composition flexible d’applications en intelligence ambiante. (A multi-agent middleware for flexible application composition in ambient intelligence)”, PhD Thesis, École nationale supérieure des mines de Saint-Étienne, France, 2009.
- [25] A. SORICI, G. PICARD, O. BOISSIER, A. ZIMMERMANN et A. M. FLOREA, “CONCERT : Applying semantic web technologies to context modeling in ambient intelligence”, *Computers & Electrical Engineering*, t. 44, p. 280-306, 2015. DOI : 10/gfxpgs.

