



AfIA

Association française
pour l'Intelligence Artificielle

JFSMA

Journées Francophones sur les Systèmes Multi-Agents

PFIA 2022



Table des matières

Valérie CAMPS	
Éditorial	5
Comité de programme	7
Session 1 : Résolution collective de problèmes, coordination, travail en équipe	9
G. Picard	
Coordination de trajectoires 4D par optimisation distribuée dans la gestion du trafic aérien sans pilote	10
E. Beauprez, A.-C. Caron, M. Morge, J.-C. Routier	
Échange de tâches pour la réduction de la durée moyenne de réalisation	20
Session 2 : Négociation multi-agent, consensus	30
J. Guéron, G. Bonnet	
Un protocole de concessions monotones pour la formation distribuée de coalitions	31
A. Milot, E. Chauveau, S. Lacroix, C. Lesire	
Allocation par enchères et planification hiérarchique pour un système multirobot - Application au cas de la chasse aux mines sous-marines	41
Session 3 : Communication	51
L. Dupuis de Tarlé, G. Pigozzi, J. Rouchier	
Spécificité de l'argumentation scientifique dans un débat	52
Y. Bourahla, M. Atencia, J. Euzenat	
Transmission de connaissances et sélection	61
Session 4 : Émergence, auto-organisation	71
H.-N. Ngo, E. Kaddoum, M.-P. Gleizes, J. Bonnet, A. Goursolle	
P-ADRIP : un système multi-agent auto-organisateur pour la prévision du trafic routier	72
D. Andrea Guastella, V. Camps, M.-P. Gleizes	
Estimation de données environnementales manquantes sans déploiement de capteurs supplémentaires : le système HybridIoT	82
Session 5 : Simulation, modélisation	92
P. Gontier, N. Bareille, S. Picault	
Dairy Health Manager : un simulateur multi-agents flexible pour l'étude des maladies des animaux d'élevage	93
T. De Blauwe, N. Sabouret, D. Lourdeaux	
Méta-modèle d'agent pour la génération de comportements variables induits par des modèles cognitifs modulaires	103
F. Taillandier, A. Moatty, C. Curt, P. Di Maiolo, B. Beullac, P. Brueder	
La modélisation Agent pour sensibiliser aux Solutions Fondées sur la Nature pour gérer les inondations	113
Session 6 : Démonstrations et posters	123
E. Grislin-Le-Strugeon, H. Ouarnoughi	
Modélisation et simulation multi-agents pour la gestion des tâches entre véhicules et infrastructures dans le cadre de la ville intelligente	124

J. Cumin, F. Ramparany, I.-F. Trentin, O. Boissier Maisons Collaboratives	128
I. Taleb, G. Guerard, F. Fauberteau, N. Nguyen Modélisation Multi-Agent pour les Réseaux Énergétiques Insulaires	132
J.-D. Collard, E. Stattner, P. Gergos Une approche collaborative pour la recherche de places de stationnement	136
B. Mermet, G. Simon Comment les échecs de preuve peuvent aider à la correction de spécifications erronées de Systèmes Multi-Agents	140
M.T. Bennai, L. Gallay, Z. Guessoum, S. Cormier, S. Mazouzi, Apprentissage profond & systèmes multi-agents - Application à la segmentation d'images médicales	141
B. Perez, C. Lang, J. Henriet, L. Philippe, F. Auber Architecture SMA/RàPC enrichie d'outils d'aide à la décision pour prédire les risques en chirurgie	142

Éditorial

Journées Francophones sur les Systèmes Multi-Agents

Les Journées Francophones sur les Systèmes Multi-Agents (JFSMA) sont le rendez-vous annuel de la communauté des chercheurs francophones travaillant dans le domaine des Systèmes Multi-Agents (SMA).

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 sociales (économie, sociologie, etc.) et aux sciences naturelles (éthologie, etc.). Ces journées visent à faire le point sur les différents travaux de la communauté scientifique du domaine mais également à créer un espace de débats d'idées et d'échanges de proximité entre les chercheurs confirmés et les jeunes chercheurs de la discipline. Ces journées sont également largement ouvertes aux entreprises impliquées dans des projets de recherche, de développement ou d'innovation.

Les précédentes journées se sont tenues à Toulouse (1993), Grenoble (1994), Chambéry (1995), Port-Camargue (1996), Nice (1997), Nancy (1998), L'Île de la Réunion (1999), Saint-Étienne (2000), Montréal (2001), Lille (2002), Hammamet (2003), Paris (2004), Calais (2005), Annecy (2006), Carcassonne (2007), Brest (2008), Lyon (2009), Mahdia (2010), Valenciennes (2011), Honfleur (2012), Lille (PFIA, 2013), Lorient-sur-Drôme (2014), Rennes (PFIA, 2015), Saint Martin du Vivier - Rouen (2016), Caen (PFIA, 2017), Métabief - Besançon (2018), Toulouse (PFIA, 2019), Angers (PFIA virtualisée, 2020) et enfin Bordeaux (PFIA virtualisée, 2021).

Après deux années de virtualisation imposées par la pandémie, cette nouvelle édition des JFSMA va retrouver le chemin de la convivialité en se tenant en présentiel à St Etienne pour fêter ses 30 ans.

Par tradition, chaque édition des JFSMA met en avant une thématique spécifique que les auteurs sont invités à prendre en compte dans leurs contributions. Cette année, le thème des journées est « SMA et Smart Cities ». Les contributions à l'honneur étaient donc celles dans lesquelles des agents autonomes logiciels et matériels, plongés dans des environnements dynamiques, partiellement accessibles et indéterministes, interagissent et coopèrent dans le but d'apporter des solutions pour améliorer la qualité de vie, la transition écologique ainsi que la mobilité et les transports. Cette édition aura permis d'aborder certains problèmes induits par ce cadre applicatif, à savoir les problèmes d'optimisation, d'interopérabilité, de gestion de données massives et hétérogènes, de temporalité, de passage à l'échelle, d'ouverture... mais aussi sur les possibles solutions s'appuyant sur les modèles multi-agents, les simulations multi-agents, la résolution collective de problèmes, ...

Cet ouvrage contient les **versions préliminaires** des contributions sélectionnées et présentées lors de la trentième édition des Journées Francophones sur les Systèmes Multi-Agents (JFSMA) qui s'est tenue à Saint-Etienne, du 27 au 29 juin 2022, dans le cadre de la Plate-forme Intelligence Artificielle (PFIA 2022). Parmi les 24 articles soumis cette année, le comité de programme des JFSMA'22 a sélectionné 9 publications longues, 2 publications courtes, 4 démonstrations et 3 posters, afin de nous proposer un programme riche et varié. Toutes ces contributions sont regroupées dans le présent ouvrage en 6 parties, correspondant aux 6 sessions de la conférence.

Je tiens à remercier les 75 chercheurs francophones qui ont participé au processus de soumission afin de partager leurs travaux et susciter des échanges fructueux. Je remercie également les membres du comité de programme pour la qualité de leur travail de relecture et d'évaluation tout au long du processus de sélection, ainsi que les comités de programmation et d'organisation de la plate-forme IA pour avoir géré l'ensemble des aspects logistiques et techniques. Je remercie enfin les membres du comité consultatif pour la confiance qu'ils m'ont accordée en me confiant la présidence des JFSMA'22. Les implications de chacun ont permis l'édition de cet ouvrage et ont contribué à la réussite de cette édition anniversaire des JFSMA.

30 ans déjà...

Ce présent document contient les versions préliminaires des articles soumis aux JFSMA'22.

Les **versions finales** des articles sont disponibles dans les actes des Journées Francophones sur les Systèmes Multi-Agents édités par Cépaduès : SMA et Smart Cities - Valérie Camps — Trentièmes Journées Francophones sur les Systèmes Multi-Agents, JFSMA 2022, Saint-Étienne, France, 27 - 29 juin, 2022. Cépaduès, ISBN 9782364939448.

Comité de programme

Présidente

- Valérie Camps (IRIT, Université Toulouse III-Paul Sabatier, Toulouse, France).

Comité consultatif

- Jean-Paul Jamont (LCIS, Université Grenoble Alpes, France);
- Fabien Michel (LIRMM, Université de Montpellier, France);
- Gauthier Picard (DTIS - SYD, ONERA, France);
- Nicolas Sabouret (LIMSI, Université Paris Sud, France);
- Olivier Simonin (CITI lab, INSA Lyon, France).

Membres

- Carole Adam (LIG, CNRS, Université Grenoble Alpes, France);
- Emmanuel Adam (LAMIH, Université Polytechnique Hauts-De-France, France);
- Flavien Balbo (LIMOS, École des Mines de Saint-Etienne, France);
- Aurélie Beynier (LIP6, Sorbonne Université, France);
- Olivier Boissier (LIMOS, École des Mines de Saint-Etienne, France);
- Grégory Bonnet (GREYC, Université Caen Normandie, France);
- Jonathan Bonnet (Continental CSV, Toulouse);
- Florent Carlier (CREN, Le Mans Université, France);
- Anne-Cécile Caron (CRIStAL, Université de Lille, France);
- François Charpillat (LORIA, INRIA, Nancy, France);
- Vincent Chevrier (LORIA, Université de Lorraine, France);
- Caroline Chopinaud (Hub France IA, France);
- Rémy Courdier (LIM, Université de la Réunion, France);
- Yves Demazeau (LIG, CNRS, France);
- Julie Dugdale (LIG, Université de Grenoble Alpes, France);
- Stéphane Galland (CIAD, Université de Technologie Belfort-Montbéliard, France);
- Sylvain Giroux (DOMUS, Université de Sherbrooke, Canada);
- Emmanuelle Grislin (LAMIH, INSA HdF, Université Polytechnique Hauts-De-France, France);
- Zahia Guessoum (LIP6 & CREStIC, Université de Reims, France);
- Salima Hassas (LIRIS, Université de Lyon, France);
- Cédric Herpson (LIP6, Sorbonne Université, France);
- Guillaume Hutzler (IBISC, Université Evry-Paris Saclay, France);
- Elsy Kaddoum (IRIT, Université Toulouse Jean Jaurès, France);
- Jean-Daniel Kant (LIP6, Sorbonne Université, France);
- Christophe Lang (FEMTO-ST, Université de Franche-Comté, France);
- Domitile Lourdeaux (UTC, Université de Technologie de Compiègne);
- René Mandiau (LAMIH, Université Polytechnique Hauts-De-France, France);
- Nicolas Marilleau (UMMISCO, IRD, France);
- Philippe Mathieu (CRIStAL, Université de Lille, France);
- Bruno Mermet (GREYC, Université Caen Normandie, France);
- Frédéric Migeon (IRIT, Université Toulouse III-Paul Sabatier, Toulouse, France);
- Maxime Morge (CRIStAL, Université de Lille, France);
- Gildas Morvan (LGI2A, Université d'Artois, France);
- Jean-Pierre Müller (SENS, CIRAD, Université de Montpellier, France);
- Antoine Nongaillard (CRIStAL, Université de Lille, France);
- Michel Occello (LCIS, Université Grenoble Alpes, France);
- Wassila Ouerdane (MICS, École Centrale Supélec, France);
- Alexandre Pauchet (LITIS, INSA Rouen Normandie, France);
- Sébastien Picault (INRAE, UMR BIOEPAR, France);
- Suzanne Pinson (LAMSADE, Université Paris Dauphine, France);
- Éric Platon (CosmosX, Japon);
- Clément Raievsky (LCIS, Université Grenoble Alpes, France);

- Lilia Rejeb (SMART Lab, Institut Supérieur de Gestion, Tunisie);
- Valérie Renault (CREN, Le Mans Université, France);
- Julien Saunier (LITIS, INSA Rouen Normandie, France);
- Cristina Stoica Maniu (L2S, CentraleSupélec, France);
- Patrick Tailliandier (COGIT, IGN, France);
- Paul Théron (Thalès TCS, France);
- Laurent Vercouter (LITIS, INSA Rouen Normandie, France);
- Mahdi Zargayouna (GRETIA, Université Gustave Eiffel, France).

**Session 1 : Résolution collective de problèmes, coordination, travail
en équipe**

Coordination de trajectoires 4D par optimisation distribuée dans la gestion du trafic aérien sans pilote

G. Picard

gauthier.picard@onera.fr

ONERA/DTIS, Université de Toulouse, France

Résumé

Nous proposons une stratégie de déconfliction dans la gestion du trafic aérien sans pilote (UTM), où les drones sont dotés de capacités de décision pour mettre à jour leurs trajectoires 4D face à des événements imprévisibles ou lorsque des trajectoires prioritaires sont ajoutées dans l'espace aérien. Les drones sont conscients des conflits potentiels proches fournis par la couche de communication UTM, et peuvent se coordonner pour réparer les trajectoires et résoudre les conflits de manière décentralisée. Cela réduit l'accès à un goulot d'étranglement décisionnel central et permet une déconfliction réactive. Nous proposons des comportements coordonnés basés sur les DCOP, évalués sur des scénarios denses dans une zone limitée avec de nombreux contrats 4D, des incidents potentiels et des procédures d'urgence.

Mots-clés : Gestion du trafic aérien sans pilote ; Coordination ; DCOP ; Contrats 4D

Abstract

We propose a deconfliction strategy in unmanned air traffic management (UTM), where drones are provided with decision capabilities to update their 4D trajectories when facing unpredictable events or when priority trajectories are added in the airspace. The drones are aware of nearby potential conflicts provided by the UTM communication layer, and can coordinate to repair trajectories and resolve conflicts in a decentralized manner. This reduces access to a central decision bottleneck and enables reactive deconfliction. We propose coordinated behaviors based on DCOPs, evaluated on dense scenarios in a limited area with many 4D contracts, potential incidents and emergency procedures.

Keywords: Unmanned Air Traffic Management ; Coordination ; DCOP ; 4D Contracts

1 Introduction

En 2018, le bureau NextGen de l'administration fédérale américaine de l'aviation (FAA) a publié un premier concept d'opérations (ConOps) global pour la gestion du trafic des systèmes d'aéronefs sans pilote (UTM) qui présente une vision et décrit les exigences opérationnelles et techniques associées pour

1. Nous utiliserons dans cet article, de manière interchangeable les termes *drone* et *UAV* pour désigner des aéronefs sans pilotes.

développer une architecture support et fonctionner dans un écosystème UTM [7]. Il s'agit d'un système communautaire de gestion du trafic dans lequel les opérateurs et les entités fournissant des services de soutien aux opérations sont responsables de la coordination, de l'exécution et de la gestion des opérations, avec des règles de conduite établies par la FAA. Cet ensemble fédéré de services permet une gestion coopérative des opérations entre les exploitants de systèmes d'aéronefs sans pilote (UAS), facilitée par des fournisseurs de services tiers (USSP) grâce à des échanges d'informations en réseau. L'UTM est conçu pour répondre à la demande et aux attentes d'un large éventail d'opérations dont la complexité et le risque ne cessent de croître, grâce à un marché ouvert innovant et concurrentiel de fournisseurs de services. Les services fournis par les USSP sont interopérables pour permettre à l'écosystème UTM de répondre aux besoins de la communauté des opérateurs de drones. Ce document de travail propose également un ensemble de scénarios soulevant de fortes problématiques de coordination, dont la résolution s'avère être de véritables défis scientifiques comme l'illustre [10], et qui permettent également de définir des scénarios pour l'évaluation des solutions candidates.

Du côté européen, le modèle opérationnel envisagé, appelé U-Space, identifie également les besoins de coordination entre les acteurs, et de distribution des décisions [21]. Bien qu'elles présentent des différences notables (classification des drones, sectorisation de l'espace aérien, niveau de sécurité, responsabilités des USSP) comme identifié dans [13], ces ConOps convergent sur de nombreux points (orientation service, modularité, confidentialité, acteurs de haut niveau, séparation en niveaux stratégique, tactique et réactif), notamment en ce qui concerne la coordination des USSP et des drones.

Dans cet article, nous proposons un mécanisme de coordination adaptatif permettant aux systèmes d'UAV (autonomes ou opérés) évoluant dans un espace aérien urbain à route libre de gérer les conflits de trajectoire dus aux retards, aux incidents ou à l'apparition de nouvelles trajectoires hautement prioritaires. Bien qu'il existe des stratégies d'adaptation (e.g. report ou élévation de trajectoires) pour éviter les futurs conflits, nous proposons ici de faire en sorte que les drones se coordonnent en utilisant un service de messagerie directe (soit directement

de véhicule à véhicule, soit en utilisant une communication à l'échelle de la plateforme) pour résoudre les conflits, et pour choisir les meilleures actions par rapport à des objectifs multiples. Le mécanisme de coordination est basé sur l'optimisation distribuée sous contraintes (DCOP), afin d'assurer que les drones choisissent collectivement des actions de déconfliction, tout en minimisant certains indicateurs opérationnels. Le sujet de notre étude étant l'interaction entre les UAVs et les USSPs, les résultats proposés sont suffisamment génériques pour être applicables aux différents concepts d'opérations UTM et U-Space. Notre approche aborde la réparation de contrats 4D en ligne de manière distribuée. A notre connaissance, il n'existe pas d'approche multi-agents pour ce problème, qui est un sujet nouveau faisant suite aux récents concepts d'opération dans les espaces aériens urbains.

2 Scénario illustratif

Utilisons un scénario illustratif pour mettre en évidence les besoins de coordination et d'adaptation dans des contextes UTM, représentés dans la figure 1. Nous réutilisons principalement le scénario proposé dans [7], centré sur un espace aérien à route libre. Nous considérons une zone urbaine de 2 km sur 2 km où plusieurs opérateurs de drones ont demandé des trajectoires 4D pour effectuer différentes tâches, comme la livraison, la surveillance ou la capture d'images au sol. Ces trajectoires ont été demandées à l'UTM, qui en retour envoie les points de passage 4D (espace 3D et temps) que les drones doivent suivre pour accomplir leurs tâches. L'UTM fournit également des marges de tolérance permettant de s'écarter de la trajectoire initiale, représentées par des tubes de sécurité encapsulant les trajectoires. Les trajectoires (et leurs tubes de sécurité respectifs) fournies par l'UTM sont garanties sans conflit spatio-temporel mais doivent être demandées quelques heures avant les opérations. Les trajectoires 4D résultantes sont ainsi contractées (et appelées *contrats 4D*), et les drones s'engagent à respecter ces contrats. Cependant, dans certains cas, les drones peuvent ne pas être en mesure de respecter leur trajectoire. Par exemple, un vent imprévisible peut modifier radicalement les trajectoires, en particulier pour les drones petits et légers [18]. De plus, certaines tâches peuvent exiger qu'un drone reste en position pour exécuter une routine de surveillance (e.g., un drone de surveillance détectant un danger dans sa zone de surveillance doit capturer la scène sous différents angles). Enfin, certaines trajectoires hautement prioritaires peuvent être ajoutées à l'espace aérien pour des raisons d'urgence (e.g., intervention d'un hélicoptère d'évacuation médicale), ce qui peut générer plusieurs conflits avec les trajectoires actuelles. Les contrats 4D ne gèrent pas de telles trajectoires conditionnelles et non déterministes, et doivent donc être mis à jour lorsque de tels événements se produisent.

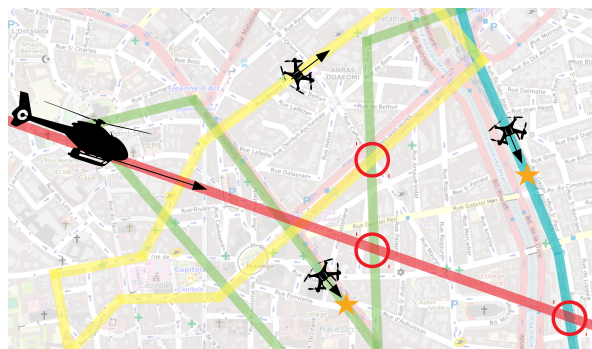


FIGURE 1 – Un scénario type avec trois drones suivant leurs trajectoires (vert, bleu et jaune) gérant quelques incidents (étoiles orange), un hélicoptère médical sur sa trajectoire d'urgence (rouge), et quelques conflits identifiés (cercles rouges)

Dans notre article, nous considérerons des drones ayant des trajectoires de surveillance, et qui peuvent avoir à mettre en œuvre une routine de surveillance d'incident (avec un vol stationnaire), suspendant ainsi leur trajectoire actuelle. L'occurrence de tels incidents n'est pas connue à l'avance, et est supposée totalement aléatoire. Nous considérerons également certains événements d'urgence, dont la destination n'est pas connue à l'avance, qui sont hautement prioritaires et ne peuvent être négociés. Dans la figure 1, trois drones suivent leurs trajectoires (verte, bleue et jaune) et gèrent certains incidents (étoiles orange), qui vont retarder leurs trajectoires planifiées, et donc générer un conflit (cercle rouge au croisement jaune-vert) à une position qui était spatio-temporellement sûre auparavant. Plus conflictuel encore, un hélicoptère médical doit traverser l'espace aérien en urgence, ce qui entraîne deux autres conflits (cercles rouges aux croisements vert-rouge et bleu-rouge).

Nous positionnons notre travail au niveau des UAS où les UAV peuvent échanger directement des informations via la communication directe, ils peuvent recevoir des commandes et des ordres des opérateurs, et obtenir des informations en temps réel (e.g. les trajectoires, les conflits, les procédures d'urgence) de certains USSP. Nous envisageons que les mécanismes de coordination tactique et réactive proposés puissent avoir lieu entre plusieurs UAS (semi-)autonomes ou entre plusieurs opérateurs UAS. Nous nous concentrons notamment sur les petits drones capables d'effectuer un vol stationnaire et opérant à basse altitude (entre 0m et 300m).

3 Concepts et modèle du problème

Nous présentons ici les modèles des trajectoires 4D, des drones et des services UTM nécessaires à la déconfliction.

3.1 Trajectoires 4D

Une trajectoire est un ensemble $W \subset \mathbb{R}^4$ de *points 4D* $w = (x, y, z, t)$ où x et y sont des coordonnées sur le plan 2D (ou des coordonnées GPS), z est l'altitude, et t le temps. Ces points définissent les points de passage que les drones doivent survoler. Nous appelons un *segment* $l = (w_s, w_e)$ une ligne entre deux points consécutifs. Nous ne considérerons que les segments horizontaux et verticaux. Les drones évoluent sur des plans ou se déplacent vers un autre plan en utilisant un segment vertical. Ainsi l'espace aérien est divisé en plusieurs plans séparés par une hauteur constante, notée Z_{sep} . Sur chaque plan, les trajectoires sont libres, c'est-à-dire qu'il n'existe pas de réseau de vol prédéfini. Une trajectoire est également définie par des marges de sécurité, afin d'éviter que les drones n'entrent en collision, laissant le temps et l'espace aux routines d'évitement réactives de subsumer les ordres de navigation. Classiquement, ces tubes de sécurité sont définis horizontalement (pour les dimensions x et y), verticalement (pour z) et temporellement (pour t). Un tube de sécurité est donc défini par $\tau = (h, v, t)$. Un drone diverge lorsqu'il se trouve en dehors du tube de sécurité de son segment de vol actuel. Un drone viole une trajectoire si sa position est à l'intérieur du tube de sécurité d'un autre drone. Formellement, un point w_0 est à l'intérieur d'un tube de sécurité (h, v, t) pour le segment $l = (w_1, w_2)$ s'il satisfait aux trois conditions (1) et (2) et (3) :

$$\frac{|(x_2 - x_1)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_1)|}{d_{xy}(w_1, w_2)} \leq h \quad (1)$$

$$\min(z_1, z_2) - v \leq z_0 \leq \max(z_1, z_2) + v \quad (2)$$

$$|t_0 - (t_1 + \frac{d_{xy}(w_1, p_{xy}(w_0, l))}{d_{xy}(w_1, w_2)} d_t(w_1, w_2))| \leq t \quad (3)$$

avec d_{xy} la distance euclidienne 2D, d_t la distance temporelle 1D, et $p_{xy}(w, l)$ la projection 2D de w sur l . Deux segments sont en *conflit* s'ils sont sur le même plan (soit horizontalement, soit verticalement) et s'ils se coupent au même moment (à la tolérance t près).

La construction d'un ensemble de trajectoires 4D est un problème opérationnel classique qui fait l'objet d'une littérature abondante. Il a été très bien étudié dans le contexte de la gestion du trafic aérien [5]. La construction de trajectoires sans conflit est un problème d'optimisation difficile, souvent résolu à l'aide de métaheuristiques comme le recuit simulé [11] ou les algorithmes évolutionnaires [22]. En présence de petits drones capables de changer de direction et de vitesse de manière plus flexible que les avions classiques, le problème reste difficile, mais d'autres techniques comme le PSO [1] ou même des techniques multi-agents [24] ont été envisagées. Nous étudions ici un espace aérien à route libre *non structuré*, c'est-à-dire qu'il n'y a pas

de réseau de routes aériennes prédéfinies auquel les trajectoires sont contraintes, contrairement aux concepts opérationnels habituels de l'ATM [17]. Dans notre cas, nous considérerons que les trajectoires initiales sont sans conflit, obtenues à partir de n'importe quel générateur de trajectoires (incrémental dans nos expériences).

Nous nous concentrons sur la procédure de réparation et non sur la génération de l'ensemble initial de trajectoires. Nous cherchons à optimiser certains critères liés à la qualité de la réparation. Nous visons à : (i) *minimiser le nombre de conflits* générés par l'adaptation de la trajectoire, afin de garantir la sécurité du vol ; (ii) *minimiser le nombre de points de passage manqués*, afin de garantir la qualité des trajectoires, notamment dans un contexte urbain, où les trajectoires sont définies pour remplir certaines missions entre et à ces points de passage ; (iii) *minimiser le retard global* induit par l'adaptation. Un tel problème n'est pas trivial et peut nécessiter un compromis ; par exemple, sauter un segment conflictuel améliore la sécurité mais réduit la qualité de service.

3.2 UAVs

Notre scénario considère un ensemble U de drones, chacun étant noté $u = (p, s, d, T)$, où $p = (x, y, z, t) \in \mathbb{R}^4$ est sa position, $s = (h, v, a) \leq (h_{max}, v_{max}, a_{max}) \in \mathbb{R}^3$ représente ses vitesses horizontale (en $m.s^{-1}$), verticale (en $m.s^{-1}$) et angulaire (en $rad.s^{-1}$) actuelles. La valeur $d \in [0, 2\pi]$ définit sa direction actuelle, et T est sa trajectoire 4D. Notez que ces valeurs de vitesse peuvent être égales à zéro (par exemple, aucun mouvement). Dans des conditions nominales, les drones modifient leur vitesse et leur direction afin d'atteindre leur prochain point de passage à temps, ou de rentrer dans leur tube de sécurité s'ils ont été forcés d'en sortir (par exemple en raison d'un vent fort ou de l'évitement d'un obstacle). En cas d'incident à proximité de sa position, un drone restera à la même position (il garde un point de passage à sa position actuelle) pour observer la situation pendant un certain temps. Ensuite, il retournera à sa trajectoire nominale. Nous modélisons ce comportement de suivi de cible par un modèle de mouvement simple. Lorsque u est à la position p et qu'il vise à atteindre la position o avec un angle $\angle(p, o) \in [-\Pi, \Pi]$, il appliquera les variations suivantes à ses trois degrés de liberté :

$$\Delta h = \min\left(\frac{d_{xy}(p, o)}{d_t(p, o)}, \Delta h_{max}\right) \quad (4)$$

$$\Delta v = \min\left(\frac{d_z(p, o)}{d_t(p, o)}, \Delta v_{max}\right) \quad (5)$$

$$\Delta a = \text{sgn}(\angle(p, o)) \cdot (\min(\angle(p, o), \Delta a_{max})) \quad (6)$$

conduisant à la variation de position suivante :

$$\Delta p = (h \cdot \cos(d), h \cdot \sin(d), v, a) \quad (7)$$

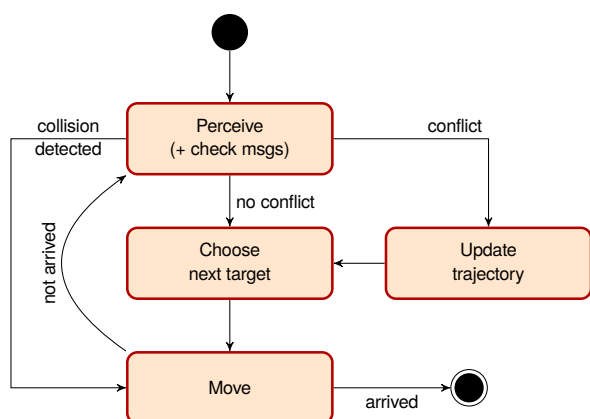


FIGURE 2 – Comportement générique d'un UAV

Le comportement nominal d'un drone représenté sur la figure 2 consiste d'abord à percevoir l'environnement et à vérifier les messages entrants. Ensuite, si aucun conflit, ni obstacle proche n'est détecté, le drone choisit une position cible (soit un point de passage, une localisation d'incident, ou sa position contractée s'il a divergé), puis se déplace en ajustant ses vitesses à l'aide de (4), (5) et (6) pour atteindre une nouvelle position $p + \Delta p$ le rapprochant de sa position cible. Les drones communiquent avec les services de l'UTM (par exemple, géolocalisation, détection de conflits) et avec les drones voisins (directement ou via l'USSP). Ces communications peuvent signaler des conflits. Dans ce cas, le drone déconflictera en mettant éventuellement à jour sa trajectoire. Les drones sont également équipés d'une routine réactive d'évitement des collisions s'ils détectent un mobile trop proche de leur position. Dans ce cas, le drone effectue un comportement de "détection et évitement" que nous ne détaillons pas dans cet article.

3.3 Services UTM

Les opérateurs de drones s'abonnent à des services afin d'obtenir des informations et de fournir des informations sur leurs missions en cours. Des informations en temps réel (sur les positions, les conflits ou les nouvelles trajectoires insérées) sont envoyées par les services aux drones abonnés, tandis que les demandes de mise à jour de trajectoire et les rapports d'incidents sont envoyés par les drones aux USSP. Nous considérons au moins les services suivants :

Un *service de messagerie directe* (DMS), qui permet aux UAV de communiquer de manière P2P. Les UAV utiliseront ce service pour envoyer des messages de coordination (liés à la méthode de résolution DCOP, voir la section 4).

Un *service de détection de conflit* (CDS), qui envoie aux abonnés les conflits devant se produire dans les prochaines minutes, suite à une mise à jour de trajectoire ou à un ajout de trajectoire. Une information de conflit contient les segments qui se croisent,

les drones respectifs et le point d'intersection. On note $C(u)$ l'ensemble des conflits dans lesquels u est impliqué. Afin de rester suffisamment réactif, la détection des conflits est effectuée à un horizon limité : elle ne vérifie que les k prochains segments pour chaque trajectoire, à partir de l'instant courant.

Un *service de mise à jour de trajectoire* (TUS), que les drones sollicitent pour reporter leur trajectoire actuelle ou pour ajouter de nouveaux points de passage (e.g. pour contourner un obstacle en dehors des tubes de sécurité). En retour, le service renvoie la nouvelle trajectoire et les nouveaux conflits (s'il y en a). En effet, si un drone doit reporter sa trajectoire, il peut produire des conflits sur les segments futurs.

Un *service d'évaluation de trajectoire* (TAS) qui est interrogé pour évaluer le remplacement d'une trajectoire par une autre. Ce service renvoie la nouvelle trajectoire et certaines mesures d'évaluation (e.g., le nombre de conflits générés et le retard résultant par rapport à la trajectoire remplacée).

3.4 Actions de déconfliction et comportements

Après le démarrage du système (sans conflit), des événements non prévus (comme des incidents ou des urgences) exigeront éventuellement la rupture des contrats. Puisque les conflits consistent principalement en des intersections sur le même plan, et puisque les drones considérés sont capables d'effectuer un vol stationnaire, trois options principales sont ouvertes pour mettre à jour les contrats : *postpone* (retarder), *elevate* (changer d'altitude) et *skip* (supprimer un point de passage).

Postpone. Le principe de action de déconfliction est simple : reporter tous les points de passage du contrat après une date donnée pour un temps donné, afin de pouvoir éviter un conflit ou d'avoir suffisamment de temps pour observer un incident. Cette action est automatiquement appelée lorsqu'un drone détecte un incident à proximité de sa position. Il demandera alors au service de mise à jour du contrat un report pour un temps dépendant de l'incident. Tous les points de passage après sa position spatio-temporelle actuelle sont alors retardés. Les drones peuvent également demander un report afin d'éviter un futur conflit potentiel identifié par le service de détection des conflits. Dans ce cas, les drones ajouteront un nouveau point de passage avant l'intersection et retarderont leur route d'une durée égale au temps nécessaire pour sortir d'un tube de sécurité. Nous appellerons postpone, les UAVs effectuant uniquement cette action de déconfliction.

Elevate. Une autre approche de la déconfliction avec des drones capables de voler verticalement consiste à faire en sorte que les drones changent de plan (vers le bas ou vers le haut), afin de contourner le lieu du conflit, puis de revenir à leur plan précédent. Cette action est effectuée pour éviter les obstacles ou pour prévenir un futur conflit potentiel.

Lorsqu'il demande une mise à jour de la trajectoire, le drone fournit le conflit ou la position à éviter, et le service de mise à jour de la trajectoire renvoie une trajectoire mise à jour avec un "pont" au-dessus de la position à éviter, c'est-à-dire l'ajout de 4 points de passage pour créer un segment vertical, suivi d'un segment horizontal et enfin d'un autre vertical. Un pont est soit ascendant, soit descendant (en fonction de l'altitude). Nous appellerons *elevate*, les UAV effectuant uniquement cette action de déconflition.

Skip. Enfin, on peut simplement sauter un segment conflictuel, en supprimant son point de départ. Ainsi, le drone contournera le conflit en se déplaçant directement vers le point de passage après le segment conflictuel. Lorsqu'il demande à sauter un segment, le drone fournit simplement les conflits au service de mise à jour de la trajectoire qui supprimera le point de passage correspondant et renverra la nouvelle trajectoire. Nous appellerons *skip*, les UAVs effectuant uniquement cette action de déconflition.

4 Coordination basée sur les DCOP

Cette section définit une coordination coopérative entre les drones, de sorte qu'ils décident conjointement quels drones déclencheront quelle action de déconflition pour un conflit donné. Nous modélisons ce problème de décision collective comme un DCOP.

4.1 Le cadre des DCOP

Un problème d'optimisation avec contraintes distribuées (DCOP) est la variante distribuée de l'optimisation avec contraintes [16]. Ici, un groupe d'agents doit choisir des valeurs pour un ensemble de variables de manière décentralisée afin de minimiser une fonction de coût ou de maximiser une fonction d'utilité. Formellement, un DCOP est un tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu, f \rangle$, où : $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$ est un ensemble d'agents ; $\mathcal{X} = \{x_1, \dots, x_n\}$ sont des 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 des valeurs dans $\mathcal{D}_{x_i} = \{v_1, \dots, v_k\}$; L'élément $\mathcal{C} = \{f_1, \dots, f_m\}$ est un ensemble de contraintes souples, où chaque f_i définit un coût dans $\mathbb{R}^+ \cup \{+\infty\}$ pour chaque combinaison d'affectations à un sous-ensemble de variables (une contrainte est initialement connue des seuls agents concernés) ; $\alpha : \mathcal{X} \rightarrow \mathcal{A}$ fait correspondre 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 complète de variables. L'objectif d'optimisation est représenté par la fonction f , qui, en général, est considérée comme la somme des coûts : $f = \sum_i f_i$. Une *solution* à un DCOP P est une affectation complète de toutes les variables. Une solution est *optimale* si elle minimise f .

Les DCOP ont été largement étudiés et appliqués à de nombreux domaines [8], car : (i) ils se concentrent

sur des approches décentralisées dans lesquelles les agents coordonnent vers une solution commune par le biais d'échanges de messages locaux ; (ii) ils exploitent la structure du domaine (en l'encodant dans des contraintes) pour résoudre des problèmes de calcul difficiles ; (iii) il existe une grande variété de méthodes de résolution allant des méthodes exactes aux techniques heuristiques et approximatives. Nous renvoyons le lecteur à [8] pour une étude sur les DCOP [8]. Les méthodes de résolution utilisent des envois directs de messages qui, dans le contexte de notre UTM, sont effectués via un DMS.

4.2 Résoudre des conflits avec un DCOP

Modélisons maintenant notre problème de coordination en un DCOP qui doit être instancié chaque fois que certains UAV sont conscients d'un conflit. L'idée est double : (i) permettre aux UAV de choisir entre plusieurs actions de déconflition lorsqu'un conflit est détecté ; et (ii) faire en sorte que les UAV impliqués dans le même conflit se coordonnent pour choisir celui qui est le plus apte à effectuer la déconflition.

L'ensemble d'agents $\mathcal{A} \subseteq U$ est l'ensemble des drones qui ont été alertés par le *service de détection des conflits* car impliqués dans certains conflits. Chacun de ces drones u connaît ses conflits $C(u)$ et les autres drones avec lesquels il est en conflit, $U(C(u)) = \{v \in U \mid v \neq u, C(u) \cap C(v) \neq \emptyset\}$. Pour un conflit donné c , un drone est capable d'effectuer certaines actions de déconflition ; par exemple $\text{postpone}(c, 20)$, $\text{elevate}(c, -15)$, $\text{elevate}(c, +15)$ et $\text{skip}(c)$. On note \mathcal{I} cet ensemble d'actions.

Les décisions consistent à choisir l'action de déconflition à déclencher pour chaque conflit connu. Notons $x_{u,c,i} \in \{0,1\}$ la variable de décision indiquant si le drone u décide de résoudre le conflit c en utilisant l'action i . Ainsi, $\mathcal{X} = \{x_{u,c,i} \mid u \in \mathcal{A}, c \in C(u), i \in \mathcal{I}\}$, $\mathcal{D} = \{d_{x_{u,c,i}} = \{0,1\} \mid u \in \mathcal{A}, c \in C(u), i \in \mathcal{I}\}$, et $\alpha : x_{u,c,i} \mapsto u$.

Les contraintes se divisent en deux catégories : les coûts unaires (préférences pour les actions) et les coûts de coordination qui interdisent de résoudre le même conflit plus d'une fois. En ce qui concerne les coûts unaires, nous modélisons les préférences pour les actions générant moins de conflits sans diminuer la qualité du service (c-à-d. augmentant le nombre de points de passage manqués) et sans retarder la mission. Les agents doivent être capables d'évaluer les actions en utilisant le *service d'évaluation de la trajectoire* (TAS). Nous notons $v_{\text{conf}}(i,c)$ (resp. $v_{\text{miss}}(i,c)$ et $v_{\text{delay}}(i,c)$) le nombre de conflits (resp. le nombre de points manqués et le retard) de la trajectoire lors de l'exécution de l'action i pour résoudre le conflit c . Nous agrégeons ces multiples évaluations objectives locales par linéarisation et hiérarchisation,

comme suit, pour tous les $u \in U$, $c \in C(u)$ et $i \in \mathcal{I}$:

$$f_{\text{pref}}(x_{u,c,i}) = \omega^2 \cdot v_{\text{conf}}(i,c) + \omega \cdot v_{\text{miss}}(i,c) + v_{\text{delay}}(i,c) \quad (8)$$

avec ω un nombre suffisamment grand pour donner la préférence à la résolution des conflits par rapport aux points de passage manqués et à la réduction des retards. Ces évaluations agrégées modélisent les objectifs multiples discutés dans la section 3. Concernant les contraintes de coordination, nous devons nous assurer qu'exactement une variable $x_{i,c,u}$ est fixée à 1 pour un même conflit. Ainsi, pour chacun de ces conflits c :

$$\sum_{u \in U} \sum_{i \in \mathcal{I}} x_{i,c,u} = 1 \quad (9)$$

Afin de s'intégrer dans le cadre de la DCOP, cette contrainte dure est codée en une contrainte souple comme suit :

$$f_{\text{exo}}(x_{u_1,c,i_1}, \dots, x_{u_{|U|},c,i_{|U|}}) = \begin{cases} 0, & \text{si } (??) \\ +\infty, & \text{sinon} \end{cases} \quad (10)$$

Par conséquent, \mathcal{C} est l'ensemble des contraintes de (8) et (??). La fonction objectif est la somme de toutes ces contraintes.

Ce DCOP peut être résolu de manière distribuée par les agents concernés, en utilisant n'importe quelle méthode de résolution de la littérature, chaque fois que les agents doivent se coordonner. Dans le cas où un agent détecte un conflit avec une trajectoire non négociable (par exemple, une urgence), il essaiera toujours de résoudre le même problème mono-agent, en décidant quelle action est la meilleure.

5 Evaluation expérimentale

Nous évaluons maintenant la performance des comportements coordonnés et adaptatifs sur des scénarios denses, où plusieurs drones adaptent leurs trajectoires suite à l'occurrence d'incidents et l'ajout de trajectoires d'urgence.

5.1 Cadre expérimental

Nous analyserons les performances des comportements de déconflition étudiés avec un nombre croissant d'UAV (et de trajectoires). L'environnement de simulation est codé en Java et exécuté sur un processeur 20-core Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz, 62GB RAM, Ubuntu 18.04.5 LTS, avec une JVM OpenJDK 11.0.9.1. Nous avons exécuté 30 instances d'ensembles de trajectoires et d'incidents générés aléatoirement pour chaque taille de flotte, et nous traçons les valeurs moyennes, avec un intervalle de confiance de $[0,05 ; 0,95]$. Les algorithmes DCOP utilisés ici sont issus de la bibliothèque FRODO [12]. Les valeurs générées aléatoirement sont choisies uniformément dans les intervalles fournis. Le temps

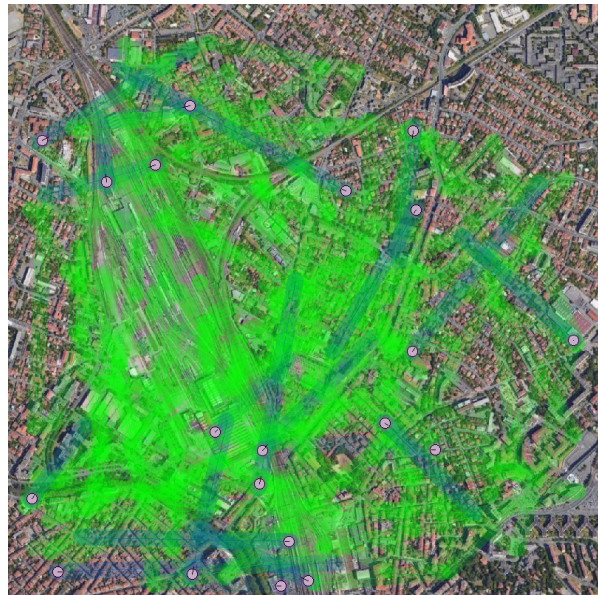


FIGURE 3 – Un exemple de scénario sur une zone urbaine de 2 km x 2 km. 20 UAV sont représentés par des cercles (avec leur vecteur de vitesse). Les contrats 4D (et leurs tubes) sont représentés en vert. Les segments actuels sont représentés en bleu.

de calcul indiqué est le temps de simulation d'un processeur mono-CPU à 20 cœurs.

Nous considérons une zone de 2km par 2km, illustrée dans la figure 3, avec des plans verticaux d'espace aérien à 20m, 40m et 60m. Nous considérons des drones avec les caractéristiques suivantes : $h_{\text{max}} = 18m \cdot s^{-1}$, $v_{\text{max}} = 6m \cdot s^{-1}$, $a_{\text{max}} = \Pi/2 \text{rad} \cdot s^{-1}$, $\Delta h_{\text{max}} = \Delta v_{\text{max}} = 6m \cdot s^{-2}$, $\Delta a_{\text{max}} = \Pi/2 \text{rad} \cdot s^{-2}$. La vitesse initiale est fixée à $(0,0,0)$. Les trajectoires initiales d'un UAV sont générées de manière aléatoire et incrémentale comme suit. Le point de départ est positionné aléatoirement dans l'espace aérien, à l'altitude $z=0$ et au temps $t=0$. Les points suivants sont choisis aléatoirement dans une distance comprise entre 200m et 1000m, de manière à ce qu'il n'y ait pas de conflit avec les segments existants, en augmentant ou diminuant éventuellement l'altitude. Le temps pour le prochain point est également fixé en utilisant une vitesse de croisière par défaut, égale à $(12,3,0)$ (équiv. à 40km/h horizontalement). Les UAVs ont initialement 60 points de passage vers leur destination, puis un dernier segment de retour vers leur origine, ce qui conduit à un ensemble dense de trajectoires et de croisements. Les tubes de sécurité sont définis par $(h,v,t) = (30,15,1)$, ce qui signifie que les UAV doivent être séparés d'au moins 30m horizontalement, 15m verticalement, avec 1s de tolérance temporelle. Pour ajouter des événements imprévisibles, nous générons 3 trajectoires d'urgence, consistant à partir d'une position aléatoire à la frontière de la zone, à se déplacer vers un point aléatoire à au moins 1km de distance du

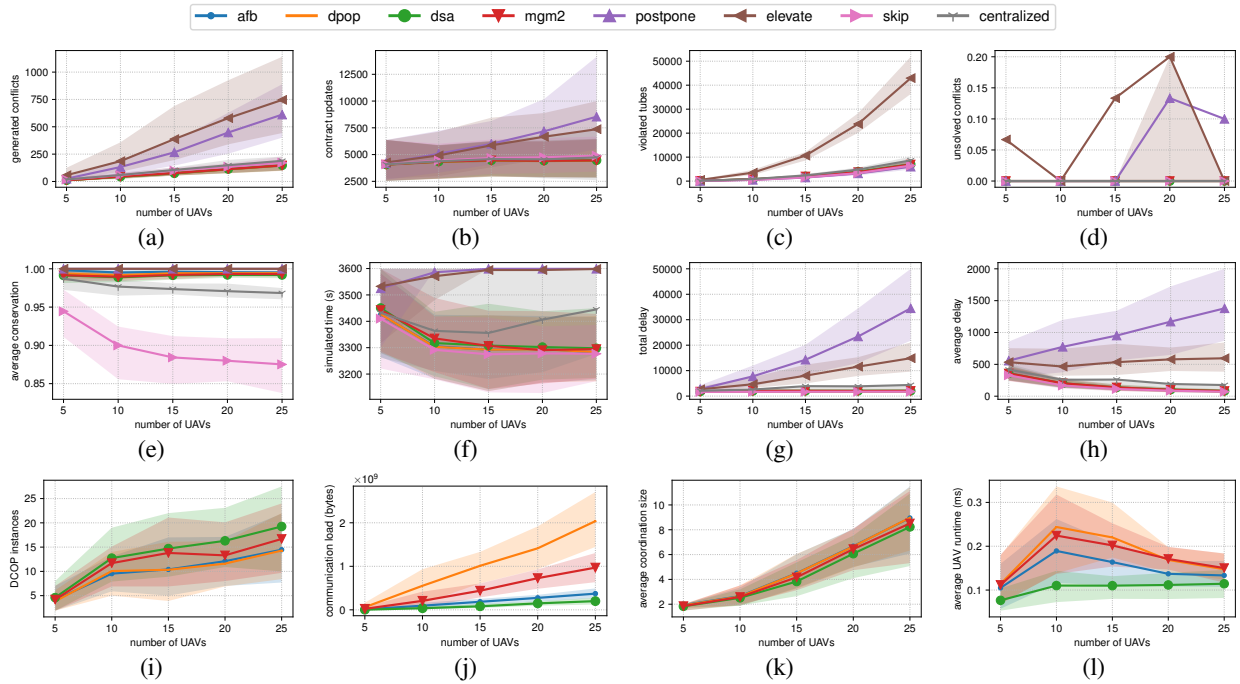


FIGURE 4 – Valeurs moyennes pour plusieurs mesures de performance avec un nombre croissant de drones

point initial, puis un point d'attente pendant 360s (pour simuler une extraction d'urgence), et enfin un retour au point initial. Ces trajectoires d'urgence ne sont pas modifiables. A chaque seconde simulée, il y a également une chance sur 1000 qu'un incident se produise à proximité d'un UAV choisi aléatoirement, devant ainsi capturer la scène pendant une durée aléatoire comprise entre 30s et 120s.

Nous évaluons les comportements suivants : (i) postpone caractérise les drones qui n'effectuent qu'une action postpone($c,20$) lorsqu'un conflit est identifié (abonnement aux services CDS et TUS); (ii) elevate caractérise les drones qui n'effectuent que des actions elevate($c, \pm 15$), selon leur plan actuel, lorsqu'un conflit est identifié (abonnement aux services CDS et TUS); (iii) skip représente les drones qui n'exécutent l'action skip(c) que lorsqu'un conflit est détecté (abonnement aux services CDS et TUS); (iv) afb [9], dpop [19], dsa (variante C) [23], mgm2 [14] représentent des drones qui prennent une décision coordonnée et adaptative à l'aide d'un algorithme DCOP², qui optent entre postpone(c,d) avec $d \in \{20,40,60\}$, elevate($c, \pm 15$) et skip(c) (abonnement aux services DMS, CDS, TAS et TUS); et enfin, (v) centralized est un algorithme de déconflition basé sur une recherche arborescente qui calcule la séquence optimale d'actions de réparation (du même ensemble que les DCOP) pour réparer complètement un ensemble de trajectoires conflictuelles, en optimisant l'objectif dérivé de l'équation (8). Tous

2. Nous n'incluons pas l'algorithme MaxSum [6], car il ne peut pas converger dans la plupart de nos instances, fournissant ainsi des résultats de faible qualité tout en nécessitant de nombreux messages.

les drones sont équipés d'une routine et d'un équipement de prévention des collisions. Le service CDS est configuré avec un horizon $k = 10$, ce qui dans notre cas représente environ 10min dans le futur.

La figure 4 présente les métriques de performance obtenues par simulation, avec un nombre croissant de drones; et la figure 5 montre l'évolution de certaines métriques, avec le temps, pour une instance spécifique avec 25 drones, 3 procédures d'urgence et 25 incidents.

5.2 Conflits et contrats

La première ligne de la figure 4 (4a-4d) porte sur les conflits et les trajectoires. Alors qu'en gestion du trafic aérien classique, avec un espace structuré, le report au départ est utilisé pour réduire les conflits [2], dans notre scénario à route libre, postpone génère de nombreux conflits en raison des retards accumulés sur les croisements de trajectoires, qui ne sont pas structurées comme des réseaux de flux, comme le montre la figure 4a. Tous les autres comportements génèrent encore des conflits (environ 200 sur les grandes instances), puisque près de 70 trajectoires sont ajoutées sur les grandes instances.

Face aux conflits, les UAV mettent à jour leurs contrats, qui ne sont plus garantis sans conflit. Ainsi, les drones ont tendance à demander de nombreuses mises à jour de contrats, comme l'illustre la figure 4b. C'est particulièrement vrai pour postpone et elevate, qui retardent les trajectoires d'une certaine durée, ce qui, la plupart du temps, n'est pas suffisant pour résoudre les conflits, ou qui génère d'autres conflits. Dans les grandes instances, une

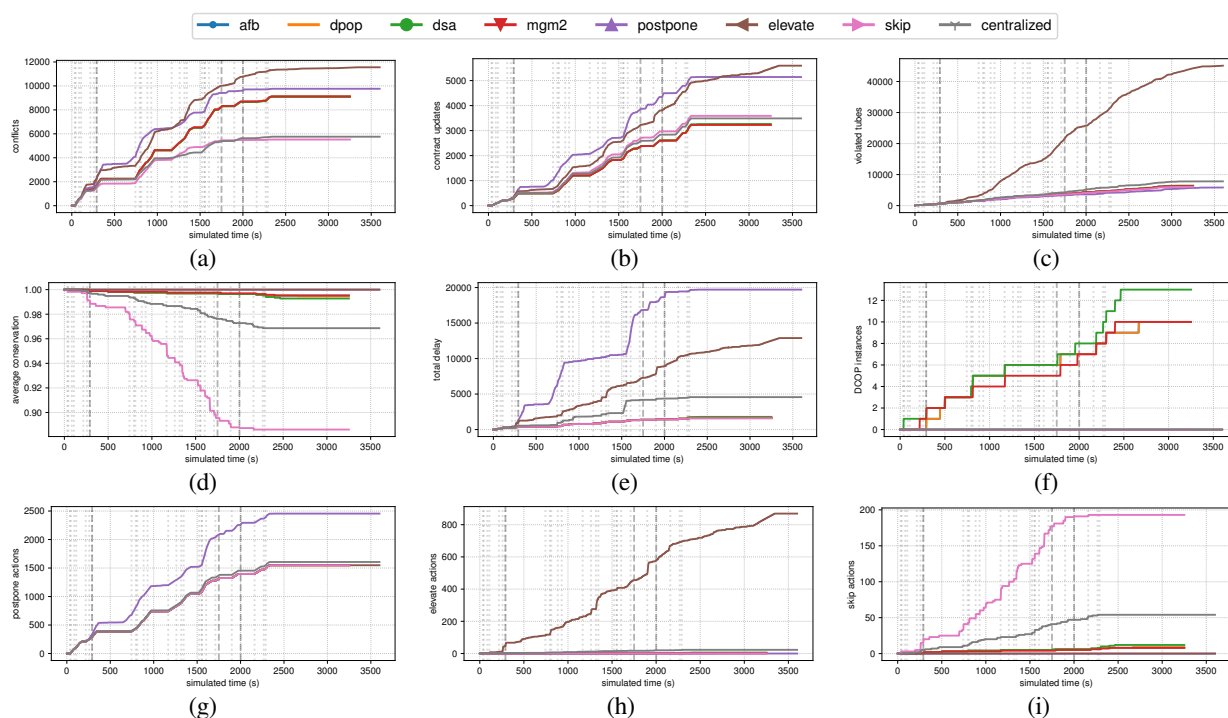


FIGURE 5 – Évolution des mesures de performance au cours d’une simulation avec 25 drones, 3 procédures d’urgence (lignes pointillées grises) et 25 incidents (lignes pointillées grises)

mise à jour de la trajectoire est demandée toutes les secondes, en moyenne. Cette dynamique de mise à jour des trajectoires fait que certains drones violent les tubes de sécurité, et déclenchent les routines anti-collision, comme l’illustrent les figures 4c et 4d. En effet, lorsqu’un UAV demande la mise à jour d’un contrat avec un nouveau segment très proche de sa position actuelle (dans l’espace et le temps) et de la position d’un autre UAV, cela peut conduire à de telles situations, avant qu’un des deux UAV ne soit en mesure de mettre à jour à nouveau son contrat. Notamment, *elevate* fait que les UAVs violent 4 fois plus de tubes de sécurité que les autres comportements. Ces nombres de mises à jour et de violations sont élevés car nous choisissons ici de générer des trajectoires très denses, ce qui représente une sorte de pire scénario, où même les marges de sécurité ne sont pas suffisantes pour éviter les collisions.

En observant notre instance (figure 5), nous pouvons voir à quel point cette configuration est dynamique, ne laissant aucun répit aux drones pour réparer les trajectoires et remplir leurs missions, qui durent environ 1h. Il n’y a qu’une fenêtre "calme" entre $t = 291$ et $t = 744$. Nous pouvons observer comment l’ajout d’incidents déclenche immédiatement des actions de report (que tous les agents mettent en œuvre pour gérer les incidents). En particulier, *postpone* et *elevate* génèrent de nombreux conflits juste après l’ajout d’une première trajectoire d’urgence, comme le montre la figure 5a. Face à ces conflits, les drones tentent constamment de réparer les trajectoires,

jusqu’à ce qu’il n’y ait plus d’incidents et d’urgence dans l’espace aérien, comme l’illustre la figure 5b, après $t = 2290$. Même lorsque les contrats sont stables, les drones peuvent parfois violer certains tubes de sécurité, comme le montre la figure 5c, car ils ont été calculés à l’aide d’une projection temporelle, qui peut ne pas être très précise.

5.3 Conservations des contrats et délais

La deuxième ligne de la figure 4 présente les métriques liées aux deuxième et troisième critères d’optimisation que nous examinons, à savoir la conservation des trajectoires et les retards. Dans la figure 4e, qui indique le pourcentage moyen de points de passage initiaux conservés dans les trajectoires finales, *skip* est le comportement le moins conservateur (jusqu’à 12,5% de points sont abandonnés), car il contourne automatiquement les points de passage en cas de conflit, ce qui permet également de réduire le temps nécessaire pour terminer toutes les trajectoires (indiqué dans la figure 4f). Les deux comportements *elevate* et *postpone* ajoutent du retard pour résoudre les conflits (soit en changeant d’altitude, donc en passant du temps en altitude, soit en reportant les prochains points de passage), comme le montrent les figures 4g et 4h montrant le retard total accumulé et le retard moyen par UAV. Notons que chaque comportement ajoute du retard : cela est dû au fait que les drones traitent de nombreux incidents (jusqu’à 35 dans les plus grandes instances) les obligeant à rester en

position pendant un certain temps, accumulant ainsi du retard. Le solveur centralisé perd en performance sur les grandes instances car il cherche d'abord à minimiser les conflits et les points de passage manqués, et reporte donc plus fréquemment que les solveurs basés sur DCOP. Les approches basées sur le DCOP co-optimisent localement les conflits, la conservation de la trajectoire et les retards, ce qui en fait un compromis efficace.

En examinant notre instance isolée, nous pouvons observer comment *skip* saute constamment des points de passage pour répondre aux conflits, comme le montrent les figures 5d et 5e. *skip* supprime 3 fois plus de points que les comportements basés sur les DCOP. Les adaptations aux incidents ajoutent continuellement des retards aux trajectoires, en raison de l'action de report déclenchée dans de tels cas.

5.4 Mise en œuvre des DCOP

La troisième ligne de la figure 4 présente les métriques de fonctionnement des DCOP. Alors que les trajectoires générées sont denses dans une zone limitée, les procédures DCOP entre plusieurs drones sont assez peu fréquemment déclenchées, comme le montre la figure 4i. Cela s'explique par le fait que les décisions coordonnées basées sur la DCOP parviennent à résoudre les conflits pour l'horizon donné, puis ne nécessitent pas d'autre coordination pendant un certain temps, sauf si d'autres perturbations se produisent. Il est intéressant de noter que les solveurs incomplets (*dsa*, *mgm2*) nécessitent plus de coordination que les solveurs complets (*afb* et *dpop*). En effet, comme les décisions prises par ces solveurs incomplets ne sont pas optimales, elles peuvent donner lieu à des décisions conflictuelles nécessitant une coordination supplémentaire de déconfliction. Pendant la coordination, les conflits concernent en moyenne un tiers de la flotte totale, comme le montre la figure 4k. La coordination peut théoriquement regrouper de nombreux agents, mais dans notre cadre, la taille restreinte des DCOP permet d'utiliser des méthodes de solution complètes comme *afb* et *dpop*, ce qui n'a pas d'impact important sur le temps d'exécution de la décision individuelle du drone, comme le montre la figure 4l. Si l'on considère la charge de communication dans la figure 4j, *dpop* nécessite l'échange de messages peu nombreux mais de grande taille puisque les DCOP à résoudre sont cycliques et nécessitent donc l'échange de messages de taille exponentielle. *mgm2*, en raison de ses paramètres de coordination à 2, échange de nombreux petits messages.

Les comportements basés sur les DCOP sont déclenchés lorsqu'ils font face à certaines des perturbations, comme le montre la figure 5f, et appliquent les actions avec les meilleures évaluations, à savoir *elevate* (pour conserver les points de passage) et *skip* (pour limiter les retards), comme le montrent les figures 5h et 5i. L'action *postpone* n'est pas utilisée

dans ces conditions, comme on peut l'observer dans la période "calme" de la figure 5g. Cela explique comment les DCOP obtiennent de bons résultats en tant que compromis entre la conservation du contrat et les retards. *afb* est le meilleur concurrent, avec une bonne déconfliction nécessitant une faible charge de communication. Cependant, *afb* n'est pas robuste à la perte de messages, contrairement à *dsa*. D'autres expérimentations devraient être menées pour évaluer l'impact du manque de fiabilité des communications sur les performances du DCOP.

5.5 Résumé

Ces résultats soulignent que dans la configuration étudiée extrêmement dynamique, les drones doivent constamment mettre à jour leurs trajectoires. Il est clair que le comportement *postpone* est mauvais sur deux dimensions : les délais et les conflits (donc la sécurité). Le comportement *elevate* est un bon candidat pour la conservation de la trajectoire, mais avec des retards supplémentaires et de nombreuses violations de tubes, tandis que *skip* génère peu de conflits, au prix de quelques points de passage manqués. Le comportement basé sur un DCOP se positionne comme un bon compromis entre *elevate* et *skip*, répondant ainsi à notre objectif de minimiser les conflits générés, de maximiser la conservation de la trajectoire et de minimiser les retards – de la même qualité qu'une déconfliction centralisée. Dans d'autres contextes, avec de nombreux carrefours sur des espaces aériens structurés, ces résultats pourraient ne pas être valables. En effet, une grande partie des performances dépend du fait que les UAV peuvent librement changer d'altitude ou contourner certains points de passage. Dans un espace aérien plus structuré, les UAV changent normalement d'altitude dans certains couloirs verticaux limités, et ne peuvent pas voler en dehors des couloirs horizontaux, comme défini dans [4], par exemple. Dans de tels cas, *postpone* se comporterait probablement mieux, tandis que les DCOP formeraient de plus grands groupes de trajectoires conflictuelles, passant par un nombre limité de carrefours. Nous gardons cette investigation et cette analyse pour des travaux futurs.

6 Conclusion

Dans cet article, nous avons promu et mis en œuvre un mécanisme de coordination multi-agents pour aider les drones évoluant dans un espace aérien dynamique à route libre à adapter leurs trajectoires. La coordination est construite sur un protocole distribué de résolution de contraintes, qui vise à minimiser les conflits futurs, à maximiser la conservation des trajectoires initiales, et à réduire le délai global impliqué par les actions de déconfliction. Sur le scénario simulé que nous avons développé, formant un espace aérien dense avec des événements aléatoires à gérer, nous montrons que l'utilisation d'une déconfliction

coordonnée et adaptative, fournit des solutions avec moins de conflits, de points de passage manqués et de retard accumulé, par rapport aux solutions non coordonnées et fixes. Cependant, cette coordination nécessite l'échange de messages de communication supplémentaires. Dans l'ensemble, l'approche basée sur le DCOP est un candidat prometteur pour l'installation de la déconfliction autonome dans l'UTM, et ouvre la voie à de futures recherches.

Puisque les visions de l'UTM et de l'U-Space évoluent, notamment vers des espaces aériens plus structurés [3, 4], nous envisageons d'étendre notre cadre et nos services à de tels contextes, qui promettent d'être plus difficiles à résoudre en raison du manque de liberté de trajectoire et de la plus grande taille des groupes de déconfliction. Nous souhaitons en particulier adapter la méthode de résolution DCOP aux situations de déconfliction de trajectoire. En fait, nous avons utilisé ici une simple agrégation linéaire de trois critères, qui génère toujours des trajectoires non conflictuelles. Les DCOPs multi-objectifs pourraient être une approche pertinente pour des développements futurs [15]. En outre, la conception d'algorithmes de coordination moins gourmands en communications semble être une condition nécessaire au déploiement de ces protocoles et services sur des dispositifs réels et des UTM. Cela pourrait se faire soit en limitant la taille des DCOP de déconfliction, soit en s'appuyant sur des méthodes de solution légères et robustes comme A-DSA [20], au détriment de l'optimalité.

Références

- [1] D. Alejo, J. A. Cobano, G. Heredia, and A. Ollero. Particle swarm optimization for collision-free 4d trajectory planning in unmanned aerial vehicles. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 298–307, 2013.
- [2] N. Barnier and C. Allignol. Trajectory deconfliction with constraint programming. *The Knowledge Engineering Review*, 27(3) :291–307, 2012.
- [3] A. Bauranov and J. Rakas. Designing airspace for urban air mobility : A review of concepts and approaches. *Progress in Aerospace Sciences*, 125 :100726, 2021.
- [4] C. Capitán, H. Pérez-León, J. Capitán, A. Castaño, and A. Ollero. Unmanned aerial traffic management system architecture for u-space in-flight services. *Applied Sciences*, 11(9), 2021.
- [5] D. Delahaye, S. Puechmorel, P. Tsiotras, and E. Feron. Mathematical models for aircraft trajectory design : A survey. In *Air Traffic Management and Systems*, pages 205–247, Tokyo, 2014. Springer Japan.
- [6] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised Coordination of Low-Power Embedded Devices Using the Max-Sum Algorithm. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages pp. 639–646., 2008.
- [7] Federal Aviation Agency. Utm concept of operations version 2.0, 2020.
- [8] F. Fioretto, E. Pontelli, and W. Yeoh. Distributed constraint optimization problems and applications : A survey. *Journal of Artificial Intelligence Research*, 61 :623–698, 2018.
- [9] A. Gershman, A. Meisels, and R. Zivan. Asynchronous forward-bounding for distributed constraints optimization. In *ECAI 2006*, page 103–107, NLD, 2006. IOS.
- [10] Y. Hamadi. Optimization for urban air mobility. In I. S. Kotsireas and P. M. Pardalos, editors, *Learning and Intelligent Optimization*, pages 1–8, Cham, 2020. Springer International Publishing.
- [11] A. Islami, S. Chaimatanan, and D. Delahaye. *Large-Scale 4D Trajectory Planning*, pages 27–47. Springer Japan, Tokyo, 2017.
- [12] T. Léauté, B. Ottens, and R. Szymanek. FRODO 2.0 : An open-source framework for distributed constraint optimization. In *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*, pages 160–164, Pasadena, California, USA, July 13 2009. <https://frodo-ai.tech>.
- [13] J. Lieb and A. Volkert. Unmanned aircraft systems traffic management : A comparison on the faa utm and the european corus conops based on u-space. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pages 1–6, 2020.
- [14] R. T. Maheswaran, J. P. Pearce, and M. Tambe. Distributed Algorithms for DCOP : A Graphical-Game-Based Approach. In *ISCA PDCS*, pages 432–439, 2004.
- [15] T. Matsui, M. Silaghi, K. Hirayama, M. Yokoo, and H. Matsuo. Distributed search method with bounded cost vectors on multiple objective dcops. In I. Rahwan, W. Wobcke, S. Sen, and T. Sugawara, editors, *PRIMA 2012 : Principles and Practice of Multi-Agent Systems*, pages 137–152. Springer, 2012.
- [16] P. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT : Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence Journal*, 2005.
- [17] C. Nava-Gaxiola, C. Barrado, and P. Royo. Study of a full implementation of free route in the european airspace*. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–6, 2018.
- [18] G. Perozzi, D. Efimov, J.-M. Biannic, and L. Planckaert. Trajectory tracking for a quadrotor under wind perturbations : sliding mode control with state-dependent gains. *Journal of the Franklin Institute*, 355(12) :4809–4838, 2018.
- [19] 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.
- [20] 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.
- [21] SESAR. U-space : Concept of operations. Technical report, SESAR Joint Undertaking, 2019.
- [22] S. Yan and K. Cai. A multi-objective multi-memetic algorithm for network-wide conflict-free 4d flight trajectories planning. *Chinese Journal of Aeronautics*, 30(3) :1161–1173, 2017.
- [23] 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, 2005.
- [24] W. Zhao, Z. Fang, and Z. Yang. Four-dimensional trajectory generation for uavs based on multi-agent q learning. *Journal of Navigation*, 73(4) :874–891, 2020.

Échange de tâches pour la réduction de la durée moyenne de réalisation

Ellie Beauprez

Ellie.Beauprez@univ-lille.fr

Anne-Cécile Caron

Anne-Cecile.caron@univ-lille.fr

Maxime Morge

Maxime.Morge@univ-lille.fr

Jean-Christophe Routier

Jean-Christophe.Routier@univ-lille.fr

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

Résumé

Dans cet article, nous étudions le problème de la réallocation de tâches pour l'équilibrage de charge dans les modèles distribués de traitement de données massives. Nous proposons ici une stratégie qui repose sur des agents coopératifs pour optimiser le réordonnement de tâches dans un ensemble de jobs devant être exécutés le plus tôt possible. Elle permet aux agents de déterminer localement les prochaines tâches à exécuter, à déléguer, voire à échanger grâce à leur modèle des pairs. La nouveauté réside dans la capacité des agents à échanger des tâches à travers des négociations bilatérales concurrentes. Nos expérimentations montrent que la durée moyenne de réalisation atteinte par notre stratégie reste proche de celle obtenue avec une heuristique classique, avec un temps de réordonnement significativement réduit.

Mots-clés : Résolution collective de problèmes, Négociation multi-agents

Abstract

In this paper, we study the problem of task reallocation for load-balancing in distributed data processing models that tackle vast amount of data. We propose here a strategy based on cooperative agents used to optimize the rescheduling of tasks in multiple jobs which must be executed as soon as possible. It allows agents to determine locally the next tasks to process, to delegate, eventually to swap according to their peer modelling. The novelty lies in the ability of agents to swap tasks thanks to concurrent bilateral negotiations. Our experimentation reveals that our strategy reaches a flowtime which is close to the one reached by the classical heuristic approach, and significantly reduces the rescheduling time.

Keywords: Distributed Problem Solving, Agent-based Negotiation

1 Introduction

La problématique de l'affectation efficace de tâches parmi des entités exécutantes est commune à de nombreuses applications réelles pour la logistique [10], la robotique collective [8, 14], le calcul distribué [13], ou comme ici le traitement de données massives [1]. Cet article traite d'une classe d'applications pratiques où : (a) des jobs (c.-à-d. des ensembles de tâches) concurrents doivent être exécutés le plus tôt possible, et (b) les ressources (c.-à-d. les données) requises sont distribuées. Nous considérons particulièrement le modèle de traitement le plus répandu pour traiter des données massives sur une grappe de serveurs, c.-à-d. le patron de conception MapReduce [16]. Les jobs y sont composés d'un ensemble de tâches exécutées par les différents nœuds où sont réparties les ressources. Comme plusieurs ressources sont requises pour réaliser une tâche sur un nœud, son exécution nécessite de récupérer des ressources disponibles sur d'autres nœuds, ce qui induit un surcoût.

De nombreux travaux adoptent le paradigme multi-agents pour aborder le problème de la réallocation de tâches et de l'équilibrage de charge dans les systèmes distribués [2]. L'approche centrée individus permet la distribution d'heuristiques pour des problèmes impraticables à cause de la combinatoire des ordonnancements pour permettre le passage à l'échelle. De plus, intrinsèquement réactives, les méthodes multi-agents de réaffectation s'adaptent aux estimations inexactes des temps d'exécution et aux perturbations (consommation/libération de tâches, ralentissement des exécutants, etc.) La plupart de ces travaux adoptent l'approche orientée marché [15, 8], voire en s'appuyant sur des méthodes d'apprentissage qui nécessitent un historique [14, 13]. Par contraste, nous supposons comme [1] que : (a) les agents sont coopératifs, c.-à-d. ils ont une perception locale et partielle de l'allocation, mais ils partagent le même ob-

jectif, et (b) aucun modèle préalable n'existe, ni des données, ni de l'environnement, de par la classe d'applications pratiques visées. Nous allons ici au-delà en considérant plusieurs jobs composés d'ensembles de tâches, chacune pouvant être exécutée par un seul des agents, tous compétents. Les agents souhaitent minimiser la durée moyenne de réalisation des jobs, c.-à-d. le *flowtime* moyen. La principale difficulté réside dans la formulation de systèmes complexes d'assignation tâches-exécutants décentralisés et adaptatifs, c.-à-d. la conception des comportements individuels joués de manière asynchrone par les exécutants qui doivent aboutir à l'émergence d'affectations faisables qui combinent les objectifs des commanditaires.

Nous proposons une stratégie qui décide quelle réallocation bilatérale est suggérée ou acceptée. Elle repose sur un modèle des pairs et détermine le comportement de l'agent à chaque point de choix dans le protocole de négociation. Dans le prolongement de [3], nos contributions sont les suivantes :

1. La notion de délégation a été généralisée sous la forme de réallocation bilatérale ce qui permet d'envisager des échanges ;
2. le critère de rationalité des réallocations a été redéfini ;
3. la stratégie de contre-offre permet à un agent de compléter une proposition de délégation par un échange de tâche.

Après un aperçu des travaux connexes dans la section 2, nous formalisons le problème d'allocation multi-agents de jobs composés de tâches (cf. section 3). La section 4 décrit les opérations de consommation/réallocation et le processus de négociation. La section 5 précise comment les agents choisissent quelles tâches négocier et avec qui. Notre évaluation empirique est décrite dans la section 6. La section 7 résume notre contribution et présente nos perspectives.

2 Travaux connexes

La théorie de l'ordonnancement [7] propose des méthodes hors-ligne pour résoudre différents problèmes d'affectation efficace de tâches parmi des entités exécutantes. Par exemple, le problème d'ordonnancement qui consiste à minimiser la durée totale de réalisation, notée $C(\vec{A})$, avec m exécutants multi-tâches et n tâches mono-exécutant dont les coûts dépendent de l'entité exécutante (noté R_m) peut être formalisé par programmation linéaire — *Linear Programming* (LP). Ce problème se réduit à un problème d'appariement pondéré dans un graphe biparti avec

n tâches et $n \times m$ positions. Ce problème est polynomial [9]. Reposant sur l'algorithme de Ford-Fulkerson, la complexité de l'algorithme décrit par [6] est $O(\max(mn^2, n^3))$. Cette approche n'est pas toujours adaptée à la réallocation de tâches dans des systèmes distribués où décentralisation et adaptativité sont nécessaires. En effet, un contrôle global constitue un goulot d'étranglement en matière de performance, car il doit en permanence collecter des informations sur l'état du système. À l'opposé, nos agents prennent des décisions locales sur une allocation existante dans le but d'améliorer l'équilibrage de charges. De plus, les problèmes d'ordonnancement classiques sont statiques. L'estimation inexacte du temps d'exécution des tâches, aggravée par des perturbations (consommation de tâches, libération de jobs, ralentissement des nœuds, etc.) peut nécessiter d'importantes modifications de l'allocation existante pour qu'elle reste optimale.

Le paradigme multi-agents est particulièrement approprié pour la conception et l'implémentation de mécanismes distribués et adaptatifs de réaffectation de tâches-exécutants [2]. Les modèles existants se distinguent de par la nature des tâches et des agents, qu'ils représentent les exécutants ou les commanditaires des tâches. S'inspirant de théories économiques, la « programmation orientée marché » aborde les problèmes de planification distribuée à travers la recherche d'un équilibre pour un jeu non-coopératif [15]. Les agents délèguent des tâches voire les échangent. Parmi les méthodes orientées marché, on distingue trois familles.

DCOP. Les problèmes de réaffectation peuvent être représentés sous la forme d'un problème d'optimisation sous contraintes distribué — *Distributed Constraint Optimization Problems* (DCOP). La principale difficulté dans la mise en œuvre de ces méthodes pour la réaffectation de tâches réside dans la représentation d'un problème réaliste sous la forme d'un DCOP, car elle nécessite une expertise de la méthode de résolution (e.g. [10]).

CBBA. L'algorithme à base de consensus (CBBA — *Consensus Based Bundle Algorithm*) [8] est une méthode multi-agents d'affectation en deux phases qui consiste à : (a) sélectionner les tâches à négocier ; (b) déterminer l'agent qui remporte ces négociations. Dans la continuité, Turner et al. étudient l'affectation en continu de tâches à une flotte de robots pour maximiser le débit de réalisation avec des ressources en carburant limitées [14]. Grâce à l'ap-

prentissage automatique supervisé à partir des exécutions précédentes, les robots choisissent dynamiquement et de manière décentralisée la meilleure heuristique de sélection de tâche.

MARL. Les problèmes de réaffectation peuvent également être modélisés via des processus de décision markoviens [5]. Les méthodes de résolution approchée ne peuvent être appliquées que sur de très petites instances de problèmes. L'apprentissage multi-agents par renforcement (MARL-*Multi-Agent Reinforcement Learning*) nécessite une connaissance parfaite de l'environnement et requiert une phase d'apprentissage [13].

À l'inverse, nous ne considérons aucun modèle préalable, ni des données, ni de l'environnement, car cela n'est pas pertinent pour la classe d'applications pratiques qui nous concerne. Par exemple, Baert et al. visent dans [1] un objectif égalitaire qui est la minimisation du temps nécessaire à la réalisation de l'ensemble des tâches (noté $C_{max}(\vec{A})$). Nous considérons ici le problème de la coordination des décisions entre agents pour trouver une solution globalement optimale pour des fonctions multi-objectifs. Les agents tentent de minimiser la durée moyenne de réalisation de plusieurs jobs concurrents, chacun constitué de plusieurs tâches.

Cet article est dans le prolongement de [3].

1. Nous généralisons notre cadre formel pour considérer n'importe quelle réallocation bilatérale, e.g. des échanges pour réduire la durée moyenne de réalisation.

2. Nous redéfinissons le critère de rationalité des réallocations. Précédemment défini à partir du *flowtime* local, c'est-à-dire la durée de réalisation pour les agents impliqués dans la réallocation et du *makespan*, i.e. le temps nécessaire à la réalisation de l'ensemble des jobs, dans cet article, une réallocation est rationnelle si elle permet de réduire le *flowtime* global du système. Ce critère s'avère suffisant pour garantir la convergence du processus de réallocation. Par souci de concision, nous ne présenterons pas les expérimentations qui montrant que passer d'un *flowtime* local au *flowtime* local réduit non seulement la durée moyenne de réalisation mais également le temps de réordonnement.

3 Tâches situées

Nous formalisons ici le problème d'allocation multi-agents des jobs concurrents composés de tâches situées.

Un job est un ensemble de tâches indépendantes, non divisibles et non-préemptives. L'exécution de chaque tâche nécessite l'accès à des ressources distribuées sur les nœuds du système. Nous considérons les ressources transférables et non consommables.

Définition 1 (Système distribué). *Un système distribué est un triplet $\mathcal{D} = \langle \mathcal{N}, \mathcal{E}, \mathcal{R} \rangle$ où :*

- $\mathcal{N} = \{v_1, \dots, v_m\}$ est un ensemble de nœuds ;
- \mathcal{E} est une relation d'accointances, i.e. une relation binaire et symétrique sur \mathcal{N} ;
- $\mathcal{R} = \{\rho_1, \dots, \rho_k\}$ est un ensemble de ressources de tailles $|\rho_i|$. La localisation des ressources, éventuellement répliquées, est déterminée par la fonction $l : \mathcal{R} \rightarrow 2^{\mathcal{N}}$

Pour plus de simplicité, nous faisons l'hypothèse ici qu'il y a exactement un agent par nœud et que toutes les ressources sont accessibles pour tous les agents, même celles sur les nœuds distants, car la relation d'accointances est totale.

L'exécution d'un job (sans date butoir) consiste à exécuter un ensemble de tâches indépendantes.

Définition 2 (Job/Tâche). *Soient \mathcal{D} un système distribué et Res l'espace des résultats. On considère un ensemble de ℓ jobs $\mathcal{J} = \{J_1, \dots, J_\ell\}$. Chaque job J_i est un ensemble de k_i tâches $J_i = \{\tau_1, \dots, \tau_{k_i}\}$ où chaque tâche τ est une fonction $\tau : 2^{\mathcal{R}} \mapsto Res$.*

On note $\mathcal{T} = \cup_{1 \leq i \leq \ell} J_i$ l'ensemble des n tâches sous-jacentes à \mathcal{J} et $\mathcal{R}_\tau \subseteq \mathcal{R}$ l'ensemble des ressources requises pour la tâche τ . Par souci de concision, on note $job(\tau)$ le job contenant la tâche τ . Nous faisons l'hypothèse que le nombre de jobs est négligeable par rapport au nombre de tâches, $|\mathcal{J}| \ll |\mathcal{T}|$.

Le coût d'une tâche est une estimation de son temps d'exécution par un nœud.

Propriété 1 (Coût). *Soient \mathcal{D} un système distribué et \mathcal{T} un ensemble de tâches. La fonction de coût $c : \mathcal{T} \times \mathcal{N} \mapsto \mathbb{R}_+^*$ est telle que :*

$$c(\tau, v_i) \leq c(\tau, v_j) \Leftrightarrow \sum_{\rho \in \mathcal{R}_\tau, v_i \in l(\rho)} |\rho| > \sum_{\rho \in \mathcal{R}_\tau, v_j \in l(\rho)} |\rho| \quad (1)$$

Comme la collecte de ressources distantes représente un surcoût, une tâche est moins coûteuse si les ressources nécessaires sont « plus locales » (cf. section 6). La fonction de coût peut être étendue à un ensemble de tâches :

$$\forall T \subseteq \mathcal{T}, c(T, v_i) = \sum_{\tau \in T} c(\tau, v_i) \quad (2)$$

En substance, nous considérons le problème d'allocation multi-agents de jobs composés de tâches situées.

Définition 3 (MASTA+). *Un problème d'allocation multi-agents de jobs est un quadruplet $MASTA+ = \langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ où :*

- \mathcal{D} est un système distribué de m nœuds ;
- $\mathcal{T} = \{\tau_1, \dots, \tau_n\}$ est un ensemble de n tâches ;
- $\mathcal{J} = \{J_1, \dots, J_\ell\}$ est un partitionnement des tâches en ℓ jobs ;
- $c : \mathcal{T} \times \mathcal{N} \mapsto \mathbb{R}_+^*$ est la fonction de coût.

Une allocation de tâches est une répartition des tâches dans des lots ordonnés.

Définition 4 (Allocation). *Une allocation pour un problème MASTA+ est un vecteur de m lots de tâches ordonnées $\vec{A} = ((B_1, \prec_1), \dots, (B_m, \prec_m))$ où chaque lot (B_i, \prec_i) est l'ensemble des tâches $(B_i \subseteq \mathcal{T})$ affectées au nœud v_i associé à un ordre total strict $(\prec_i \subseteq \mathcal{T} \times \mathcal{T})$. $\tau_j \prec_i \tau_k$ signifie que si $\tau_j, \tau_k \in B_i$ alors τ_j est exécutée avant τ_k par v_i . L'allocation \vec{A} vérifie :*

$$\forall \tau \in \mathcal{T}, \exists v_i \in \mathcal{N}, \tau \in B_i \quad (3)$$

$$\forall v_i \in \mathcal{N}, \forall v_j \in \mathcal{N} \setminus \{v_i\}, B_i \cap B_j = \emptyset \quad (4)$$

Toutes les tâches sont allouées (Eq. 3) et chacune n'est allouée qu'à un seul nœud (Eq. 4). Par souci de concision, on note :

- $\vec{B}_i = (B_i, \prec_i)$, le lot trié de v_i ;
- $\min_{\prec_i} B_i$, la prochaine tâche exécutée par v_i ;
- $\text{jobs}(B_i)$, l'ensemble des jobs affectés à v_i , i.e. les jobs ayant au moins une tâche dans B_i ;
- $v(\tau, \vec{A})$, le nœud chargé de τ dans \vec{A} ;
- $w_i(\vec{A}) = \sum_{\tau \in B_i} c(\tau, v_i)$, la charge de travail du nœud v_i pour l'allocation \vec{A} .

Comme on suppose que les nœuds sont toujours actifs, la durée de réalisation d'une tâche (*completion time*) correspond au temps d'attente avant que la tâche soit entamée plus l'estimation de son temps d'exécution :

$$C_\tau(\vec{A}) = t(\tau, v(\tau, \vec{A})) + c(\tau, v(\tau, \vec{A})) \quad (5)$$

$$\text{avec } t(\tau, v_i) = \sum_{\tau' \in B_i | \tau' \prec_i \tau} c(\tau', v_i)$$

Contrairement aux coûts, les durées de réalisation dépendent de l'ordre d'exécution.

Pour évaluer la qualité d'une allocation de tâches, nous considérons le *flowtime* moyen, qui mesure le temps moyen écoulé entre la date de

libération des jobs et leur date d'achèvement, et le *makespan* qui est le temps nécessaire à la réalisation de l'ensemble des jobs.

Définition 5 (Flowtime/Makespan). *Soient MASTA+ un problème d'allocation de tâches et \vec{A} une allocation. On définit :*

- la durée de réalisation de $J \in \mathcal{J}$ pour \vec{A} ,

$$C_J(\vec{A}) = \max_{\tau \in J} \{C_\tau(\vec{A})\} \quad (6)$$

- le *flowtime* (moyen) de \mathcal{J} pour \vec{A} ,

$$C_{\text{mean}}(\vec{A}) = \frac{1}{\ell} C(\vec{A}) \text{ avec } C(\vec{A}) = \sum_{J \in \mathcal{J}} C_J(\vec{A}) \quad (7)$$

- le *makespan* de \mathcal{J} pour \vec{A} ,

$$C_{\text{max}}(\vec{A}) = \max_{v_i \in \mathcal{N}} \{w_i(\vec{A})\} \quad (8)$$

Contrairement au *makespan*, le *flowtime* dépend de l'ordre d'exécution des tâches sur chacun des nœuds.

Exemple 1 (MASTA+). *À partir du système distribué $\mathcal{D} = \langle \mathcal{N}, \mathcal{E}, \mathcal{R} \rangle$ avec $\mathcal{N} = \{v_1, v_2, v_3\}$, $\mathcal{E} = \{(v_1, v_2), (v_1, v_3), (v_2, v_3)\}$ et $\mathcal{R} = \{\rho_1, \dots, \rho_9\}$ où les ressources sont répliquées sur 2 nœuds (cf. Fig. 1a), nous considérons MASTA+ = $\langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ avec $\mathcal{T} = \{\tau_1, \dots, \tau_9\}$ où chaque tâche τ_i nécessite la ressource ρ_i , $\mathcal{J} = \{J_1, J_2, J_3\}$ tel que $J_1 = \{\tau_1, \tau_2, \tau_3\}$, $J_2 = \{\tau_4, \tau_5, \tau_6\}$ et $J_3 = \{\tau_7, \tau_8, \tau_9\}$ et la fonction de coût donnée dans la table 1. Nous supposons que le coût d'une tâche est proportionnel à la taille des ressources et qu'il est deux fois plus important si la ressource est distante. Nous considérons ici l'allocation \vec{A} (cf. figure 1b) avec $\vec{B}_1 = (\tau_5, \tau_8, \tau_3, \tau_2)$, $\vec{B}_2 = (\tau_4, \tau_9)$ et $\vec{B}_3 = (\tau_7, \tau_1, \tau_6)$. Le *makespan* et le *flowtime* sont $C_{\text{max}}(\vec{A}) = 12$ et $C(\vec{A}) = 8 + 12 + 12 = 32$.*

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9
$c(\tau, v_1)$	5	3	1	8	2	10	4	2	4
$c(\tau, v_2)$	10	3	2	4	2	5	2	2	8
$c(\tau, v_3)$	5	6	1	4	4	5	2	4	4

TABLE 1 – Le coût des tâches pour chaque nœud

En résumé, le coût des tâches dépend du nœud qui l'exécute en raison de la localité des ressources. Notre objectif est de minimiser le *flowtime* moyen des jobs composés de tâches.

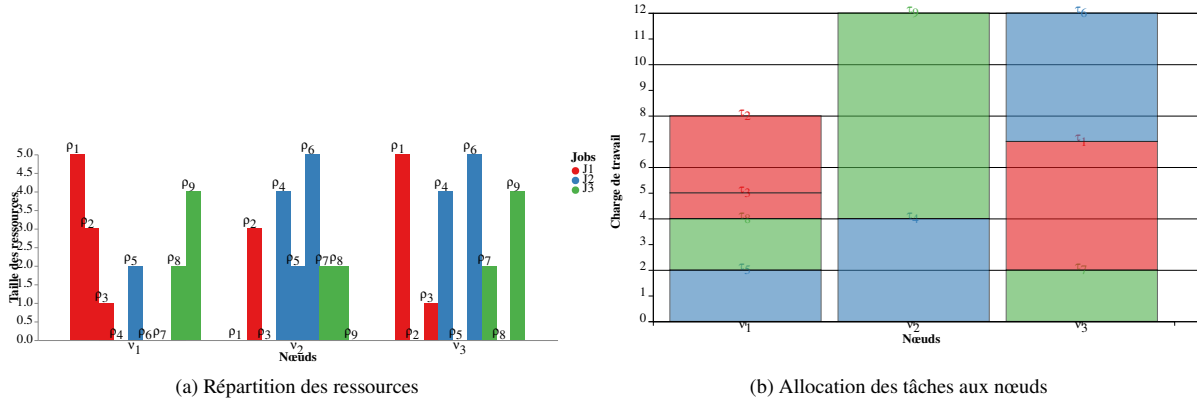


FIGURE 1 – Distributions des ressources et des tâches pour notre exemple fil rouge

4 Consommation et réallocation

Nous décrivons ici les opérations de consommation et de réallocation ainsi que le protocole de négociation.

Une **consommation** de tâche consiste pour un nœud à supprimer une tâche de son lot pour l'exécuter. La stratégie de consommation d'un agent spécifie l'ordonnancement des tâches pour le nœud dont il a la charge. Comme nous voulons minimiser le *flowtime* moyen des jobs, nous considérons ici une stratégie orientée job qui trie le lot d'abord par job puis par tâche au sein d'un même job (les tâches d'un même job sont consécutives dans le lot). En particulier, les jobs les moins coûteux sont prioritaires sur ceux plus coûteux pour minimiser localement le délai de réalisation des jobs. Par la suite, $J_1 \triangleleft_i J_2$ signifie que les tâches de J_1 sont prioritaires sur celles de J_2 et $\tau_1 \triangleleft_i \tau_2$ que la tâche τ_1 est prioritaire sur la tâche τ_2 . Plus formellement :

$$\forall \tau_j, \tau_k \in B_i \quad \tau_j \triangleleft_i \tau_k \Leftrightarrow \text{job}(\tau_j) \triangleleft_i \text{job}(\tau_k) \quad (9)$$

$$\vee (\text{job}(\tau_j) = \text{job}(\tau_k) \wedge \tau_j \triangleleft_i \tau_k)$$

L'ajout ou la suppression d'une liste de tâches T dans le lot \vec{B}_i du nœud v_i peuvent changer l'ordre d'exécution des tâches puisque ces opérations impliquent un réordonnancement du lot :

- $\vec{B}_i \oplus T$ désigne le lot qui contient l'ensemble des tâches $B_i \cup T$ trié selon \triangleleft_i ;
- $\vec{B}_i \ominus T$ désigne le lot qui contient $B_i \setminus T$ trié selon \triangleleft_i .
- $\vec{B}_i \ominus T_1 \oplus T_2$ désigne le lot qui contient $B_i \setminus T_1 \cup T_2$ trié selon \triangleleft_i .

Une **réallocation bilatérale** est une opération qui modifie l'allocation courante via l'échange de tâches entre deux agents.

Définition 6 (Réallocation bilatérale). Soit $\vec{A} = (\vec{B}_1, \dots, \vec{B}_m)$ une allocation pour le problème $MASTA^+ = \langle \mathcal{D}, \mathcal{J}, \mathcal{J}, c \rangle$. La réallocation bilatérale de la liste non vide de tâches T_1 allouées au proposant v_i en échange de la liste de tâches T_2 allouées au répondant v_j dans \vec{A} ($T_1 \subseteq B_i$ et $T_2 \subseteq B_j$) aboutit à l'allocation $\gamma(T_1, T_2, v_i, v_j, \vec{A})$ avec les m lots $\gamma(T_1, T_2, v_i, v_j, \vec{B}_k)$ définis tels que :

$$\gamma(T_1, T_2, v_i, v_j, \vec{B}_k) = \begin{cases} \vec{B}_i \ominus T_1 \oplus T_2 & \text{si } k = i, \\ \vec{B}_j \ominus T_2 \oplus T_1 & \text{si } k = j, \\ \vec{B}_k & \text{sinon} \end{cases} \quad (10)$$

Pour une réallocation bilatérale $\gamma(T_1, T_2, v_i, v_j, \vec{A})$, on distingue deux cas :
 — un **échange** où les deux listes de tâches sont non vides ($T_1 \neq \emptyset \wedge T_2 \neq \emptyset$), noté $\sigma(T_1, T_2, v_i, v_j, \vec{A})$. Si $|T_1| = |T_2| = 1$, on parle d'échange unaire ;

— une **délégation** où un agent donne une partie de ses tâches à l'un de ses pairs sans contre-partie ($T_2 = \emptyset$), notée $\delta(T_1, v_i, v_j, \vec{A})$.

Afin d'améliorer une allocation, nous introduisons la notion de réallocation bilatérale socialement rationnelle.

Définition 7 (Réallocation bilatérale socialement rationnelle). Soit \vec{A} une allocation pour le problème $MASTA^+ = \langle \mathcal{D}, \mathcal{J}, \mathcal{J}, c \rangle$. La réallocation bilatérale $\gamma(T_1, T_2, v_i, v_j, \vec{A})$ est socialement rationnelle ssi le *flowtime* global décroît,

$$C(\gamma(T_1, T_2, v_i, v_j, \vec{A})) < C(\vec{A}) \quad (11)$$

Une allocation est dite **stable** s'il n'existe aucune réallocation bilatérale socialement rationnelle.

Contrairement à [3], nous ne considérons pas comme socialement rationnelles des réallocations qui réduisent le *flowtime* local (le délai de réalisation des jobs pour les seuls nœuds impliqués dans la réallocation) qui ne permet pas de garantir la convergence du processus de réallocation, ni même comme rationnelles des réallocations qui réduisent le *flowtime* local et le *makespan* (la charge maximale des agents). La réduction du *flowtime* global permet de garantir la terminaison du processus. Dorénavant lorsque nous parlerons de *flowtime* il s'agira, sauf précision, du *flowtime* global, noté $C(\vec{A})$ (cf. Eq. 7).

Propriété 2 (Terminaison). *Soit \vec{A} une allocation pour un problème MASTA+ qui n'est pas stable. Toute séquence de réallocations bilatérales socialement rationnelles issue de \vec{A} est finie car elle atteint une allocation stable.*

Esquisse de preuve 1. *Comme il existe un nombre fini d'allocations et que $C(\vec{A})$ décroît strictement à chaque étape, il ne peut y avoir qu'un nombre fini de telles réallocations.*

Pour réaliser des réallocations de tâches, les agents réalisent de multiples négociations bilatérales à un tour. Chaque négociation, reposant sur un protocole d'offres alternées [12], inclut trois étapes de décision : (a) la stratégie d'offre du proposant qui sélectionne une délégation, (b) la stratégie de contre-offre qui permet au répondant de déterminer s'il décline la délégation, l'accepte, voire fait une contre-offre et (c) l'éventuelle réallocation est confirmée ou annulée par le proposant selon les consommations qui ont eu lieu de manière concurrente.

Exemple 2 (Consommation et réallocation). *Considérons le problème MASTA+ de l'exemple 1, en particulier l'allocation \vec{A} représentée dans la figure 1b. Selon la stratégie de consommation adoptée par les agents, chaque lot est trié par job, c.-à-d. les jobs les moins coûteux sont prioritaires (e.g. $J_3 \blacktriangleleft_3 J_1 \blacktriangleleft_3 J_2$). L'ordre naturel sur les identifiants permet de défaire les égalités (e.g. $J_2 \blacktriangleleft_1 J_3$). Les tâches au sein d'un même job sont triées par ordre de coût croissant ($\tau_3 \triangleleft_1 \tau_2$). La délégation de la tâche τ_9 par le nœud v_2 au nœud v_1 aboutissant à l'allocation $\vec{A}' = \delta([\tau_9], v_2, v_1, \vec{A})$ (cf. figure 2a) est socialement rationnelle car elle fait décroître le *flowtime* de 32 à 31. Toutefois, l'échange de $\tau_9 \in B_2$ contre $\tau_5 \in B_1$ entre les nœuds v_2 et v_1 , qui aboutit à l'allocation $\vec{A}'' = \sigma([\tau_9], [\tau_5], v_2, v_1, \vec{A})$ (cf. figure 2b), est*

*meilleur car il fait décroître le *flowtime* de 32 à 29.*

5 Stratégie de négociation

Nous décrivons ici la stratégie de négociation et nous esquissons le comportement des agents.

Le **modèle des pairs** est construit à partir des messages échangés entre agents. Avant le processus de négociation et après chaque réallocation bilatérale dans laquelle il est impliqué, l'agent v_i informe ses pairs de ce que chaque job J lui coûte, $c(J, v_i)$. Le modèle de la cible v_k par le sujet v_i repose sur :

1. la base de croyances du sujet, éventuellement partielle ou obsolète, qui contient les croyances concernant les coûts des jobs pour v_k ($c^i(J, v_k)$, $\forall J \in \mathcal{J}$) et donc les croyances concernant la charge de travail de v_k ($w_k^i(\vec{A}) = \sum_{J \in \mathcal{J}} c^i(J, v_k)$);

2. la stratégie de consommation de la cible supposée par le sujet, notée $(\mathcal{J}, \blacktriangleleft_k^i)$.

Par souci de cohérence, on écrit $c^i(J, v_i) = c(J, v_i)$ et $w_i^i(\vec{A}) = w_i(\vec{A})$.

Le sujet peut alors déduire :

— la durée de réalisation ($C_J^i(\vec{B}_k)$) du job J pour une cible k , éventuellement lui-même ($v_k = v_i$), après l'ajout ($C_J^i(\vec{B}_k \oplus \vec{T})$), la suppression ($C_J^i(\vec{B}_k \ominus \vec{T})$) et le remplacement de tâches ($C_J^i(\vec{B}_k \ominus \vec{T}_1 \oplus \vec{T}_2)$);

— la durée de réalisation d'un job J pour l'allocation, $C_J^i(\vec{A}) = \max_{v_k \in \mathcal{N}} C_J^i(\vec{B}_k)$ où $C_J^i(\vec{B}_i) = C_J(\vec{B}_i)$;

— le nœud **limitant** pour chaque job J , noté $v_{\max}^i(\vec{A}, J)$, i.e. le nœud v_k pour lequel la durée de réalisation de ce job est la durée de réalisation maximale, $C_J^i(\vec{B}_k) = C_J^i(\vec{A})$;

— le *flowtime* de l'allocation $C^i(\vec{A}) = \sum_{J \in \mathcal{J}} C_J^i(\vec{A})$.

La **règle d'acceptabilité** est une décision locale prise par un agent pour accepter ou décliner une réallocation.

Définition 8 (Acceptabilité). *Soit \vec{A} une allocation pour un problème MASTA+. La réallocation bilatérale $\gamma(T_1, T_2, v_i, v_j, \vec{A})$ est acceptable par l'agent $v_k \in \mathcal{N}$ ssi l'agent croit que le *flowtime**

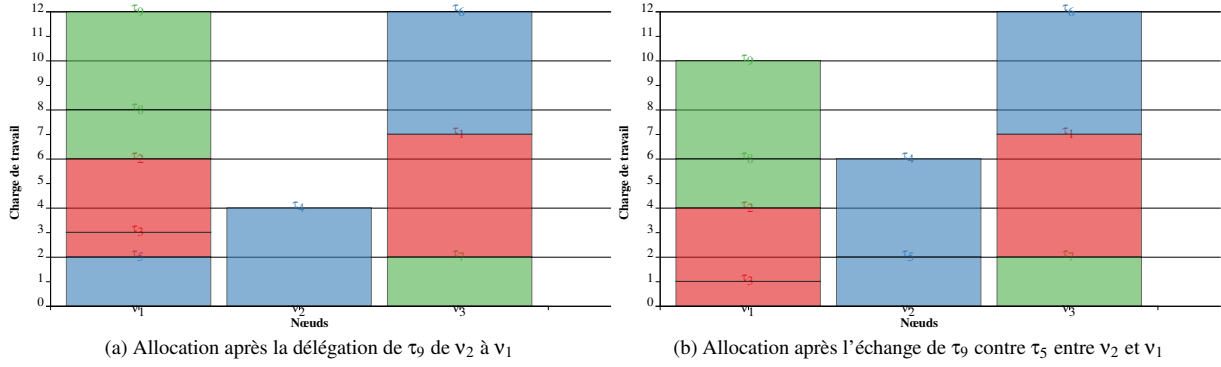


FIGURE 2 – Allocations de l'exemple fil rouge résultant de réallocations bilatérales

diminue strictement,

$$\sum_{J \in \mathcal{J}} \max_{v_o \in \mathcal{N} \setminus \{v_i, v_j\}} (C_j^k(\overrightarrow{B_i \ominus T_1 \oplus T_2}), C_j^k(\overrightarrow{B_j \ominus T_2 \oplus T_1}), C_j^k(\overrightarrow{B_o})) < C^k(\overrightarrow{A}) \quad (12)$$

L'acceptabilité repose sur la croyance à propos des durées de réalisation des jobs pour les nœuds avant et après la réallocation (Eq. 12).

Nous proposons ici un processus où les agents initient des négociations bilatérales concurrentes qui doivent aboutir à des réallocations socialement rationnelles.

La **stratégie d'offre** d'un agent identifie une délégation en trois étapes. Il s'agit donc pour un agent v_i de sélectionner une offre, c'est-à-dire une tâche à déléguer à un receveur dans un ensemble \mathcal{N}' afin de réduire la durée de réalisation d'un job pour lequel il est limitant dans un ensemble \mathcal{J}' . Initialement, $\mathcal{J}' = \mathcal{J}$, $\mathcal{N}' = \mathcal{N}$. Notons que si la section précédente définit des réallocations de listes de tâches, notre stratégie d'offre se limite ici à des listes de taille 1.

1. Sélection d'un job. Afin de réduire non seulement la durée de réalisation globale d'un job qui lui est affecté mais également celles des jobs qui suivent dans son lot, notre heuristique sélectionne le job J_* le plus prioritaire parmi ceux dont il est l'agent limitant,

$$J_* = \min_{\triangleleft_i} \{J \in \text{jobs}(B_i) \cap \mathcal{J}' \mid v_{\max}^i(\overrightarrow{A}, J = v_i)\} \quad (13)$$

2. Sélection d'un receveur. Les jobs d'un receveur impactés par la délégation sont ceux placés après J_* selon \triangleleft_j . Afin de ne pas augmenter la durée de réalisation de ces jobs, notre heuristique sélectionne un receveur v_* pour qui la

somme des différences entre la durée de réalisation pour l'allocation et celle pour l'agent est la plus grande,

$$v_* = \text{rnd} \left\{ \arg \max_{v_j \in \mathcal{N}'} \sum_{J_* \triangleleft_j} (C_j^i(\overrightarrow{A}) - C_j^i(\overrightarrow{B}_j)) \right\} \quad (14)$$

où rnd est une fonction de choix aléatoire qui à chaque ensemble de nœuds associe un nœud de cet ensemble.

3. Sélection d'une délégation. Afin de réduire les durées de réalisation, l'initiateur sélectionne une tâche distante, c.-à-d. dont la délégation réduira son coût d'exécution. Notre heuristique sélectionne la tâche du job J_* ou des jobs qui le précèdent dans \overrightarrow{B}_i avec le meilleur gain en terme de coût. En cas d'égalité, c'est la tâche prioritaire du lot qui est choisie,

$$\min_{\triangleleft_i} \left\{ \arg \max_{\tau \in \mathcal{T}' \cap B_i \cap \{J \mid J = J_* \vee (J \triangleleft_j J_*)\}} c(\tau, v_i) - c(\tau, v_*) \right\}$$

Afin de déterminer l'offre, on distingue une stratégie **conservatrice** qui propose la délégation $\delta([\tau_*], v_i, v_*, \overrightarrow{A})$ si elle est acceptable pour l'initiateur (cf Def. 8) comme dans [3] et une stratégie **libérale** qui propose des délégations même si elles ne sont pas acceptables.

La **stratégie de contre-offre** d'un répondant v_j à une délégation $\delta(\tau_1, v_i, v_j, \overrightarrow{A})$ lui permet de proposer une tâche dans l'ensemble $\mathcal{T}' = \mathcal{T}$ en échange de celle proposée.

1. Sélection d'une tâche. Afin de réduire les durées de réalisation, le donneur sélectionne une tâche distante, i.e. dont la délégation réduira son coût puisqu'elle sera exécutée localement. Notre heuristique sélectionne la tâche distante qui est la plus prioritaire.

$$\tau_* = \min_{\triangleleft_j} \{ \tau \in \mathcal{T}' \cap \overrightarrow{B}_j \mid c(\tau, v_j) - c(\tau, v_i) < 0 \} \quad (15)$$

2. Validation. Dans le but de garantir la convergence du processus de négociation, le déclenchement d'un échange par un répondant est soumis à son acceptabilité de l'échange. De plus, par souci d'efficacité, un échange est déclenché à condition qu'il permette de réduire le *flowtime* plus encore que la délégation initialement proposée. Formellement,

$$\begin{aligned} & \sum_{J \in \mathcal{J}} \max_{\forall v_o \in \mathcal{N} \setminus \{v_i, v_j\}} \\ & (C_J^k(\overrightarrow{B_i \ominus \tau_1 \oplus \tau_*}), C_J^k(\overrightarrow{B_j \ominus \tau_* \oplus \tau_1}), C_J^k(\overrightarrow{B_o})) \\ & < \sum_{J \in \mathcal{J}} \max_{\forall v_o \in \mathcal{N} \setminus \{v_i, v_j\}} \\ & (C_J^k(\overrightarrow{B_i \ominus \tau_1}), C_J^k(\overrightarrow{B_j \oplus \tau_1}), C_J^k(\overrightarrow{B_o})) \end{aligned} \quad (16)$$

Dans le cas contraire, \mathcal{T}' devient $\mathcal{T}' \setminus \{\tau_*\}$ dans l'étape 1. En cas d'échec, aucune contre-partie n'est proposée et le répondant évalue l'acceptabilité de la délégation pour la rejeter ou l'accepter.

Comportement d'agent. Dans notre approche, une réallocation bilatérale de tâches est le résultat de négociations entre agents qui adoptent tous le même comportement spécifié dans [4] par un automate qui consiste en **2 phases** successives : 1. les proposants adoptent la stratégie d'offre conservatrice et les répondants acceptent une délégation s'ils la considère acceptable. Ils la rejettent sinon ;

2. les agents adoptent la stratégie d'offre libérale et la stratégie de contre-offre. Si au moins une réallocation est réalisée, ces deux phases sont réitérées. Sinon le processus est clos.

Exemple 3 (Stratégie de négociation). *Considérons l'allocation \vec{A} pour le problème MASTA+ de l'exemple 1 où l'agent v_3 , pour lequel on suppose les croyances à jour, reçoit de la part de l'agent v_2 la proposition de la délégation $\delta(\tau_9, v_2, v_1, \vec{A})$. L'agent v_1 sélectionne la tâche avec le gain le plus important (Eq. 15), $\tau_* = \tau_5$. L'échange est acceptable, le *flowtime* global décroît strictement et il est meilleur que celui après la seule délégation (cf. figures 1b et 2b). En résumé, le répondant v_1 propose τ_5 en contre-partie de la tâche τ_9 pour atteindre l'allocation $\vec{A}' = \sigma(\tau_9, \tau_5, v_2, v_1, \vec{A})$.*

6 Évaluation empirique

Après une présentation des conditions expérimentales, nous comparons empiriquement notre approche avec une heuristique classique.

1. <https://gitlab.univ-lille.fr/maxime.morge/smastaplus/-/tree/master/doc/specification>

L'application pratique que nous considérons est le déploiement distribué du patron de conception MapReduce pour le traitement de jeux de données massives sur une grappe de serveurs, comme avec Spark [16]. Nous nous focalisons ici sur la phase *reduce* des jobs MapReduce que nous formalisons par un problème MASTA+ où plusieurs jobs sont soumis de façon concurrente. La fonction de coût est définie telle que :

$$\begin{aligned} c_i(\tau, v_j) &= \sum_{\rho_j \in \mathcal{R}_\tau} c_i(\rho_j, v_j) \\ \text{avec } c_i(\rho_j, v_i) &= \begin{cases} |\rho_j| & \text{si } v_i \in l(\rho_j) \\ \kappa \times |\rho_j| & \text{sinon} \end{cases} \end{aligned} \quad (17)$$

où nous avons fixé empiriquement $\kappa = 2$ comme une valeur réaliste pour capturer le surcoût induit par la récupération des ressources distantes.

Notre prototype [4] est implémenté avec le langage de programmation Scala et la bibliothèque Akka [11] adaptée aux applications orientées messages, fortement concurrentes, distribuées et robustes. Nous supposons que : (a) le délai de transmission des messages est arbitraire mais non négligeable, (b) l'ordre des messages par pair émetteur-récepteur est préservé, (c) la distribution des messages est garantie. Les expériences ont été réalisées sur une lame munie de 20 CPUs avec 512Go de RAM.

Nous considérons des instances de MASTA+ avec $m \in [2; 16]$ nœuds/agents, $\ell = 4$ jobs, $n = 3 \times \ell \times m$ tâches et 10 ressources par tâche. Chaque ressource ρ_i est répliquée 3 fois et $|\rho_i| \in [0; 100]$. Nous générons 10 instances de MASTA+, et pour chacune nous générons aléatoirement 10 allocations initiales. Nous évaluons les médianes et les écarts types de deux métriques : la durée de réalisation (Eq. 7) et le temps de réordonnement.

Les hypothèses que nous voulons tester sont : (1) la durée de réalisation atteinte par notre stratégie est proche de celle obtenue par l'approche classique ; (2) notre approche accélère le réordonnement.

Les figures 3a et 3b comparent respectivement la durée de réalisation et le temps de réordonnement de notre stratégie de délégation avec celle de notre stratégie d'échange ainsi qu'avec un algorithme de montée en gradient qui débute avec la même allocation initiale générée aléatoirement. À chaque étape, l'algorithme de montée en gradient sélectionne parmi tous les

2. Les algorithmes DCOP que nous avons évalués ne passent pas à l'échelle sur ce type de problème.

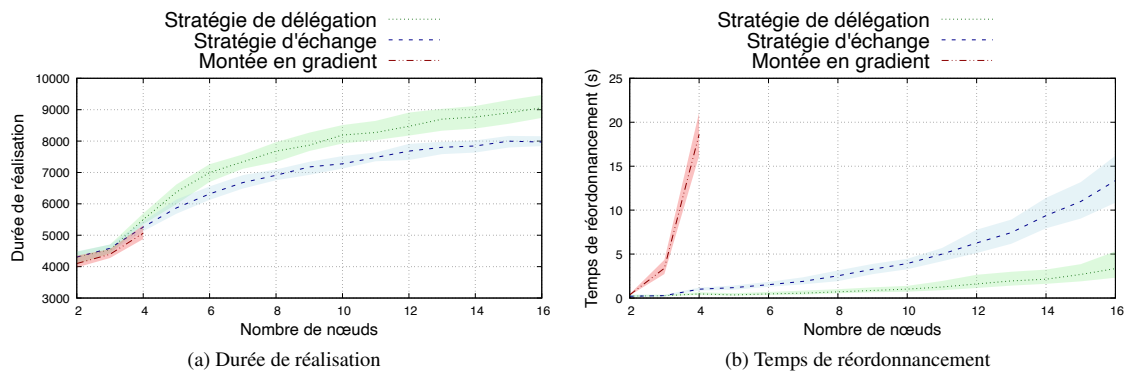


FIGURE 3 – Durée de réalisation et temps d'ordonnement pour nos stratégies et la montée en gradient

échanges unaires possibles celui qui minimise la durée moyenne de réalisation. Nous observons que notre stratégie d'échange atteint des solutions de qualité similaire aux solutions obtenues par la montée en gradient et, évidemment, légèrement meilleure que la stratégie de délégation. Nous observons également que, si le temps de réordonnement de la stratégie d'échange est légèrement moins bon que pour la stratégie de délégation, il reste approximativement identique et donc largement meilleur que celui de l'algorithme de montée en gradient qui croît exponentiellement avec le nombre de noeuds. Même si, contrairement à ce dernier, notre stratégie n'évalue pas tous les échanges unaires possibles, elle s'avère efficace en sélectionnant les tâches distantes dont la délégation réduit le coût. Les échanges permettent d'échapper à des minimums locaux et de déclencher de nouvelles délégations.

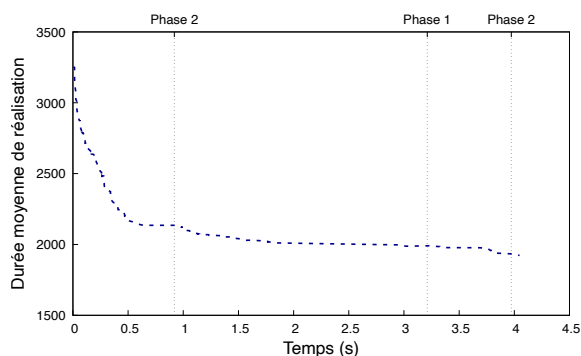


FIGURE 4 – Exécution de la stratégie d'échange

La figure 4 représente, pour une exécution particulière, l'évolution du *flowtime* moyen de la stratégie d'échange. Comme la stratégie de délégation, elle opère 87 délégations dans la phase

1. Ensuite, la stratégie d'échange poursuit en opérant 32 échanges dans la phase 2 qui, à leur tour, permettent de déclencher 23 délégations et 14 échanges. La stratégie d'échange prolonge la stratégie de délégation en atteignant une allocation similaire avec le même temps de réordonnement tout en permettant de l'améliorer par la suite.

7 Conclusion

Dans cet article, nous avons proposé un système multi-agents pour la réallocation de tâches sur des noeuds en fonction de la localisation des ressources nécessaires afin de réduire la durée moyenne de réalisation de jobs concurrents. Nos expériences montrent que la durée moyenne de réalisation atteinte par notre stratégie est proche de celle atteinte par l'approche heuristique classique et qu'elle réduit considérablement le temps de réordonnement. Nous avons étendu notre cadre de négociation pour considérer n'importe quelle réallocation bilatérale, en particulier des échanges. Nous avons proposé une stratégie de contre-offre pour sélectionner une contre-partie susceptible de déclencher des échanges de tâches afin d'échapper à des minimums locaux et déclencher de nouvelles délégations. Notre nouvelle stratégie d'échange prolonge notre stratégie de délégation en atteignant une allocation similaire avec le même temps de réordonnement tout en permettant de l'améliorer par la suite.

Nos travaux futurs porteront sur une stratégie d'offres constituées de lot de tâches pour réduire le temps de réordonnement et améliorer la durée moyenne de réalisation. Pour l'instant, nos expérimentations portent sur la réallocation d'un ensemble de tâches appartenant à plusieurs jobs.

Comme pour [1], nous souhaitons intégrer la consommation des tâches afin d'étudier l'adaptation de notre système à des ensembles de tâches qui évoluent dans le temps. Dans le même ordre d'idées, les travaux futurs doivent également intégrer la réallocation des tâches dans un processus d'approvisionnement pour ajouter ou supprimer en cours d'exécution des nœuds de calcul en fonction des besoins des utilisateurs de manière optimale afin de proposer un système multi-agents élastique.

Références

- [1] Quentin Baert, Anne-Cécile Caron, Maxime Morge, Jean-Christophe Routier, and Kostas Stathis. An adaptive multi-agent system for task reallocation in a MapReduce job. *Journal of Parallel and Distributed Computing*, 153 :75–88, July 2021.
- [2] Ellie Beauprez, Luc Bigand, Anne-Cécile Caron, Maxime Morge, and Jean-Christophe Routier. Réaffectation de tâches de la théorie à la pratique : état de l'art et retour d'expérience. In *Actes des JFSMA*, pages 51–60. Cépaduès, 2021.
- [3] Ellie Beauprez, Anne-Cécile Caron, Maxime Morge, and Jean-Christophe Routier. Une stratégie de négociation multi-agents pour réduire la durée moyenne de réalisation. In *Actes des JFSMA*, pages 31–40. Cépaduès, 2021.
- [4] Ellie Beauprez and Maxime Morge. Scala implementation of the Extended Multi-agents Situated Task Allocation. <https://gitlab.univ-lille.fr/maxime.morge/smastaplus>, 2020.
- [5] Aurélie Beynier, François Charpillet, Daniel Szer, and Abdel-illah Mouaddib. DEC-MDP / DEC-POMDP. In Olivier Sigaud Olivier Buffet, editor, *Markov Decision Processes in Artificial Intelligence*, pages 277–313. Wiley-ISTE, 2010.
- [6] J. Bruno, E. G. Coffman, Jr., and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Commun. ACM*, 17(7) :382–387, July 1974.
- [7] Bo Chen, Chris N. Potts, and Gerhard J. Woeginger. *Handbook of combinatorial optimization*, chapter A review of machine scheduling : Complexity, algorithms and approximability, pages 1493–1641. Springer, 1998.
- [8] Han-Lim Choi, Luc Brunet, and Jonathan P How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4) :912–926, 2009.
- [9] WA Horn. Minimizing average flow time with parallel machines. *Operations Research*, 21(3) :846–847, 1973.
- [10] Shijie Li, Rudy R. Negenborn, and Gabriel Lodewijks. A Distributed Constraint Optimization Approach for Vessel Rotation Planning. In *Computational Logistics*, pages 61–80. Springer, 2014.
- [11] Lightbend. Akka is the implementation of the actor model on the JVM. <http://akka.io>, 2020.
- [12] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1) :97–102, 1 1982.
- [13] Andrea Schaefer, Y. Shoham, and Moshe Tennenholtz. Adaptive load balancing : A study in multi-agent learning. *Journal of Artificial Intelligence Research*, 2 :475–500, 1995.
- [14] Joanna Turner, Qinggang Meng, Gerald Schaefer, and Andrea Soltoggio. Distributed strategy adaptation with a prediction function in multi-agent task allocation. In *Proc. of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 739–747, 2018.
- [15] Michael P Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1 :1–23, 1993.
- [16] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauly, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets : A fault-tolerant abstraction for in-memory cluster computing. In *Proc. of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*; San Jose, CA, USA, pages 15–28, 2012.

Session 2 : Négociation multi-agent, consensus

Un protocole de concessions monotones pour la formation distribuée de coalitions

J. Guéron^a G. Bonnet^a
 josselin.gueron@unicaen.fr gregory.bonnet@unicaen.fr

^aNormandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000, France

Résumé

Nous étudions une adaptation au cadre de la formation de coalitions d'un protocole de négociation multilatérale initialement proposé par Ulle Endriss. Ce protocole utilise un mécanisme de concession monotone et nous proposons de nouvelles stratégies de concession prenant en compte la notion de coalition. Nous étudions par la suite comment différents critères influencent les performances du protocole, y compris en termes de types de concession décrivant les concessions acceptables pour les autres agents. Nous montrons expérimentalement que deux de nos stratégies sont efficaces lorsqu'elles sont associées à trois des sept types de concession.

Mots-clés : Formation distribuée de coalitions, Théorie des jeux, Négociation

Abstract

We study an adaptation, in the context of coalition formation, of a multilatera negotiation protocol originally proposed by Ulle Endriss. This protocol uses a monotonic concession mechanism and we propose new concession strategies that take into account the notion of coalition. We then study how different criteria influence the performance of the protocol (e.g. the concession types which describe the acceptable concessions for the other agents). We show experimentally that two of our strategies are efficient when they are associated with three of the seven types of concession.

Keywords: Distributed Coalition Formation, Game Theory, Negotiation

1 Introduction

Dans un système multi-agent (SMA), les agents individuels ne sont pas toujours capables de réaliser certaines tâches seuls. Lorsque le système est composé d'agents égoïstes et rationnels, une des réponses à ce problème est la formation de coalitions. Ici, les agents forment des groupes, appelés coalitions, afin de réaliser conjointement

les tâches qui ne peuvent pas être traitées individuellement. La formation de coalitions est un processus en trois étapes : la recherche d'une structure de coalitions – c'est-à-dire d'un ensemble de coalitions acceptables (stable) pour tous les agents –, la formation de ces coalitions, et la répartition des gains entre les agents [10]. Toutefois, résoudre un problème de formation de coalitions de manière centralisé n'est pas viable pour certaines applications, telles que les chaînes logistiques ou les réseaux électriques intelligents. En effet, rechercher une structure de coalitions optimale, c'est-à-dire qui maximise le bien-être social et qui est stable, dans un cadre centralisé implique d'explorer le treillis des structures, dont la taille croît exponentiellement. Former des coalitions de manière distribuée, voire décentralisée, devient alors intéressant, même si cela implique des redondances ou des coûts de communication et de coordination. Beaucoup de travaux sur la formation de coalitions distribuée ont été proposés dans la littérature, mais ils sont souvent spécifiques au contexte [6, 14, 15, 16, 17, 18, 19], ou comprennent des contraintes fortes [3, 6, 17, 18, 19]. Par exemple, certains travaux fixent une taille de coalition maximale [14, 15, 16], ou proposent des algorithmes pour des jeux spécifiques (par exemple ayant des propriétés particulières pour leur fonction caractéristique) [6, 17]. D'autres travaux encore proposent des algorithmes distribués mais dont certaines parties sont toujours centralisées en raison de la présence d'un commissaire-priseur [6, 14, 15, 16, 17]. Une question qui se pose alors naturellement est celle de la formation distribuée de coalitions, sans contexte spécifique ni contraintes sur le modèle.

Nous proposons dans cet article un protocole distribué fondé sur des concessions monotones. Il s'agit d'une adaptation pour la formation de coalitions d'un protocole de concessions monotones proposé par Ulle Endriss [5] dans le contexte de négociations multilatérales. Dans ce protocole, les agents font des propositions et ensuite, selon une stratégie de concession qui caractérise quel agent doit concéder, les agents

peuvent concéder et faire une autre proposition. Cette dernière suit également une autre stratégie qui caractérise quelles propositions seront acceptables pour les autres agents. Le point clé de notre contribution réside dans le fait que le protocole originel ne considère des négociations qu'entre les agents individuels, alors que dans la formation de coalitions nous devons prendre en compte des groupes d'agents, c'est-à-dire les coalitions. Nous adaptons en particulier les stratégies de concession en fonction non seulement des gains individuels mais aussi du gain des coalitions. Nous analysons ensuite expérimentalement notre protocole, en termes de partitions explorées, de prix de la stabilité et d'optimalité.

Cet article est structuré comme suit. Dans la section 2, nous présentons les jeux de coalitions et les travaux relatifs à la formation distribuée de coalitions. Dans la section 3, nous faisons le lien entre négociation et formation de coalitions, puis nous décrivons notre protocole. Enfin, la section 4 est consacrée aux résultats expérimentaux.

2 État de l'art

2.1 Formation de coalitions

Quand les agents coopèrent, ils forment une *coalition*. Celle-ci produit un certain montant d'*utilité*. Une partition des agents en coalitions est appelée une *structure de coalitions*.

Definition 1 (Jeu de coalitions) *Un jeu de coalitions est un tuple $\mathcal{G} = \langle N, v \rangle$ où $N = \{a_1, \dots, a_n\}$ est un ensemble d'agents, et $v : 2^N \rightarrow \mathbb{R}$ est la fonction caractéristique qui indique l'utilité $v(C)$ de chaque coalition $C \subseteq N$.*

Une *solution* à un jeu de coalitions à utilité transférable est définie comme suit.

Definition 2 (Solution) *Une solution à un jeu de coalitions \mathcal{G} est un tuple $S_{\mathcal{G}} = \langle CS, \vec{x} \rangle$ où CS est une structure de coalitions de N , $\vec{x} = \{x_1, \dots, x_n\}$ est un vecteur de gains pour les agents où $x_i \geq 0$ est le gain de l'agent a_i .*

Les agents étant supposés égoïstes, quand une solution est proposée, tous les agents doivent l'accepter, c'est-à-dire qu'ils ne doivent pas vouloir former ou rejoindre une autre coalition où ils gagneraient plus. Une solution acceptable par tous les agents est dite *stable*. Différents critères

de stabilité peuvent être définis et l'ensemble des solutions à un jeu respectant un de ces critères est appelé un *concept de solution*.

Intéressons-nous au concept du *cœur* et sa généralisation, l' *ϵ -cœur* [7, 13]. Le cœur est l'ensemble des solutions $\langle CS, \vec{x} \rangle$ pour lesquels il n'existe aucune autre coalition qui pourrait être formée et qui produirait une utilité supérieure à la somme des gains de ses agents dans \vec{x} . Si le cœur peut être vide, il existe une variante non vide, appelée *ϵ -cœur*. Cette variante permet aux agents de réduire leur gain de la valeur ϵ , dans le but de rendre une solution stable. Ainsi, nous utilisons dans cet article uniquement l' *ϵ -cœur*, car toute solution à un jeu est comprise dans un *ϵ -cœur* dès lors que l'on a un ϵ suffisamment grand, étant donné que ϵ représente la pire perte en termes de gain parmi les agents afin d'assurer la stabilité. Plus précisément, nous nous intéressons à l' *ϵ -cœur* ayant le plus petit ϵ pour lequel une solution existe, appelé le *dernier cœur* [4].

Definition 3 (ϵ -cœur) *Une solution (CS, \vec{x}) appartient à l' ϵ -cœur si et seulement si :*

$$\forall C \subseteq N, x(C) \geq v(C) - \epsilon \text{ avec } x(C) = \sum_{i \in C} x_i$$

Trouver de manière exacte une solution qui appartient à un concept de solution est un problème d'optimisation complexe. Classiquement, une solution approchée est acceptable. Le calcul se fait en deux étapes : (1) trouver une structure de coalitions qui maximise le bien-être social et qui est dite *optimale*; (2) partager les gains entre agents selon une règle de répartition qui fait sens. L'approche classique pour trouver la structure de coalitions optimale, par exemple incarnée par l'algorithme de Rahwan et Jennings [8], consiste à débiter l'exploration par la *grande coalition* (qui contient tous les agents) puis la décomposer et évaluer si cette décomposition est bénéfique pour le bien-être social. Si c'est le cas, le bien-être social est enregistré puis la décomposition continue. Si la décomposition n'était pas bénéfique, l'algorithme revient en arrière et essaye une autre décomposition. Toutefois, cette approche soulève deux problèmes : le premier est que cette façon de faire n'est pas adéquate pour beaucoup d'applications réelles, et la seconde est que dans le concept de cœur, la structure maximisant le bien-être social n'est pas nécessairement stable [4] (par exemple quand le cœur est vide trivialement).

2.2 Formation distribuée de coalitions

Il existe beaucoup de travaux sur la formation distribuée de coalitions dans la littérature, mais ils sont souvent spécifiques au contexte ou imposent des contraintes fortes sur le jeu. Nous pouvons trouver des contraintes sur la fonction caractéristique, où la forme de celle-ci est connue [6, 17], ou des contraintes sur les solutions possibles, qui limitent par exemple arbitrairement la taille des coalitions [14, 15, 16] ou les restreignent au sein d'un graphe [3, 18, 19]. Certains modèles incluent un commissaire-priseur qui orchestre le processus, ce qui est une forme de recentralisation [6, 14, 15, 16, 17].

Par exemple, Shehory et Kraus [14, 15, 16] forment des coalitions pour des problèmes d'affectation de tâches. Ici, les agents ne connaissent que leur propre utilité et communiquent avec les autres agents pour calculer une utilité espérée pour chaque coalition. Le processus est le suivant : les agents créent une liste restreinte de coalitions de taille maximale fixée qu'ils souhaitent former. Ils contactent les agents impliqués afin d'estimer leurs valeurs. Chaque agent calcule localement une valeur appelée *poids*, fondée sur l'utilité espérée, le coût de formation et la taille de chaque coalition. Ensuite, la coalition avec le poids le plus faible est formée. Le processus est répété jusqu'à ce que tous les agents rejoignent une coalition. Les solutions sont considérées comme stables tant que les coalitions formées peuvent réaliser leurs tâches, même si ce ne sont pas des solutions optimales.

2.3 Protocole de négociation monotone

C'est dans un cadre différent qu'un protocole intéressant a été proposé par Ulle Endriss [5]. Dans ce travail, les agents doivent se mettre d'accord sur une proposition commune et négocient en suivant un protocole de concessions monotones. Au premier tour, chaque agent fait une proposition initiale, en proposant celle qui lui rapporte le gain le plus élevé. À chaque tour suivant, les agents font des propositions simultanées, ensuite les agents peuvent maintenir leur proposition ou concéder, et ainsi de suite jusqu'à ce qu'un accord commun ou un conflit définitif émerge. Les propositions sont définies comme suit.

Definition 4 (Proposition) Une proposition $p_{a_i} = \{x_1, \dots, x_n\}$ de l'agent a_i est un vecteur de gain où $x_{a_j}(p_{a_i}) \geq 0$ est le gain de a_j .

Ensuite, la deuxième étape implique deux concepts importants : la *stratégie de concession* et la *type de concession*. La première décrit comment les agents décident qui doit concéder, et la seconde décrit comment les concessions (i.e. les nouvelles propositions) doivent être faites. Les principales stratégies sont *Willingness to Risk Conflict* (WRC) et *Product Increasing* (PI).

La stratégie WRC décrit comment un agent juge le risque s'il concède, comparé à ce que les autres agents lui offrent. Autrement dit, l'agent qui concède est celui qui perd le moins de gain entre sa propre proposition et celles des autres.

Definition 5 (Willingness to Risk Conflict)

L'agent qui concède est l'agent a_i pour qui la valeur Z_{a_i} est la plus petite, où :

$$Z_{a_i} = \begin{cases} 1 & \text{si } x_{a_i}(p_{a_i}) = 0 \\ \frac{x_{a_i}(p_{a_i}) - \min_{\forall j \in N} x_{a_i}(p_{a_j})}{x_{a_i}(p_{a_i})} & \text{sinon} \end{cases}$$

La stratégie PI favorise une distribution égalitaire, c'est-à-dire que l'agent qui distribue le moins également les gains concédera.

Definition 6 (Product Increasing) L'agent qui concède est l'agent a_i pour qui la valeur Z'_{a_i} est la plus petite, où :

$$Z'_{a_i} = \prod_{\forall j \in N} x_{a_j}(p_{a_i})$$

Les types de concession décrivent les différentes façons que les agents ont de concéder afin que leur nouvelle proposition soit acceptable pour les autres agents. Ces types sont fondés sur la différence entre le vecteur de gain de l'ancienne et celui de la nouvelle proposition. Les différents types sont les suivants :

- Fort** : le gain de tous les autres agents croît,
- Faible** : le gain d'un autre agent croît,
- Pareto** : le gain des agents reste au moins égal, et croît pour au moins un,
- Égalitaire** : le gain minimal des autres croît,
- Utilitaire** : la somme des gains des autres croît,
- Nash** : le produit des gains des autres croît,
- Égocentrique** : le gain du concédant décroît.

3 Un protocole distribué

Comme mentionné en introduction, nous fondons notre protocole sur celui de Ulle Endriss [5]. En effet, un parallèle peut être fait entre

l'accord dans le protocole de négociation et la stabilité dans la formation de coalitions. Chacun de ces termes caractérise le moment où les agents sont tous d'accord. Ainsi, les concessions représentent des agents qui acceptent un gain plus faible afin d'atteindre la stabilité, comme un concept de solution qui autorise d'abandonner une part de gains pour éviter les déviations.

3.1 Propositions et gains

Désormais, une proposition est une solution : une structure de coalitions et un vecteur de gain.

Definition 7 (Proposition) *Étant donné un jeu \mathcal{G} , une proposition de l'agent a_i , notée p_i , est une solution $S_{\mathcal{G}} = \langle \mathcal{CS}, \vec{x} \rangle$ où \vec{x} est un vecteur de gains $\langle x_{a_j}^{C_j^i} \rangle$ où C_j^i est la coalition de l'agent a_j dans la proposition p_i .*

Pour évaluer une proposition, nous devons définir comment les agents distribuent l'utilité produite par les coalitions. Nous supposons que les agents souhaitent négocier sur le gain qu'ils reçoivent à la formation des coalitions. Notons que cet article s'intéresse au protocole distribué et non à la recherche de la meilleure distribution possible. Ainsi, nous considérons une règle de distribution des gains rationnelle et égalitariste.

Cette règle est rationnelle car les coalitions proposées génèrent assez d'utilité pour payer à chaque agent l'utilité produite par sa coalition singleton. Le reste est appelé le *surplus*.

Definition 8 (Surplus) *Le surplus S_C d'une coalition C est :*

$$S_C = v(C) - \sum_{a \in C} v(a)$$

Par définition, les coalitions singleton ont un surplus de 0. La règle est égalitaire car le surplus est distribué également entre les agents.

Definition 9 (Part de surplus) *La part de surplus $S_C^{a_i}$ d'un agent a_i dans sa coalition C , est le surplus de C divisé par son nombre d'agents :*

$$S_C^{a_i} = \frac{S_C}{|C|}$$

Definition 10 (Règle de distribution) *Le gain de l'agent a_i appartenant à la coalition C avec un surplus $S_C > 0$ est défini comme :*

$$x_{a_i}^C = v(\{a_i\}) + S_C^{a_i}$$

3.2 Stratégies et types de concession

À chaque pas de temps, les agents font des propositions, et suivant une stratégie de concession, un ou plusieurs agents doivent abandonner leurs propositions et en faire de nouvelles. Pour ceci, nous utilisons et adaptons la stratégie WRC présente dans les travaux d'Endriss. Nous proposons trois adaptations que nous présentons ci-après : une stratégie est fondée sur les gains individuels des agents, une sur l'utilité des coalitions, et une dernière basée sur le surplus des coalitions. Nous les appelons respectivement *WRC-classic*, *WRC-coalitions* et *WRC-surplus*.

Definition 11 (WRC-Classic) *L'agent qui concède est l'agent a_i pour qui la valeur Z_{a_i} est la plus petite, où :*

$$Z_{a_i} = \frac{x_i(p_i) - \min_{\forall j \in N} x_i(p_j)}{x_i(p_i)}$$

où $x_i(p_j)$ est le gain de l'agent a_i dans la proposition p_j .

Definition 12 (WRC-Coalitions) *L'agent qui concède est l'agent a_i pour qui la valeur Z_{a_i} est la plus petite, où :*

$$Z_{a_i} = \frac{x_{C_i}(p_i) - \min_{\forall j \in N} x_{C_i}(p_j)}{x_{C_i}(p_i)}$$

où $x_{C_i}(p_j)$ est la somme des gains de tous les agents dans la coalition C_i dans la proposition p_j , où C_i est la coalition dans laquelle l'agent a_i est dans sa propre proposition :

$$x_{C_i}(p_j) = \sum_{k \in C_i} x_k(p_j)$$

Definition 13 (WRC-Surplus) *L'agent qui concède est l'agent a_i pour qui la valeur Z_{a_i} est la plus petite, où :*

$$Z_{a_i} = \frac{S_{C_i}(p_i) - \min_{\forall j \in N} S_{C_i}(p_j)}{S_{C_i}(p_i)}$$

où $S_{C_i}(p_j)$ est la somme des parts de surplus de tous les agents dans la coalition C_i dans la proposition p_j , où C_i est la coalition dans laquelle l'agent a_i est dans sa propre proposition :

$$S_{C_i}(p_j) = \sum_{k \in C_i} S_{C_j^k}^k$$

où C_k^j est la coalition de l'agent a_k dans la proposition p_j .

Dans toutes ces stratégies, la condition où $Z_{a_i} = 1$ (l'agent ne peut plus concéder) si l'agent a_i ne gagne rien dans sa propre proposition est modifiée pour que $Z_{a_i} = 1$ si l'agent a_i se propose lui-même dans sa coalition singleton ou dans toute coalition dont le surplus est nul. Une telle règle représente l'absence de coopération car aucun agent n'acceptera d'être irrationnel.

Enfin, quand un agent concède, il doit faire une nouvelle proposition selon un certain type. Nous n'avons pas besoin d'adapter ces types à la formation car ils s'appliquent parfaitement aux vecteurs de gains, sans nécessité de changement.

3.3 Étapes du protocole

Maintenant que les éléments du protocole sont définis et adaptés, nous pouvons détailler les étapes du protocole.

1. Chaque agent calcule le surplus des coalitions dont il peut faire partie,
2. Chaque agent crée une liste noire de coalitions et structures de coalitions,
3. Au premier tour, chaque agent fait une proposition initiale en choisissant sa coalition, notée C^* , parmi celles qui maximisent sa part de surplus, puis en choisissant la structure de coalitions qui maximise le bien-être social, notée CS^* , et qui inclut la coalition choisie C^* ,
4. À chaque tour suivant, chaque agent garde sa proposition ou concède, selon sa stratégie de concession,
5. Si un agent concède, la structure de coalitions précédente CS^* est ajoutée à la liste noire, et l'agent essaye de construire une nouvelle proposition satisfaisant son type de concession avec une autre structure de coalitions $CS^{*'}$ qui inclut également sa coalition choisie C^* . Si une structure $CS^{*'}$ ne satisfait pas un type de concession, elle est ajoutée à la liste noire. S'il n'y a plus de structure de coalitions possible avec la coalition C^* , cette dernière est ajoutée à la liste noire, et l'agent choisit une autre coalition, $C^{*'}$, qui maximise sa part de surplus. Un agent qui propose une coalition $C^{*'}$ dont le surplus est égal à 0 se retire du processus car il ne pourra pas gagner quoique ce soit.

6. Répéter à partir de l'étape (4) jusqu'à ce qu'un accord soit atteint ou qu'aucun agent ne reste dans le processus, c'est-à-dire qu'aucun agent ne puisse faire une proposition où il gagnerait quelque chose (nous appellerons ce cas un *conflit*).

Si aucun agent ne peut faire une nouvelle proposition, le processus s'arrête et le protocole retourne la structure de coalitions singleton.

3.4 Propriétés

Le protocole originel se positionne au regard de certaines propriétés.

Terminaison : le protocole de négociation termine toujours,

Absence de blocage : si aucun agent ne peut effectuer une action, alors le protocole est dans un état terminal,

Vérifiabilité : il est possible de vérifier que tous les agents respectent les règles du protocole,

Confidentialité : les agents peuvent garder des informations privées,

Compositionnalité : la composition de deux propositions consécutives suivant le même type de concession est également une proposition simple du même type.

Notre protocole respecte les trois premières propriétés. Étant donné que le nombre de structures de coalitions et le nombre d'agents sont finis, et que les agents ne peuvent pas faire deux fois la même proposition grâce à la liste noire, la propriété de *terminaison* est respectée pour tout type de concession. Et ce même pour les concessions *faibles*, qui ne respectent pas la propriété dans le protocole originel en raison de boucles possibles sur les mêmes propositions.

La propriété d'*absence de blocage* est respectée car, à chaque pas de temps, au moins un agent doit concéder et, si un agent ne peut plus faire de proposition, celui-ci quitte le jeu. Si aucun agent ne reste dans le jeu, nous nous retrouvons dans l'état terminal de conflit, c'est-à-dire la formation de la structure de coalitions singleton.

La propriété de *vérifiabilité* est respectée car la fonction caractéristique et la règle de distribution sont connues, tout comme les types et stratégies de concession. Chaque agent peut donc vérifier les propositions des autres. Il peut également reconstituer leurs listes noires, ce qui a pour conséquence que la propriété de *confidentialité* n'est pas respectée.

Concernant la propriété de *compositionnalité*, celle-ci n'est pas respectée dans le cadre des concessions *faibles*. Prenons un exemple à trois agents a_1 , a_2 et a_3 . Supposons que a_1 fera deux concessions consécutives. Avant la première concession, a_2 est dans une coalition non-singleton. Lors de la première concession, a_2 est proposé dans sa coalition singleton (et donc baisse en gain) tandis que a_3 augmente en gain. Lors de la concession suivante, a_3 baisse en gain tandis que a_2 revient dans une autre coalition non-singleton (et donc récupère du gain). Cependant, rien n'empêche a_2 et a_3 d'avoir moins de gain dans la dernière proposition qu'avant les concessions, et donc que la *compositionnalité* n'est pas nécessairement respectée.

4 Expérimentations

Afin d'évaluer si notre protocole est efficace, et surtout pour comparer les différents types et stratégies de concession, nous procédons empiriquement. Nous générons des jeux aléatoires avec différentes fonctions caractéristiques et appliquons le protocole plusieurs fois sur chaque jeu, avec différents paramètres pour les agents. Ces paramètres sont le type et la stratégie de concession utilisés.

4.1 Protocole expérimental

Nous construisons 100 jeux différents avec des fonctions caractéristiques uniques, pour 8 agents. Il est fait l'hypothèse que les agents sont homogènes, c'est-à-dire qu'ils utilisent tous les mêmes stratégies et types de concession qui paramètrent le protocole.

Les fonctions caractéristiques sont tirées selon le modèle NDCS (*Normally Distributed Coalition Structures*) [11, 9]. Ce modèle permet d'avoir des fonctions caractéristiques structurées, mais sans contraindre fortement le modèle comme avec des structures monotones ou superadditives [4]. Ainsi, l'utilité $v(C)$ de chaque coalition $C \subseteq N$ est tirée selon une loi normale $\mathcal{N}(|C|, \sqrt{|C|})$. La fonction caractéristique est ensuite normalisée sur l'intervalle $[0, 1]$.

4.2 Mesures de performances

Afin d'évaluer notre modèle, nous définissons quatre mesures empiriques, respectivement fondées sur le dernier cœur, sur le meilleur ϵ -cœur atteignable par notre protocole, le bien-être social, et le nombre de Bell, c'est-à-dire le nombre

de partitions possibles pour un nombre d'agents donné [2, 12]. Les deux premières mesurent la distance entre les solutions trouvées par le protocole par rapport au dernier cœur, ce dernier comprenant les meilleures solutions stables que l'on puisse trouver. Ces mesures nous permettent d'évaluer la perte de stabilité due à la négociation et à notre règle de distribution spécifique du surplus. La troisième est le *prix de la stabilité* [1, 10], qui mesure le gain que les agents doivent abandonner, par rapport au bien-être social maximal, afin de former une structure stable. La dernière mesure le nombre de structures de coalitions qui ont été explorées, c'est-à-dire le nombre de structures qui ont été sélectionnées à un moment dans le protocole, sans être nécessairement proposées. Afin d'avoir des données commensurables et comparables, chaque mesure est définie comme un ratio, avec des valeurs comprises sur l'intervalle $[0, 1]$.

La première mesure est le ratio entre la valeur ϵ de l' ϵ -cœur auquel appartient la solution trouvée par le protocole et la valeur ϵ du dernier cœur, c'est-à-dire le minimum atteignable. Ainsi, si le dernier cœur n'est pas un 0-cœur, les solutions du protocole ne seront pas désavantagées contrairement à une comparaison directe au 0-cœur.

Definition 14 (Ratio au dernier cœur) *Étant donné un jeu \mathcal{G} , soient $\epsilon^*(\mathcal{G})$ la valeur ϵ du dernier cœur, et $\epsilon^p(\mathcal{G})$ la valeur ϵ de l' ϵ -cœur auquel appartient la solution retournée par le protocole. Le ratio au dernier cœur, noté $R^*(\mathcal{G})$, est défini par :*

$$R^*(\mathcal{G}) = \frac{1 - \epsilon^p(\mathcal{G})}{1 - \epsilon^*(\mathcal{G})}$$

La seconde est le ratio entre la valeur ϵ de l' ϵ -cœur auquel appartient la solution trouvée par le protocole et la valeur ϵ du meilleur ϵ -cœur (c'est-à-dire ayant la valeur ϵ la plus faible) atteignable par notre protocole s'il couvre toutes les partitions possibles, selon notre règle de distribution.

Definition 15 (Ratio à l'optimal-protocole) *Étant donné un jeu \mathcal{G} , soient $\epsilon^{p^*}(\mathcal{G})$ la valeur ϵ de l' ϵ -cœur optimal selon notre protocole et la règle de distribution, et $\epsilon^p(\mathcal{G})$ la valeur ϵ de l' ϵ -cœur auquel appartient la solution retournée par le protocole. Le ratio à l'optimal-protocole, noté $R^p(\mathcal{G})$, est défini par :*

$$R^p(\mathcal{G}) = \frac{1 - \epsilon^p(\mathcal{G})}{1 - \epsilon^{p^*}(\mathcal{G})}$$

La troisième, définie par Anshelevich *et al.* [1], est le ratio entre le bien-être social de la solution trouvée par le protocole, qui est stable, et le bien-être social maximal pour le jeu.

Definition 16 (Prix de la stabilité) *Étant donné un jeu \mathcal{G} , soient $\Pi(\mathcal{G})$ l'ensemble de toutes les structures de coalitions pour \mathcal{G} , et $S_G^p = \langle CS^p, \bar{x}^p \rangle$ une solution au jeu \mathcal{G} retournée par le protocole. Le prix de la stabilité pour cette solution, noté $PS(S_G^p)$, est défini par :*

$$PS(S_G^p) = \frac{\sum_{C \in CS^p} v(C)}{\max_{\pi \in \Pi(\mathcal{G})} \sum_{C' \in \pi} v(C')}$$

La dernière est la proportion de structures de coalitions explorées par le protocole, c'est-à-dire qui ont été proposées ou sélectionnées puis mises dans la liste noire durant le protocole, comparé au nombre total de structures possibles, donné par le nombre de Bell. Étant donné que chaque agent peut proposer chaque partition une seule fois, et que leur liste noire est individuelle, le treillis des structures de coalitions peut être exploré autant de fois qu'il y a d'agents.

Definition 17 (Ratio de Bell) *Étant donné le nombre de partitions explorées par l'agent a_i et le nombre de Bell, respectivement notés p_e^i et B_n (avec $n = |N|$), le ratio de Bell, noté $B^\%$, est :*

$$B^\% = \frac{\sum_{i \in N} p_e^i}{n \times B_n} \text{ avec } B_{m+1} = \sum_{k=0}^m \binom{m}{k} B_k$$

4.3 Résultats

Les figures 1 et 2 montrent le ratio au dernier cœur, le ratio à l'optimal-protocole, le prix de la stabilité et le ratio de Bell pour chaque stratégie de concession possible. La figure 1 montre les résultats pour les types de concession *faible*, *égocentrique*, *Nash*, *utilitaire* et *fort*, et la figure 2 les résultats pour les types *égalitaire* et *Pareto*. Les données sont triées sur le ratio de l'optimal-protocole du plus grand au plus petit, et les trois autres mesures suivent ce tri pour rester liées au jeu correspondant. Également, les figures sont présentées dans un ordre particulier qui permet au lecteur de constater une évolution entre les résultats en fonction des différents types de concession, de ce qui semble empiriquement le moins bon, vers le meilleur.

Concernant les différentes mesures et leurs significations graphiques : plus le ratio de Bell, étiqueté *bell*, est proche de 0, moins il y a eu de partitions explorées. Pour les trois autres mesures, la meilleure valeur est 1. Pour le ratio au dernier cœur, étiqueté *least-core*, et le ratio à l'optimal-protocole, étiqueté *optimal*, cela signifie que les solutions retournées par le protocole sont respectivement présentes dans le dernier cœur du jeu, et présentes dans le dernier cœur du jeu connaissant la règle de distribution. Pour le prix de la stabilité, étiqueté *welfare*, cela signifie que les solutions retournées par le protocole maximisent le bien-être social. Si ces mesures s'éloignent de 1, cela montre une perte d'efficacité.

Tout d'abord, nous pouvons voir que la courbe du ratio de Bell a une forme similaire pour toutes les paires (type, stratégie). Nous pouvons estimer l'exploration moyenne sur tous les jeux à 10%, indépendamment du type et de la stratégie de concession. De plus, nous pouvons voir qu'il n'y a pas de corrélation apparente entre le fait d'explorer et une meilleure stabilité, ni l'inverse. Concernant le ratio à l'optimal-protocole, nous pouvons voir que les stratégies et les types de concession influent sur les résultats. Avec les types de concession *égocentrique*, *faible*, *utilitaire* et *Nash*, le ratio à l'optimal-protocole atteint des pertes de 40%, avec parfois des effondrements atteignant les 60% pour certaines paires de paramètres. Bien qu'elle comprenne les plus gros effondrements, la stratégie WRC-surplus comprend avec la stratégie WRC-coalitions le meilleur taux de valeurs optimales. Plus précisément, sur les trois derniers types de concession (*fort*, *égalitaire* et *Pareto*), la stratégie WRC-classic atteint des pertes maximales d'environ 30%, contre 20% pour les deux autres stratégies. Ces deux stratégies ont donc les meilleurs résultats, avec environ 60% des jeux avec un ratio à l'optimal-protocole de 1. Les deux dernières mesures, le ratio au dernier cœur et le prix de la stabilité, se comportent de la même manière, à savoir qu'elles comportent des pics d'effondrement sur les mêmes jeux. Pour les types *faible* et *égocentrique*, nous pouvons voir de grandes pertes d'efficacité (régulièrement au dessus de 30%), et ce pour toutes les stratégies. En s'intéressant aux différences entre les stratégies sur ces types, nous pouvons voir que les pertes sont plus limitées avec WRC-coalitions. Les valeurs entre WRC-classic et WRC-surplus sont proches, bien que cette dernière semble décroître un peu moins. Pour les types *Nash* et *utilitaire*, les pertes sont moins grandes. Elles sont de

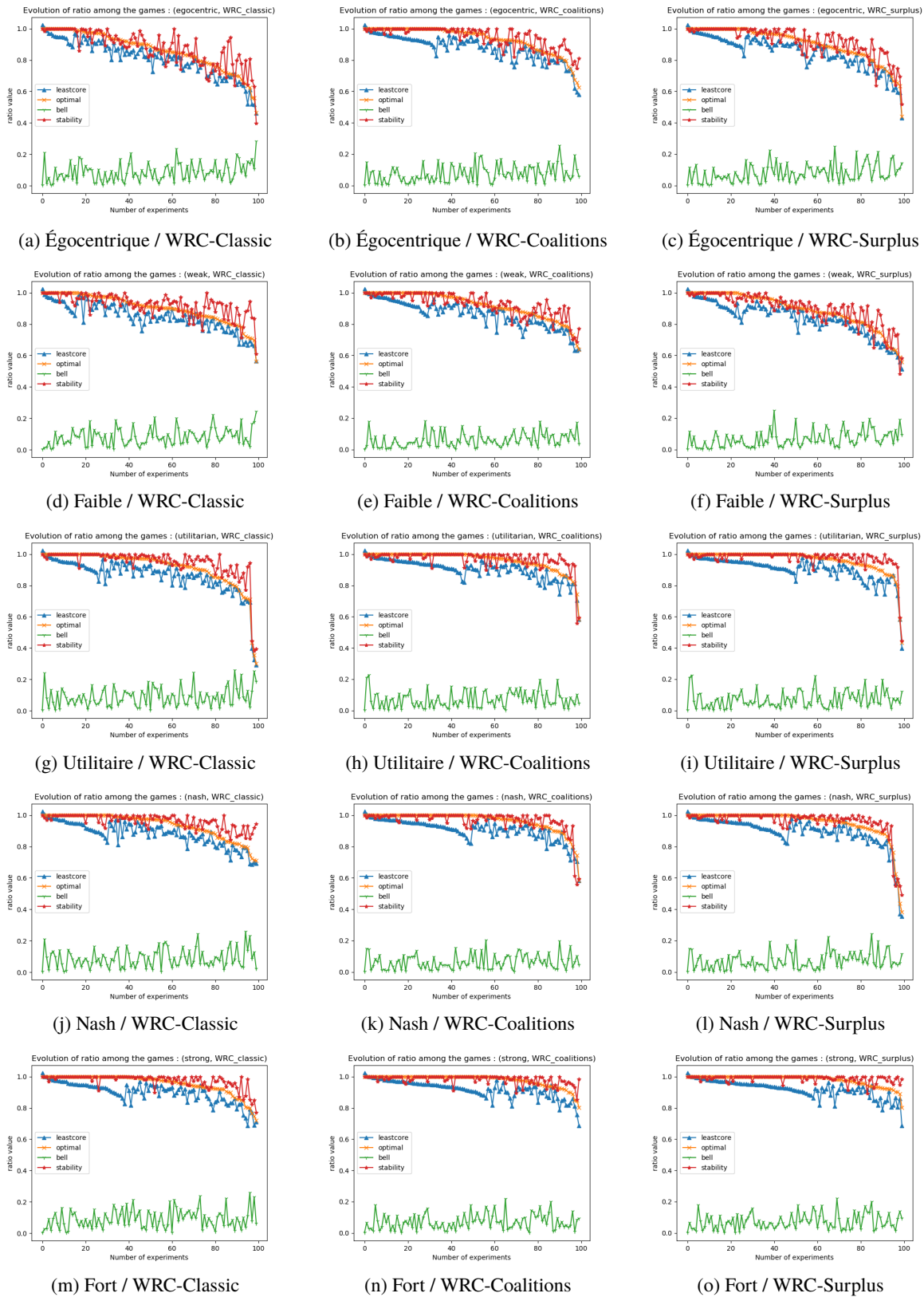


FIGURE 1 – Différents ratios pour un couple (Type / Strategy) de concession

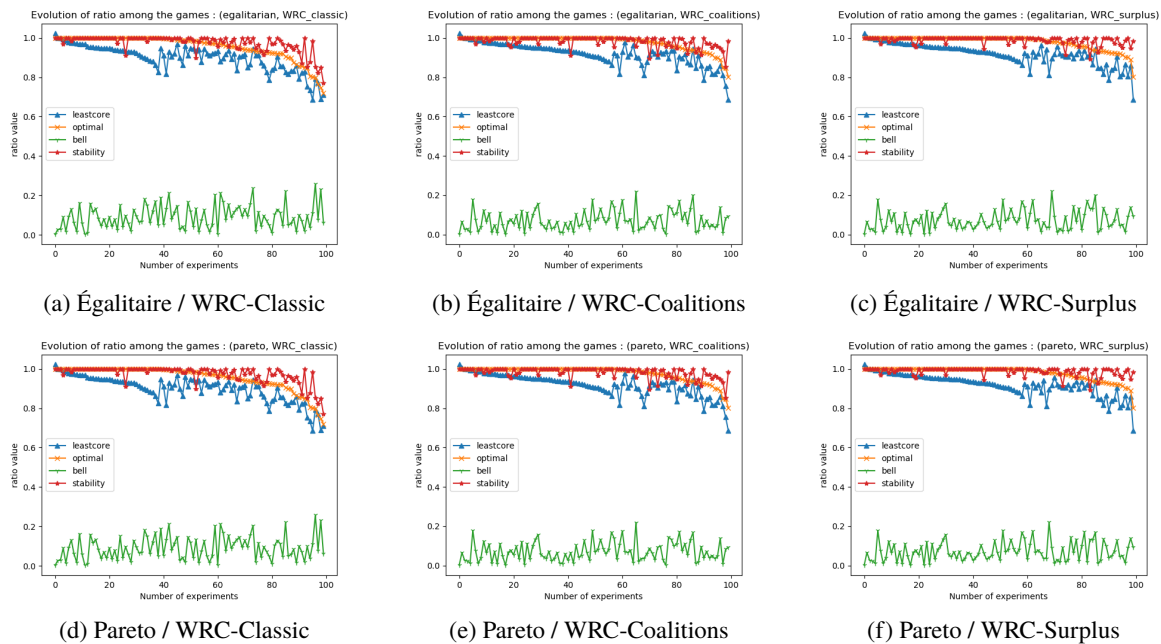


FIGURE 2 – Différents ratios pour un couple (Type / Strategy) de concession

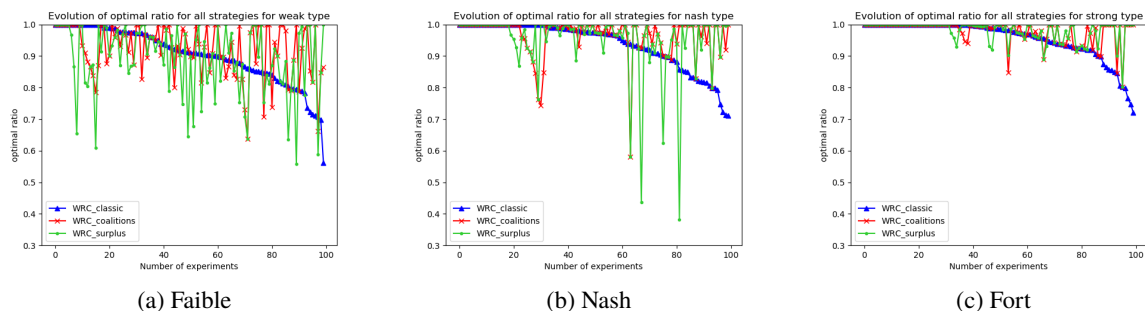


FIGURE 3 – Exemples de différences entre stratégies

l'ordre de 20%, bien qu'il y ait des effondrements autour de 60%, comme avec le ratio à l'optimal-protocole. Pour le type *Nash*, les pertes de bien-être social sont réduites avec WRC-classic, mais WRC-coalitions est légèrement meilleure pour le ratio au dernier cœur. Pour le type *utilitaire*, ce sont WRC-coalitions et WRC-surplus qui minimisent les pertes sur les mesures. Enfin, pour les trois derniers types (*fort*, *égalitaire* et *Pareto*), elles semblent toutes similaires. Il y a en réalité des différences mineures sur le nombre de partitions explorées et le nombre de concessions effectuées, mais les résultats du protocole sont identiques. Ce phénomène vient, à notre avis, de la règle de distribution choisie, qui doit contraindre davantage ces types de concession et les rapproche donc sémantiquement. Toute-

fois, il doit être noté que ces trois types sont empiriquement meilleurs que les autres, et ce notamment avec les stratégies WRC-coalitions et WRC-surplus, où les pertes sur le ratio au dernier cœur atteignent moins souvent 20%, et où le prix de la stabilité excède très rarement 10% tout en étant souvent à 0%, c'est-à-dire optimal.

La figure 3 compare le ratio à l'optimal-protocole des trois stratégies sur trois différents types de concession : *faible*, *Nash* et *fort*. Nous choisissons de montrer seulement ces trois types car comme montré sur les figures précédentes, la tendance des courbes sont semblables pour certains types : *faible* et *égoцентриque*, *Nash* et *utilitaire*, *égalitaire*, *fort* et *Pareto*. Les données affichées sont triées de la plus grande à la plus petite

valeur de ratio à l'optimal-protocole avec la stratégie WRC-classic, et les valeurs pour les autres stratégies suivent ce tri en restant liées au jeu correspondant. La figure 3.a montre des valeurs qui varient beaucoup, mais celles-ci décroissent moins et moins souvent pour WRC-coalitions. WRC-surplus semble être la moins bonne stratégie de ce cas. Concernant le type *Nash*, WRC-surplus semble également la pire stratégie, avec des pertes atteignant 60%. Comme précédemment, WRC-coalitions semble être la meilleure. Pour le type *fort*, les valeurs s'effondrent beaucoup moins, avec une perte maximale de 30% environ. Ici, WRC-coalitions et WRC-surplus semblent proches, et presque toujours plus efficaces que WRC-classic.

Pour résumer, nous avons trois types de concession qui minimisent les pertes plus que les autres : *fort*, *égalitaire* et *Pareto*. Concernant les stratégies, WRC-coalitions semble être celle qui présente les meilleurs résultats, suivie de près par WRC-surplus (qui manque cependant d'efficacité sur certains types de concession).

5 Conclusion

Nous avons proposé un protocole distribué pour la formation de coalitions, fondé sur un protocole de négociations monotones, pour lequel nous avons proposé de nouvelles stratégies de concession. Nous avons montré que nos stratégies, et notamment WRC-coalitions, sont plus performantes que la stratégie originelle lorsqu'elles sont associées à certains types de concession qui minimisent les pertes (types *fort*, *égalitaire* et *Pareto*). Cependant, notre approche distribuée nécessite l'exploration d'un grand nombre de structures de coalitions par chaque agent. Il serait donc intéressant de décentraliser totalement le protocole, et d'analyser les pertes d'efficacité dues à cette décentralisation.

Références

- [1] E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4) :1602–1623, 2008.
- [2] E. T. Bell. The iterated exponential integrals. *Annals of Mathematics*, pages 539–557, 1938.
- [3] F. Bistaffa, A. Farinelli, J. Cerquides, J. Rodríguez-Aguilar, and S. D. Ramchurn. Algorithms for graph-constrained coalition formation in the real world. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(4) :1–24, 2017.
- [4] G. Chalkiadakis, E. Elkind, and M. Wooldridge. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6) :1–168, 2011.
- [5] U. Endriss. Monotonic concession protocols for multilateral negotiation. In *Proceedings of the fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 392–399, 2006.
- [6] L. Khalouzadeh, N. Nematbakhsh, and K. Zamani-far. A decentralized coalition formation algorithm among homogeneous agents. *Journal of Theoretical & Applied Information Technology*, 22(1), 2010.
- [7] R. Mochaourab and E. A. Jorswieck. Coalitional games in mimo interference channels : Epsilon-core and coalition structure stable set. *IEEE Transactions on Signal Processing*, 62(24) :6507–6520, 2014.
- [8] T. Rahwan and N. R. Jennings. An improved dynamic programming algorithm for coalition structure generation. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pages 1417–1420, 2008.
- [9] T. Rahwan, T. Michalak, M. Wooldridge, and N. R. Jennings. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artificial Intelligence*, 186 :95–122, 2012.
- [10] T. Rahwan, T. Michalak, M. Wooldridge, and N. R. Jennings. Coalition structure generation : A survey. *Artificial Intelligence*, 229 :139–174, 2015.
- [11] T. Rahwan, S. D. Ramchurn, N. R. Jennings, and A. Giovannucci. An anytime algorithm for optimal coalition structure generation. *Journal of artificial intelligence research*, 34 :521–567, 2009.
- [12] G. Rota. The number of partitions of a set. *The American Mathematical Monthly*, 71 :498–504, 1964.
- [13] L. S. Shapley and M. Shubik. Quasi-cores in a monetary economy with nonconvex preferences. *Econometrica : Journal of the Econometric Society*, pages 805–827, 1966.
- [14] O. Shehory and S. Kraus. Task allocation via coalition formation among autonomous agents. In *IJCAI (1)*, pages 655–661. Citeseer, 1995.
- [15] O. Shehory and S. Kraus. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In *Proc. of ICMAS-96*, pages 330–337. Citeseer, 1996.
- [16] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial intelligence*, 101(1-2) :165–200, 1998.
- [17] M. Sims, C. V. Goldman, and V. Lesser. Self-organization through bottom-up coalition formation. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 867–874, 2003.
- [18] P. Tošić and G. Agha. Maximal clique based distributed group formation for autonomous agent coalitions. *A parametric model for large scale agent systems*, page 195, 2005.
- [19] P. Tošić and C. Ordóñez. Distributed protocols for multi-agent coalition formation : a negotiation perspective. In *International Conference on Active Media Technology*, pages 93–102. Springer, 2012.

Allocation par enchères et planification hiérarchique pour un système multirobot Application au cas de la chasse aux mines

A. Milot^{a,b,c}
antoine.milot@laas.fr

E. Chauveau^b
estelle.chauveau@naval-group.com

S. Lacroix^a
simon.lacroix@laas.fr

C. Lesire^c
charles.lesire@onera.fr

^aLAAS-CNRS, Université de Toulouse, CNRS, 7, Avenue du Colonel Roche, 31031 Toulouse, France

^bNaval Research, Naval Group, 99 Avenue Pierre-Gilles de Gennes, 83190 Ollioules, France

^cONERA/DTIS, Université de Toulouse, 2 avenue Edouard Belin, 31055 Toulouse, France

Résumé

Cet article présente une approche pour accomplir une mission de chasse aux mines avec un système multirobot décentralisé. Afin de prendre efficacement des décisions, l'approche décrite mêle allocation par enchères et planification hiérarchique. Des processus de supervision et de réparation sont également intégrés à l'approche afin de la rendre réactive et robuste à l'occurrence d'événements.

Mots-clés : *Systèmes MultiRobots, Allocation de tâches, Enchères, Planification hiérarchique*

Abstract

This paper presents an approach to handle a minehunting mission with a decentralized multirobot system. To efficiently take decisions, the described framework merges auctions to allocate tasks with hierarchical planning. The approach is responsive and robust to the occurrence of events thanks to supervision and repair processes.

Keywords: *Multi-Robot Systems, Task allocation, Auctions, Hierarchical planning*

1 Introduction

Afin de réduire les coûts, gagner en efficacité, et supprimer les risques humains, de nombreuses marines explorent activement l'utilisation de systèmes multirobots (MultiRobot System - MRS) pour les missions de chasses aux mines sous-marines [1, 2, 7]. En effet, compte tenu des hautes pressions, de l'absence de lumière, et de la présence d'explosif les unités robotiques sont plus adaptées à ces missions que les humains.

Une mission de chasse aux mines est composée de trois objectifs : 1. Détection 2. Identifica-

tion 3. Neutralisation. La mission consiste tout d'abord en un problème d'exploration, car on ne peut pas prévoir la présence et le nombre de mines dans la zone. Cependant, au cours de la mission, il est nécessaire de réagir à divers événements, et en particulier à la découverte d'un objet suspect. Ces événements mettent continuellement à l'épreuve le système qui doit s'adapter afin de les prendre en compte. La mission entraîne donc un fort besoin de *réactivité*.

En raison de leur charge utile limitée, les robots ont des capacités hétérogènes et doivent coopérer pour accomplir la mission. En fonction de ses capacités et de la situation actuelle, chaque robot du MRS peut réaliser différemment un objectif donné : un point essentiel au déploiement d'un MRS est le problème Multi-Robot Task Allocation (MRTA), qui optimise la répartition au sein de la flotte des tâches à réaliser, en fonction de la situation actuelle.

Enfin, les communications jouent un rôle majeur dans le déploiement d'un MRS sous l'eau, milieu dans lequel les communications sont réalisées par le biais d'ondes acoustiques. Bien qu'offrant de grandes portées, le son ne permet pas de communiquer à un débit élevé. De plus, la vitesse du son dans l'eau dépend de paramètres environnementaux tels que la salinité et la température, ce qui entraîne des pertes et des retards dans les messages. Les communications sous-marines constituent un véritable défi pour le MRS et doivent être prises en compte dans la conception du système [3].

Dans le but de surmonter ces challenges, nous proposons une approche basée sur deux concepts clés : l'allocation par enchères et la planification hiérarchique. Ces concepts nous permettent d'entremêler l'allocation et la décomposition, un élément important pour résoudre efficacement le

problème MRTA. Cette approche intègre donc des processus de décision, mais également de supervision et réparation afin de s'adapter aux aléas de la chasse aux mines.

Dans les sections suivantes, nous présentons une vue d'ensemble de l'approche proposée que nous formalisons ensuite. Puis nous discutons de ses avantages et limites avant de résumer les travaux connexes et conclure.

2 Vue d'ensemble de l'approche

Nous proposons une architecture décentralisée reposant sur l'allocation par enchères et la planification hiérarchique [13]. Une vue d'ensemble du schéma de décision est présentée dans la Fig. 1.

Lors de l'allocation de tâches par le biais d'enchères, les robots concernés sont en concurrence les uns avec les autres afin d'acheter des parties de la mission [5]. Un tour d'enchère comporte quatre étapes essentielles : 1. **Annonce** : un commissaire-priseur ouvre le tour en diffusant l'information sur la tâche à vendre ; 2. **Estimation des mises** : chaque enchérisseur estime son coût pour accomplir la tâche et envoie sa mise au commissaire-priseur ; 3. **Détermination du gagnant** : une fois que le commissaire-priseur a reçu toutes les mises, ou qu'une condition d'arrêt est atteinte, il résout le Winner Determination Problem (WDP) pour trouver une allocation ; 4. **Récompense** : si un gagnant a été trouvé à l'étape précédente, le commissaire-priseur l'informe de sa récompense.

Afin de résoudre efficacement les problèmes d'estimation des mises et le WDP, nous nous appuyons sur le concept de Hierarchical Task Network (HTN) et la planification hiérarchique associée [4]. Un HTN intègre des connaissances expertes en décomposant une tâche complexe en des tâches plus simples. Cela permet de réduire la complexité du problème grâce à la spécification de la mission.

2.1 Allocation de tâches par enchères

Les enchères sont un moyen *décentralisé et flexible* d'allouer des tâches. Comme chaque robot peut mettre des tâches en vente, de nouvelles tâches peuvent facilement être ajoutées à la mission en cours. Les enchères ne nécessitent pas de communications parfaites, où tous les membres du système doivent en permanence rester en contact. En cas de message perdu, le

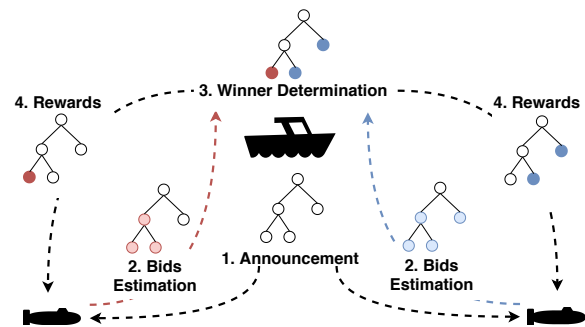


FIGURE 1 – Vue d'ensemble du schéma de décision. Les objectifs de mission décomposés en un HTN sont mis aux enchères. Chaque robot enchère sur les sous-tâches, un processus de détermination du vainqueur attribue les tâches et les envoie aux robots (“rewards”).

processus peut toujours se poursuivre avec les robots connectés restants.

En utilisant les enchères, les robots sont encouragés à coopérer localement. En raison de la distribution géographique des tâches dans une mission de chasse aux mines, cette coopération locale a du sens au niveau de la définition de la mission. Enfin, le passage à l'échelle est facilité par la conception décentralisée du système.

2.2 Intérêt des HTNs au sein d'enchères

Bien que les éléments à vendre dans les enchères soient généralement des tâches uniques, nous utilisons dans notre approche des HTNs. Ce choix est guidé par la complexité réduite des problèmes de planification hiérarchique [8].

La mise aux enchères de HTNs présente d'autres avantages. En vendant des HTNs, on peut facilement allouer des parties entières de la mission. Lors de la mise aux enchères des noeuds d'un HTN, les enchérisseurs ont la possibilité d'acheter directement une branche du HTN. Ce processus permet d'entrelacer allocation et décomposition, un élément clé pour obtenir de meilleures performances tout en résolvant un problème MRTA complexe.

La formulation des mises et la résolution d'une enchère peuvent être des problèmes difficiles, en particulier lorsque l'on considère des tâches dépendantes. Par exemple, il peut être plus intéressant pour un robot d'acheter une tâche à la condition de pouvoir également en acheter une autre. Les HTNs peuvent inclure des dépendances hiérarchiques entre les tâches et permettent donc aux robots d'acheter un ensemble de tâches sans

trop augmenter la complexité du problème de planification.

Cependant, certaines tâches peuvent être directement liées par des contraintes spécifiques à la mission qui ne peuvent pas être représentées dans un HTN classique. Par exemple, en fonction du degré de retour d'information requis par l'opérateur, une tâche d'identification peut devoir nécessiter un flux de données en direct. Dans ce cas, compte tenu des contraintes de communication, la découverte d'un objet suspect entraîne la création de deux tâches qui doivent être exécutées simultanément par deux robots : une tâche *identification* et une tâche *relais de communication*. Les deux tâches sont alors liées par une contrainte de synchronisation qui doit être prise en compte par le processus d'allocation. De telles contraintes compliquent le problème MRTA, mais peuvent cependant être traitées grâce au formalisme HTN, dans lequel des contraintes spécifiques comme celle-ci peuvent être considérées.

2.3 Connaissances Multirobot vs. locales

Nous exploitons les HTNs à deux niveaux, ils sont soit locaux à un robot et décrivant ses capacités, soit basés sur la spécification de la mission et connus de tous les agents. Cette dualité entre le niveau *multirobot* (tâches à allouer) et le niveau *local* (comment accomplir les tâches) est un point clé de notre approche. Le protocole résultant est décrit dans la Fig. 2.

Notre système a besoin en entrée d'un HTN initial, appelé *Multi-Robot Graph*, décrivant les tâches à vendre. Au commencement, la mission est un problème d'exploration consistant en une tâche *couvrir la zone de mission*. Cette zone pouvant être divisée, par exemple, en sous-zones *A* et *B* (pouvant elles-mêmes être divisées et ainsi de suite), l'exploration de la *zone de mission* peut être accomplie en explorant *A* et *B*. Cette partition de la *zone de mission* peut être établie par un opérateur humain en s'appuyant sur la connaissance de la zone et les paramètres des capteurs utilisés pour la détection. La partition est ensuite encapsulée dans le *Multi-Robot Graph* initial.

Une fois le *Multi-Robot Graph* défini, le commissaire-priseur le transmet aux enchérisseurs. Les enchérisseurs utilisent ensuite leurs connaissances *locales* pour estimer les mises. Enfin, le commissaire-priseur insère les mises reçues dans un HTN décrivant le WDP et le résout pour trouver une allocation. Lors de la construction du WDP, le commissaire-priseur

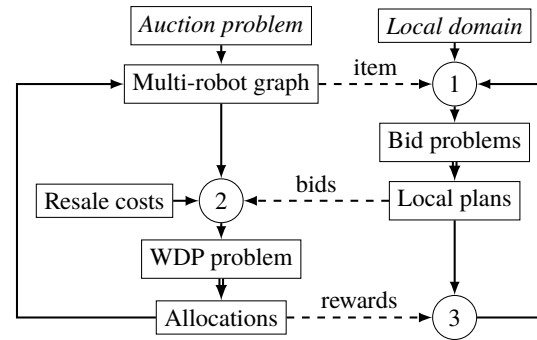


FIGURE 2 – Description du protocole. Les rectangles représentent des informations ou des structures de données gérées par les agents. Les blocs de gauche sont gérés par le commissaire-priseur, les blocs de droite par chaque enchérisseur. Les flèches en pointillés représentent les données échangées entre le commissaire-priseur et les enchérisseurs. Les cercles indiquent les processus qui agrègent les informations pour construire de nouvelles structures. Les doubles flèches indiquent les appels à un planificateur HTN.

intègre également un processus de revente. Selon la façon dont ce processus de revente est conçu, on peut contrôler le comportement d'allocation. Par exemple, nous pouvons implémenter des préférences sur l'urgence de la vente d'une tâche.

Dans des travaux antérieurs [12], nous avons testé une preuve de concept de notre architecture de décision pour résoudre une seule enchère. Nous avons obtenu des résultats prometteurs en termes de qualité de la solution, de durée et de contrôle de l'allocation.

2.4 Utilisation en ligne

Lorsque le système évolue dans un environnement complexe et contraint, le besoin de réactivité va au-delà de la seule capacité à intégrer de nouvelles tâches. En raison d'incertitudes et d'événements imprévus, un robot peut en effet ne pas être en mesure d'accomplir une tâche qu'il a précédemment achetée.

Par exemple, une tâche *cover* peut avoir été achetée pour un certain coût (e.g. une durée). Mais pendant l'exécution, des événements imprévus tels que des courants plus forts qu'attendu, peuvent avoir un impact sur le coût réel de la tâche : le plan engagé pendant l'allocation de la tâche n'est plus valide et une action de réparation est nécessaire. L'approche proposée, qui

entrelace enchères et planification hiérarchique, peut gérer de tels événements, en permettant au robot, entre autres, de revendre la tâche qu'il ne peut pas réaliser correctement.

Un autre besoin, intrinsèque à la nature même de la mission, est de pouvoir intégrer de nouveaux objectifs durant la mission. En effet, lors d'une tâche *cover* un agent peut détecter un objet suspect. Cet objet doit alors être identifié afin de déterminer s'il s'agit d'une mine ou non. Cette situation mène donc à la création d'une nouvelle tâche d'identification (ou *identify*). Cependant, par souci de charge utile les agents dédiés aux tâches de couverture ne peuvent identifier ces objets. Le robot ayant détecté l'objet va donc ouvrir une nouvelle enchère en tant que commissaire-priseur afin d'allouer ce nouvel objectif d'identification.

Ces processus spécifiques à la détection et la réparation de problèmes confèrent au système une certaine robustesse face aux imprévus.

3 Formalisation de l'approche

Dans cette section nous détaillons et formalisons les étapes de notre protocole d'enchères.

3.1 Notations

Chaque enchère correspond à un problème MRTA à résoudre. Étant donné $R = (r_1, \dots, r_n)$ un ensemble de n robots et un ensemble $Q = (t_1, \dots, t_s)$ de s tâches, résoudre le problème MRTA consiste à trouver une allocation $A : Q \rightarrow R$, i.e. allouer chaque tâche $t \in Q$ à un robot $r \in R$.

Dans ce document, nous utilisons les notations suivantes, inspirées de celles utilisées par [6, 9]. Par \mathcal{L} , nous désignons un langage du premier ordre composé d'ensembles finis de constantes, de prédicats, de symboles de tâches primitives et complexes, d'un ensemble infini de symboles de variables et d'un ensemble infini de labels désignés par \mathbb{L} . Étant donné un ensemble de termes x_1, \dots, x_k issus de ce langage, et s un symbole de tâche, nous désignons par $t = s(x_1, \dots, x_k)$ une tâche (aussi appelée *instance de tâche*). Une tâche complexe t peut être décomposée par une méthode $m = (t, tn)$, où $tn = (L, \prec, \alpha)$ est un réseau de tâches où $L \subset \mathbb{L}$ est un ensemble de labels, \prec est un ordre partiel strict sur L et $\alpha : L \rightarrow X$ établi une correspondance vers les sous-tâches de la méthode. Nous désignons ensuite un domaine de planification \mathcal{D}

par $(\mathcal{L}, T_P, T_C, M)$ où \mathcal{L} est le langage sous-jacent, T_P et T_C sont les ensembles de tâches primitives et complexes, et M l'ensemble des méthodes de décomposition. Enfin, nous définissons un problème de planification comme suit $\mathcal{P} = (\mathcal{D}, s_I, tn_I)$ où s_I est l'état initial, et tn_I est le réseau de tâches initial.

Résoudre un problème $\mathcal{P} = (\mathcal{D}, s_I, tn_I)$ consiste à trouver un *réseau de tâches solution* tn tel que tn est *primitif* et *exécutable* dans s_I , i.e. qu'il existe une séquence des tâches de tn , qui respecte les contraintes d'ordonnement, dans laquelle les préconditions d'une tâche sont valides dans l'état résultant de l'application de la tâche précédente.

3.2 Exemple d'illustration

Pour illustrer notre approche, nous considérons une mission de type *Home Port Survey* dans la rade de Brest en France. L'objectif est de couvrir la rade afin de s'assurer, avec un certain niveau de confiance, qu'aucune mine n'est présente (ou qu'elles ont toutes été localisées). Comme il s'agit d'une mission d'inspection d'une zone allée, il n'y a pas de contraintes de temps ou de discrétion.

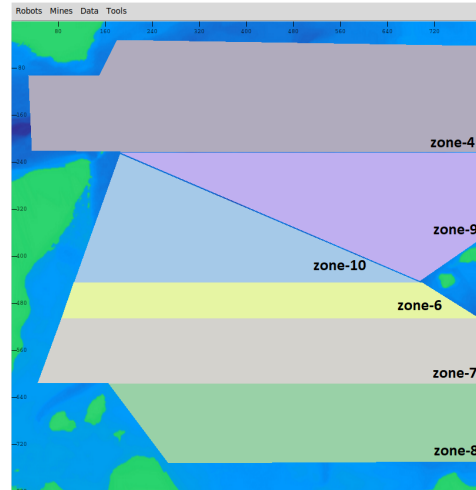


FIGURE 3 – La partition de l'ensemble de la zone de mission.

En entrée de notre système, nous considérons deux types d'information : 1. Une description des objectifs initiaux de la mission (appelés tâches multirobots); 2. Une description des capacités locales de chaque enchérisseur (i.e. comment l'enchérisseur peut accomplir une tâche multirobot). Dans notre approche, ces informations sont définies en utilisant un langage de

modélisation appelé Hierarchical Domain Definition Language (HDDL) [9].

Étant donné que la mission consiste initialement en un problème d’exploration, les objectifs initiaux sont uniquement des tâches de couvertures. Ces tâches sont basées sur une partition de la zone de mission, illustrée Fig. 7.

Tout au long de cette section nous nous référons à ce problème à des fins d’exemple.

3.3 Initialisation d’une enchère

Notre approche vise à résoudre un problème MRTA défini par un problème d’enchère $\mathcal{P}_{auc} = (\mathcal{D}_{auc}, s_I, tn_I)$ formulé par le commissaire-priseur. \mathcal{D}_{auc} est un domaine décrivant la décomposition haut-niveau des tâches à l’échelle multirobot. En effet, ce domaine décompose les tâches qui peuvent être allouées, i.e. *quoi*, sans prendre en considération *comment* un robot va accomplir les tâches en termes d’actions spécifiques (telles que *goto*, *dive*, *use-sas-sonar*...). L’état initial s_I est déterminé par le commissaire-priseur et tn_I est le réseau de tâches initial du problème. Nous considérons que tous les effets multirobots sont inclus dans la description du problème et que les actions locales n’ont pas d’impact sur eux.

Résoudre \mathcal{P}_{auc} signifie résoudre le problème MRTA associé à cette enchère. Cependant, comme le schéma de décision est décentralisé, lorsque les robots enchérissent sur une tâche, nous devons déterminer à quelle instance de cette tâche dans le plan multirobot final cette mise est associée. Par conséquent, nous ne pouvons pas raisonner uniquement sur \mathcal{P}_{auc} et nous devons identifier chaque instance de tâche.

Pour ce faire, nous utilisons \mathcal{P}_{auc} comme entrée du processus d’enchère. À partir de ce problème d’enchère, nous construisons le *Multi-Robot Graph* (désigné comme \mathcal{H}), nous détaillons sa construction dans [13]. Le *Multi-Robot Graph* décrit la décomposition hiérarchique des tâches, tandis que les labels leur confèrent une unicité en les identifiant. Pendant un tour d’enchère, les robots utilisent les labels pour partager des informations relatives au *Multi-Robot Graph*. La Fig. 4 représente le *Multi-Robot Graph* résultant pour le problème *Home Port Survey*.

À chaque tour, le commissaire-priseur envoie \mathcal{H} aux enchérisseurs avec des informations complémentaires provenant du problème d’enchère. Un *article à vendre* δ est alors défini comme un tuple

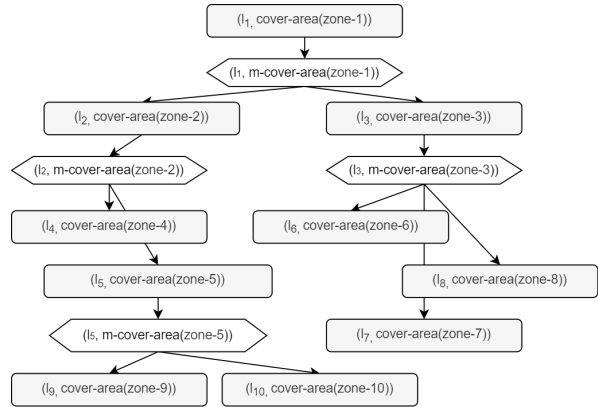


FIGURE 4 – *Multi-Robot Graph* initial pour un problème *Home Port Survey*. Les rectangles arrondis et les hexagones représentent respectivement des tâches et des méthodes labélisées.

$(\mathcal{H}_\delta, s_\delta, L_\delta)$, où \mathcal{H}_δ est le *Multi-Robot Graph* représentant la décomposition hiérarchique entre les tâches labélisées et les contraintes de précedence associées (incluses dans les méthodes de décomposition); s_δ est un ensemble de formules atomiques sur les constantes de \mathcal{H}_δ ; L_δ est l’ensemble des labels de tâches dans \mathcal{H}_δ qui sont *vendables*, i.e. sur lesquelles les robots peuvent produire des mises.

Par exemple, au premier tour du problème *Home Port Survey*, le commissaire-priseur envoie δ_1 avec \mathcal{H}_{δ_1} (illustré sur la Fig. 4) et $L_{\delta_1} = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}\}$.

3.4 Estimation des mises

Une fois qu’un article à vendre δ est reçu, l’enchérisseur doit calculer une mise pour chaque tâche labélisée réalisable parmi L_δ , i.e. chaque tâche exécutable par l’enchérisseur. La valeur de la mise correspond au coût d’exécution de cette tâche. Par conséquent, pour chaque $l \in L_\delta$, nous construisons un problème de planification $\mathcal{P}_l = (\mathcal{D}_l, s_l, tn_l)$ et utilisons un planificateur HTN pour le résoudre.

Pour estimer cette mise, les robots doivent indiquer *comment* ils comptent effectuer les tâches primitives de \mathcal{H}_δ . Ils peuvent en effet devoir effectuer des actions spécifiques, comme se déplacer jusqu’à l’emplacement de la tâche, activer des capteurs ou des actionneurs particuliers, etc. Nous considérons que les descriptions des tâches spécifiques à chaque robot sont définies dans un *problème local*. Ce problème local a les contraintes suivantes : tout d’abord, les tâches que le robot peut décomposer localement doivent

être les mêmes que celles du problème d'enchère (i.e. partager les mêmes symboles et constantes). Deuxièmement, nous considérons raisonnablement que les problèmes locaux et multirobots ne partagent aucun prédicat. Ainsi, une tâche multirobot dans le problème d'enchère ne peut pas dépendre d'un prédicat qui ne pourrait être validé que par l'action locale spécifique d'un robot. Inversement, les actions locales d'un robot ne peuvent pas dépendre des effets des tâches multirobots.

L'agrégation de l'article à vendre et du problème local correspond à l'étape ① de la Fig. 2. Elle consiste, pour chaque label $l_\delta \in L_\delta$ sur lequel l'enchérisseur veut produire une mise, à étendre \mathcal{H}_δ de la manière suivante : chaque tâche feuille l de \mathcal{H}_δ qui est soit une descendante de l_δ , soit qui a déjà été allouée à l'enchérisseur lors des tours précédents, est remplacée par une tâche complexe avec exactement une méthode ordonnée, composée (dans l'ordre) de (1) une action $start(l)$ représentant le début de l , et contenant seulement les préconditions de l , (2) une tâche complexe qui sera décomposée dans le problème local de l'enchérisseur, et (3) une action $end(l)$ représentant la fin de l , et contenant seulement les effets de l . Les autres tâches primitives sont comptabilisées à un coût de 0. Essentiellement, $start(l)$ et $end(l)$ permettent d'insérer les actions locales du robot entre les préconditions et les effets de la tâche l qui ont été définis dans \mathcal{H} .

La Fig. 5 illustre ce processus en montrant l'extension de la tâche labélisée l_4 du problème *Home Port Survey*. Les tâches en jaune correspondent à la décomposition introduite précédemment. Les tâches en orange correspondent au TDG construit avec le domaine local d'un enchérisseur. Par exemple, résoudre ce problème d'estimation de mise peut conduire à une séquence d'actions telle que $[start(l_4) \rightarrow goto(zone-4) \rightarrow \dots \rightarrow end(l_4)]$.

De plus, l'enchérisseur construit s_l en fusionnant s_δ avec son propre état initial qui inclue ses propres informations locales (par exemple, sa position actuelle, son énergie. . .). tn_l est défini par la tâche racine de \mathcal{H}_δ .

Enfin, pour chaque tâche labélisée sur laquelle nous voulons estimer une enchère, nous construisons un domaine et un problème (incluant la décomposition mentionnée précédemment, et les états initiaux provenant à la fois de l'objet à vendre et du problème local). Nous nous appuyons sur le formalisme HDDL pour définir ces domaine et problème. Dans ce problème, les

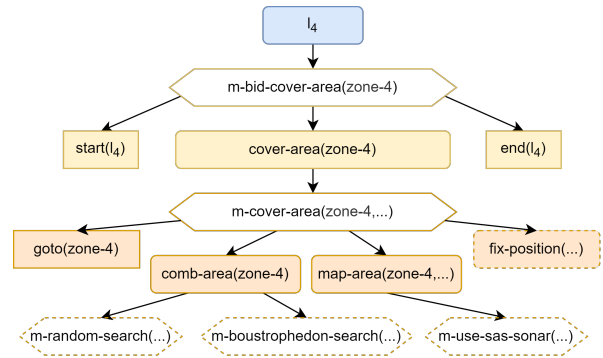


FIGURE 5 – Illustration de la décomposition de la tâche étiquetée l_4 pour l'estimation de la mise du problème *Home Port Survey*. Les hexagones représentent les méthodes, les rectangles arrondis représentent les tâches complexes tandis que les rectangles pointus représentent les tâches primitives, les lignes pointillées représentent les noeuds avec des décompositions cachées. Les noeuds bleus représentent des éléments de \mathcal{H} . La décomposition spécifique pour l'estimation de la mise est représentée en jaune. Les noeuds de décomposition locale sont représentés en orange.

actions primitives sont soit les actions $start$ et end introduites précédemment, qui ont un coût nul, soit les actions locales de l'enchérisseur, qui peuvent avoir un coût non unitaire. Un planificateur HTN résout alors ce problème, en produisant à la fois un plan local pour réaliser la tâche et un coût associé à ce plan. Les actions $start$ et end dans le plan permettent de vérifier les préconditions et effets de la tâche.

Pour finir, l'enchérisseur retourne au commissaire-priseur un ensemble B contenant des triplets (l, c, P) , où $l \in L_\delta$ est un label de tâche, c la valeur de la mise associée, i.e. le coût renvoyé par le planificateur, et P le plan de l'enchérisseur sur les tâches multirobots (i.e. composés des labels de tâches de \mathcal{H}_δ).

3.5 Résoudre le WDP

À la fin d'un tour d'enchères, le commissaire-priseur résout le WDP pour déterminer les allocations de tâches. Étant donné L_δ l'ensemble des labels des tâches à vendre et $R = (r_1, \dots, r_n)$ l'ensemble des enchérisseurs participant à l'enchère, nous notons B_{r_i} l'ensemble des mises reçues de l'enchérisseur r_i portant sur des labels dans L_δ . L'ensemble de toutes les mises reçues est désigné par $\mathcal{B} = \bigcup_{r_i \in R} B_{r_i}$. La résolution du

WDP consiste à trouver un ensemble de mises

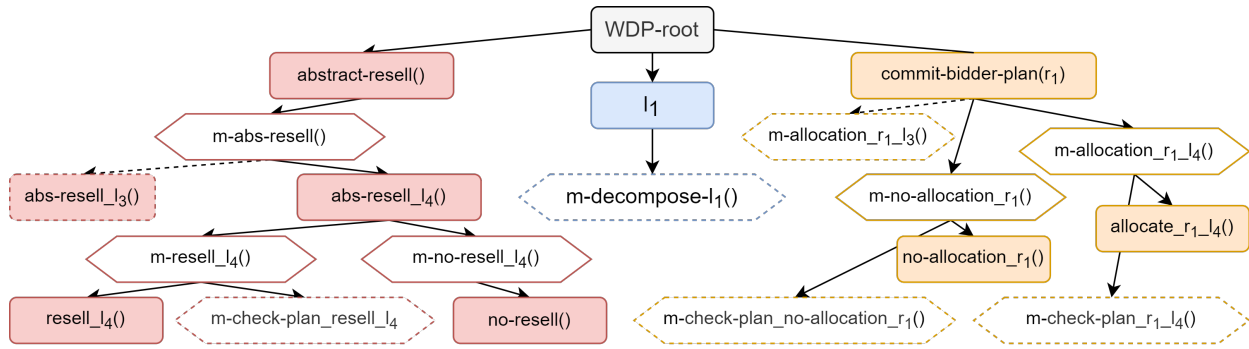


FIGURE 6 – Extrait du HTN correspondant au WDP d’un tour d’enchère du problème *Home Port Survey*. Les hexagones représentent les méthodes, les rectangles arrondis représentent les tâches complexes tandis que les lignes pointillées représentent les noeuds avec des décompositions cachées. Les parties rouges, bleues et oranges correspondent respectivement au processus de revente, aux informations du *Multi-Robot Graph*, et à l’encodage de l’enchérisseur r_1 .

gagnantes $\mathcal{B}_w \subset \mathcal{B}$ tel que chaque enchérisseur gagne au plus une mise et que chaque $l \in L_\delta$ soit gagné par au plus un enchérisseur.

Afin de déterminer cet ensemble de mises gagnantes, le commissaire-priseur doit construire \mathcal{D}_{wdp} et $\mathcal{P}_{wdp} = (\mathcal{D}_{wdp}, s_{wdp}, tn_{wdp})$ un domaine et un problème de planification dédiés à la résolution du WDP.

Pour ce faire, nous agrégeons les informations de \mathcal{H}_δ avec les mises reçues (étape ② dans la Fig. 2). Afin de supporter les problèmes partiellement ordonnés contenant des tâches liées par différentes contraintes (e.g. des contraintes causales), il est important de prendre en compte les intentions des enchérisseurs sur les tâches multirobots. Sans cela, la résolution du WDP pourrait mener à des incohérences et des impossibilités d’exécutions, nous détaillons ces problèmes dans [13]. Par conséquent, avec chaque mise est envoyé le plan multirobot déterminé par l’enchérisseur au cours du processus d’estimation. Ce plan intègre : les tâches multirobots que l’enchérisseur compte accomplir (soit déjà acquises, soit sur lesquelles il mise) ; les tâches multirobots que l’enchérisseur souhaite que d’autres accomplissent (car nécessaires à ses objectifs).

Dans le but d’assurer la cohérence de l’allocation, il est donc nécessaire d’encoder dans le WDP les plans des enchérisseurs. Pour ce faire, nous associons aux tâches du *Multi-Robot Graph* des prédicats dédiés à la vérification des plans et, pour chaque enchérisseur, nous construisons un nouvel HTN dédié à l’encodage de ses plans (i.e. un pour chaque mise et un, le cas échéant, pour le plan validé au tour précédent). Grâce à ces nouveaux prédicats et HTNs nous assurons

la cohérence des allocations entre elles.

De plus, nous devons également intégrer la contrainte d’indépendance des mises du schéma SSI qui impose qu’au plus une tâche (quel que soit son niveau d’abstraction) puisse être allouée à chaque robot. L’encodage de cette propriété est fait grâce à un mécanisme autorisant la revente d’une tâche à un tour ultérieur. Ainsi, à chaque tâche vendable est associée une action de *revente* (ou *resell*) correspondant à la non-attribution de cette tâche. Similairement au HTN dédié à l’encodage du plan d’un enchérisseur, nous créons un arbre dédié à la revente des tâches.

Ainsi, tn_{wdp} est défini par : la tâche racine de \mathcal{H}_δ ; la tâche racine du HTN de revente ; chaque tâche racine des HTNs dédiés aux enchérisseurs. Et l’état initial s_{wdp} est le même que s_δ . La Fig. 6 illustre le HTN résultant du processus de construction du WDP en incluant les possibilités d’allocation de l_4 au robot r_1 et de revente.

Comme nous ne pouvons pas forcer le WDP à allouer exactement une tâche à chaque robot, nous ne pouvons pas fixer le coût des actions *resell* à 0 : ces tâches seraient systématiquement revendues lors de l’optimisation du plan d’allocations. Par conséquent, nous devons définir des *coûts de revente* (ou *resale costs*) pour avoir un comportement d’allocation sain et efficace. Bien que ces coûts puissent être définis en fonction du domaine d’application, nous avons proposé dans [12] deux stratégies génériques pour les définir, basées sur les mises reçues pour une tâche.

En résumé, le commissaire-priseur crée, à partir de \mathcal{H}_δ et des mises reçues, un nouveau problème de planification \mathcal{P}_{wdp} correspondant au WDP

de ce tour d'enchères. Ce problème, traduit en un domaine et un problème HDDL, est ensuite résolu par un planificateur HTN qui retourne un plan contenant soit une action d'allocation, soit une action de revente pour chaque tâche. Le commissaire-priseur envoie ensuite des messages de récompense avec les mises gagnantes pour les tâches allouées. Les gagnants gardent la trace de leur récompense et s'engagent à faire le plan associé à la mise gagnante (processus ③ dans la Fig. 2). Enfin, le commissaire-priseur intègre les plans joints aux mises gagnantes dans le \mathcal{H}_δ et un nouveau tour débute avec les tâches à revendre.

3.6 Illustration

Pour illustrer la finalité de notre approche, nous considérons le déploiement d'un MRS sur la mission de *Home Port Survey*. Ce MRS est composé de trois *explorers* (appelés *AUVE1*, *AUVE2* et *AUVE3*) et d'un *identifieur* (appelé *AUVi1*). Ces deux types de robots sont respectivement responsables des tâches de couverture et d'identification.

Les objectifs de couverture de la zone sont mis en vente au cours d'une première enchère tel que décrit dans la Sec. 3.3. Considérons les résultats d'allocation du Tab. 1 pour la première enchère se déroulant sur trois tours. À la fin du premier tour, il est implicite que la résolution du WDP a mené à la revente de $\{l_5, l_6, l_8, l_9, l_{10}\}$. Pour débuter le tour suivant, le commissaire-priseur envoie donc un nouvel article à vendre composé de : \mathcal{H}_{δ_2} , i.e. \mathcal{H}_{δ_1} mis à jour avec les choix d'allocation de l_4 et l_7 ; et $L_{\delta_2} = \{l_5, l_6, l_8, l_9, l_{10}\}$.

Au terme de cette première enchère, *AUVE1*, *AUVE2* et *AUVE3* ont acquis des tâches multirobots en s'engageant sur les plans associés. Par exemple, le plan multirobot de *AUVE3* à la fin de cette enchère est la séquence l_6, l_7, l_8 .

Robots	Tour 1	Tour 2	Tour 3
<i>AUVE1</i>	l_4		
<i>AUVE2</i>		l_5	
<i>AUVE3</i>	l_7	l_6	l_8

TABLE 1 – Allocations de la première enchère. Les labels des tâches correspondent à ceux de la Fig. 4.

Par la suite nous considérons qu'un objet suspect a été détecté par *AUVE2* qui débute alors

1. Cette étape, qui supprime également la décomposition obsolète de \mathcal{H} , est assez simple, et n'est donc pas détaillée.

une seconde enchère pour allouer un nouvel objectif d'identification tel que décrit dans la Sec. 2.4. Cet objectif est acquis par *AUVi1*. Enfin, considérons que, durant l'exécution de son plan, *AUVE3* rencontre une panne de capteur qui l'empêche de terminer l_8 . Similairement à l'intégration d'un nouvel objectif, il devient commissaire-priseur d'une troisième enchère ayant pour but la revente sa tâche l_8 . Cette tâche est réallouée à *AUVE2*. La Fig. 7 illustre l'état final du système pour cet exemple. Tous les robots ont accompli leurs tâches multirobots et la mission est terminée.

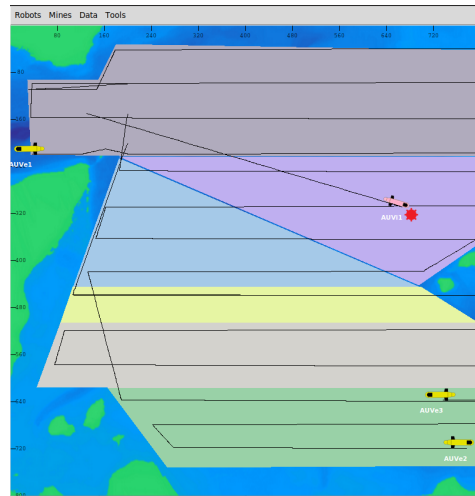


FIGURE 7 – Illustration du système dans son état final. Toutes les zones ont été couvertes et une mine a été identifiée.

4 Discussion

Dans cet section, nous résumons les atouts de l'approche et ses limites.

4.1 Avantages de l'approche

Par l'utilisation de HTNs au sein d'enchères notre approche permet d'intriquer allocation et décomposition, un des éléments primordiaux pour résoudre des problèmes MRTA difficiles. De plus, les problèmes de planification résultants bénéficient de l'usage de planificateur hiérarchique pour les résoudre efficacement [12].

L'approche supporte dans sa version actuelle des problèmes partiellement ordonnés incluant des contraintes causales. Elle intègre également des processus de supervision et réparation permettant de réagir aux aléas d'une mission de chasse aux mines, tels que l'apparition d'un nouvel objectif ou la réallocation d'anciens [14].

En outre, l'approche proposée est flexible et générique. La modélisation de la mission pour générer les entrées nécessaires en est la clé de voûte. Or, le formalisme HTN facilite grandement cette étape de modélisation, et permet d'ajouter aisément de nouvelles contraintes.

Nous pouvons également tirer parti des paramètres d'enchères pour obtenir du système différents comportements. Dans notre travail, nous utilisons un coût comme valeur de la mise. C'est la façon la plus basique de travailler avec les enchères, cependant nous pourrions utiliser des informations plus descriptives pour définir la valeur de l'enchère. Il existe tout un corpus de la littérature sur les enchères qui se concentre sur comment définir la valeur d'une mise [5]. Par exemple, nous pourrions utiliser *à quel point l'enchérisseur estime bien accomplir la tâche*.

Le processus de revente est également un moyen pratique de contrôler l'allocation. En fonction de sa conception, on peut par exemple optimiser la qualité de la solution ou minimiser le nombre de tours nécessaires. Nous renvoyons le lecteur à [12] pour des informations plus complètes sur la façon dont cela est accompli.

Grâce à ces avantages, le système peut être adapté à l'évolution rapide des doctrines dans le domaine de la chasse aux mines. On peut également l'utiliser sur des missions très différentes : dans [13] nous avons testé notre approche sur une mission plus classique de logistique.

4.2 Perspectives et limites

Bien que l'implémentation actuelle de notre approche intègre déjà la capacité d'allouer des tâches de manière décentralisée et de gérer la réparation de plans, il s'agit toujours d'un travail en cours d'évolution. Il reste par conséquent quelques défis à relever.

Le plus important est d'améliorer la description existante de la mission pour qu'elle corresponde à un cas d'utilisation plus réaliste. Actuellement, notre approche repose en effet sur une exécution séquentielle des tâches, alors que certaines tâches nécessitent des contraintes autres que des contraintes de précédence ou de causalité. Par exemple, la tâche *relais de communication* devra être exécutée dans un intervalle de temps spécifique. Afin d'implémenter ces nouvelles tâches, nous devons améliorer l'approche en intégrant des contraintes temporelles, ce que nous comptons faire à l'aide du formalisme HTN. Cela permettra d'ajouter de nouvelles tâches pour sup-

porter les relais de communication, la recharge des AUVs, ou encore l'utilisation d'une mule de données. Un autre avantage sera de faire une meilleure utilisation des robots souvent inactifs, tels que les *identifiants*, afin d'accomplir des tâches de soutien.

Notre approche comporte cependant certaines limites inhérentes à sa conception. Bien que guidé par les contraintes de l'environnement sous-marin, le choix d'une architecture décentralisée établit d'office un compromis entre la robustesse du système et la qualité des solutions. Dans les enchères, la qualité de la solution d'allocation dépend principalement de la diversité des enchérisseurs capables de communiquer avec le commissaire-priseur. Comme notre système est destiné à être utilisé dans des environnements de communication difficiles, nous comptons étudier l'impact de la qualité des communications sur les performances de la mission.

Par ailleurs, la complexité du WDP est un écueil habituel des enchères. Bien qu'elle soit réduite par l'utilisation de la planification HTN, nous devons encore étudier soigneusement les variables qui l'affectent le plus.

5 Travaux connexes

Enchères sur des HTN Les fondamentaux des schémas d'enchères sur des HTNs ont été posés dans [17]. Ce type d'approche a également été étendue pour permettre aux enchérisseurs d'acheter puis de revendre des tâches en proposant éventuellement une nouvelle décomposition [11, 10]. Cependant, dans ces approches, le WDP est résolu par un algorithme cupide couplé à un parcours en largeur de l'arbre. Les allocations de tâches sont décidées localement, sans raisonner sur l'arbre global de décomposition des tâches. Bien qu'il s'agisse d'un algorithme efficace en termes de temps de calcul, la qualité des allocations est discutable. Dans [12] nous avons comparé notre approche à cet algorithme et l'avons globalement surpassé. De plus, ces approches ne considèrent pas les contraintes temporelles et causales entre les tâches, et en raison de l'algorithme spécifique du WDP, ces contraintes ne peuvent pas être facilement intégrées.

Enchères sur des tâches avec contraintes temporelles Un schéma d'enchère séquentielle pour des tâches comprenant des contraintes temporelles a été proposé dans [15]. Le commissaire-priseur y maintient un réseau temporel (Simple

Temporal Network - STN) des éléments en vente. Cependant, cette approche est utilisée pour l'allocation des tâches d'un STN déjà précalculé, tandis que dans notre approche nous visons à résoudre à la fois l'allocation et la sélection des tâches à effectuer afin d'accomplir la mission.

Enchères et contraintes de communication

L'impact de l'instabilité des communications sur différentes approches basées enchères et les mesures possibles pour limiter cet impact ont été étudiés dans [16]. La qualité d'allocation du WDP étant intrinsèquement liée à la diversité des mises à considérer, l'influence de la qualité des communications n'est pas à négliger. L'utilisation d'accusés de réception est proposée comme un moyen simple de réduire la perte d'information, les messages d'enchères étant peu volumineux cette solution simple semble applicable au monde sous-marin sans saturer la bande passante. Néanmoins en s'appuyant sur notre approche, il apparaît également intéressant d'explorer un design du coût de revente incluant la qualité des communications. Cela permettrait de faire émerger des tendances d'allocation optimistes ou pessimistes qui soient fonction du réseau de communication du commissaire-priseur à un instant donné.

6 Conclusion

Dans cet article, nous avons présenté formellement une approche pour accomplir une mission de chasse aux mines avec un système multirobot autonome. L'approche repose sur des allocations par enchères et la planification hiérarchique et encapsule des processus de décision, de supervision, et de réparation apportant réactivité et robustesse au système. Puis nous avons résumé les avantages et limites de l'approche.

Les travaux futurs consisteront à améliorer la modélisation de la mission et à démontrer l'applicabilité de l'approche sur des problèmes plus réalistes intégrant des contraintes temporelles telles que des intervalles de temps. Nous étudierons aussi attentivement l'impact de la qualité des communications sur les performances du système et les moyens possibles pour en limiter les conséquences.

Références

- [1] Next-generation MCM systems for the Belgian and Netherlands Navies. *Mer et Marine*, 2019.
- [2] George Allison. Uncrewed minehunters to replace conventional vessels. *UK Defence Journal*, 2020.
- [3] Francesco Amigoni, Jacopo Banfi, and Nicola Basilico. Multirobot exploration of communication-restricted environments : A survey. *IEEE Intelligent Systems*, 2017.
- [4] Pascal Bercher, Ron Alford, and D. Höller. A Survey on Hierarchical Planning - One Abstract Idea, Many Concrete Realizations. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Macao, China, 2019.
- [5] M.B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-Based Multirobot Coordination : A Survey and Analysis. *Proceedings of the IEEE*, 2006.
- [6] Kutluhan Erol, James Hendler, and Dana S Nau. HTN planning : Complexity and expressivity. In *AAAI Conference on Artificial Intelligence (AAAI)*, Seattle, WA, USA, 1994.
- [7] Gabriele Ferri, Andrea Munafò, Alessandra Tesei, Paolo Braca, Florian Meyer, Konstantinos Pelekankakis, Roberto Petrocchia, João Alves, Christopher Strode, and Kevin LePage. Cooperative robotic networks for underwater surveillance : an overview. *IET Radar, Sonar & Navigation*, 2017.
- [8] Ilche Georgievski and Marco Aiello. HTN planning : Overview, comparison, and beyond. *Artif. Intell.*, 2015.
- [9] Daniel Höller, Gregor Behnke, Pascal Bercher, Susanne Biundo, Humbert Fiorino, Damien Pellier, and Ron Alford. HDDL : An Extension to PDDL for Expressing Hierarchical Planning Problems. In *AAAI Conference on Artificial Intelligence (AAAI)*, New York City, NY, USA, 2020.
- [10] Alaa M. Khamis, Ahmed M. Elmogy, and Fakhri O. Karray. Complex Task Allocation in Mobile Surveillance Systems. *International Journal on Intelligent and Robotic Systems*, 64(1), 2011.
- [11] Yabo Liu, Jianhua Yang, Yao Zheng, Zhaohui Wu, and Min Yao. Multi-Robot Coordination in Complex Environment with Task and Communication Constraints. *International Journal of Advanced Robotic Systems*, 10(5), 2013.
- [12] Antoine Milot, Estelle Chauveau, Simon Lacroix, and Charles Lesire. Market-based Multi-robot coordination with HTN planning. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021.
- [13] Antoine Milot, Estelle Chauveau, Simon Lacroix, and Charles Lesire. Solving Hierarchical Auctions with HTN Planning. In *4th ICAPS workshop on Hierarchical Planning (HPlan)*, 2021.
- [14] Antoine Milot, Estelle Chauveau, Simon Lacroix, Charles Lesire, and Loïc Tacher. Autonomous and responsive multi-robot system for minehunting. In *OCEANS 2021 Chennai*, 2022.
- [15] Ernesto Nunes and Maria Gini. Multi-robot auctions for allocation of tasks with temporal constraints. In *AAAI Conference on Artificial Intelligence (AAAI)*, Austin, TX, USA, 2015.
- [16] Michael Otte, Michael J. Kuhlman, and Donald Sofge. Auctions for multi-robot task allocation in communication limited environments. *Autonomous Robots*, 2019.
- [17] Robert Zlot and Anthony Stentz. Market-based Multirobot Coordination for Complex Tasks. *The International Journal on Robotics Research*, 25(1), 2006.

Session 3 : Communication

Spécificités de l'argumentation scientifique dans un débat

Louise Dupuis de Tarlé^a
louise.dupuis@dauphine.eu

Gabriella Pigozzi^a
gabriella.pigozzi@lamsade.dauphine.fr

Juliette Rouchier^a
juliette.rouchier@lamsade.dauphine.fr

^aUniversité Paris-Dauphine, CNRS, LAMSADE, Place du Maréchal de Lattre de Tassigny, 75016 Paris, France

Résumé

Depuis l'introduction des Abstract Argumentation Frameworks (cadres d'argumentation abstraits) par [7], de nombreux jeux d'argumentation basés sur des agents ont été développés pour étudier les débats. La question de la génération collective de connaissances a été étudiée par des modèles tels que ceux de [18]. Notre objectif est d'étudier une communauté épistémique d'agents qui construit un corpus de connaissances communes en générant et en échangeant des arguments. Nous caractérisons les conditions sous lesquelles ces agents convergent vers un résultat cohérent avec leur environnement, divergent ou se polarisent.

Mots-clés : Argumentation, Diffusion d'Opinion, Multi-Agent, Epistémologie Sociale

Abstract

Since the introduction of Abstract Argumentation Frameworks by [7], many agent-based argumentation games have been developed to study debates. The question of the collective generation of knowledge has been studied by models such as those of [18]. Our goal is to study an epistemic community of agents that builds a body of common knowledge by generating and exchanging arguments. We characterize the conditions under which these agents converge towards a result consistent with their environment, diverge or polarize.

Keywords: Argumentation, Opinion Diffusion, Multi-Agent, Social Epistemology

1 Introduction

Nous nous intéressons dans cet article à modéliser la discussion scientifique à travers deux de ses spécificités. Pour cela nous faisons des hypothèses sur le but de ces discussions (de découvrir la vérité sur certains aspects de la réalité auxquels l'expérience nous donne accès) et leur méthode (l'échange répété d'arguments contradictoires). Nous utilisons les frameworks d'argumentation abstraite, munis de sémantiques graduelles, pour baser cette modélisation, que

nous utilisons pour tester quelques hypothèses sur l'influence d'un biais cognitif : le biais de confirmation [13].

Les fondements de l'épistémologie sociale, qui étudie la connaissance et les croyances de groupes d'individus, ont été introduit par [9]. A la suite, de nombreux travaux ont utilisé les systèmes Multi-Agent pour étudier les communautés scientifiques comme exemples de communautés épistémiques. [18] et [14] utilisent des modèles de réseaux d'agents qui interagissent avec un environnement et échangent les résultats de leurs expériences. [18] met en évidence l'"Effet Zollman" : il est parfois désavantageux pour les communautés épistémiques de beaucoup communiquer. Une plus grande communication donne lieu à un consensus plus rapide mais empêche certaines idées d'être explorées, ce qui nuit au résultat collectif. Nous obtenons un comportement similaire avec notre modèle.

Notre travail est directement fondé sur [8] qui présentent une étude de la dynamique d'une sémantique graduelle, un type de sémantique appliqués aux frameworks d'argumentation abstraite qui a récemment été proposée dans la littérature [4, 12, 11, 6]. Nous reprenons de nombreux éléments du protocole de [8] : la sémantique utilisée, la représentation de la connaissance chez les agents et la création d'un débat publique. Nous y introduisons les notions de génération de graphe et d'interaction entre agents et environnement. [8] montrent que l'apprentissage d'arguments par les agents permet à leurs opinions de converger. Notre modèle reproduit ces résultats.

Enfin, [13] ont également étudié les spécificités du raisonnement et de l'argumentation dans les milieux scientifiques. Ils mettent en évidence l'existence d'un biais de confirmation qui affecte les communautés de scientifiques autant que celles de profanes. Le biais de confirmation est un biais cognitif qui consiste à privilégier ses propres idées préconçues ou hypothèses, lors de la production d'arguments ou de l'évaluation des informations disponibles. Nous introduisons le biais de confirmation comme paramètre dans

notre modèle et étudions son effet.

La première section de cet article présente quelques éléments de théorie de l'argumentation abstraite, et la seconde présente notre modèle. Nous exposons dans la dernière section les simulations menées et leurs résultats.

2 Background

2.1 Framework d'argumentation abstraite

Un framework d'argumentation abstraite (AAF pour *Abstract Argumentation Frameworks*) est constitué d'un ensemble fini d'arguments et d'une relation binaire sur cet ensemble, nommée la relation d'attaque. Les AAF peuvent être représentés par des graphes dont les nodes sont les arguments et les arrêtes la relation d'attaque. Dans la suite de cet article, nous utiliserons indépendamment les termes AAF et graphes d'argumentation. La Figure 1 présente un petit débat formé de cinq arguments, et leur représentation en tant que graphe d'argumentation avec la relation d'attaque qui les lie. On peut noter que les graphes d'argumentation *abstraite* ne disent rien dans le cas général du contenu des arguments, mais s'occupent de représenter la structure générale que forme le débat.

L'argumentation abstraite s'intéresse particulièrement à la question de l'*acceptabilité* des arguments. A partir d'un graphe d'argumentation donné, les **sémantiques** sont des fonctions qui déterminent l'acceptabilité des arguments. Les sémantiques proposées par [7] évaluent les arguments en tant qu'ensembles, qui peuvent être acceptables ou non : ici, acceptable peut être compris comme "rationnellement défendable". Récemment, de nouvelles sémantiques ont été proposées pour permettre une évaluation plus fine de la notion d'acceptabilité. Ainsi, les Ranking-based semantics (voir [5, 1, 10, 16, 17, 3, 15]) ordonnent les arguments d'un AAF afin de comparer leurs forces respectives. Un autre type de sémantique sont les sémantiques *graduelles* (voir [4, 12, 11, 6]), qui assignent un score à chaque argument.

2.2 Sémantiques Graduelles

Une **sémantique graduelle** est une fonction qui affecte à chaque argument d'un graphe d'argumentation un score numérique. Le plus souvent, ce score est compris entre 0 et 1 et représente

- Ⓐ La Terre est située au centre de l'Univers.
- Ⓑ Les calculs de Copernic suggèrent que le Soleil est situé au centre de l'Univers.
- Ⓒ La Bible nous dit que « Dieu a fixé la Terre ferme et immobile ».
- Ⓓ Le modèle géocentrique de Ptolémée est capable de prédire le mouvement des planètes.
- Ⓔ Le modèle héliocentrique de Copernic est plus simple et plus précis.

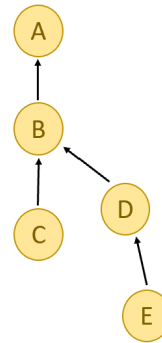


FIGURE 1 – Exemples de débat et sa représentation sous forme d'AAF.

le *degré d'acceptabilité* de l'argument. Ces sémantiques permettent de représenter des situations plus finement et présentent des dynamiques intéressantes. Une des propriétés désirables des sémantiques graduelles est qu'une attaque ne détruit pas complètement un argument, mais diminue simplement son acceptabilité. De plus, la présence de valeurs continues permet de faire le lien entre l'argumentation abstraite et le domaine de la diffusion d'opinion, où les opinions des agents sont comprises entre 0 et 1. Ce sont pour ces raisons que nous avons choisi de travailler avec une sémantique graduelle.

Parmi ces sémantiques, nous utilisons une sémantique graduelle particulière, la *h-categorizer* introduite par [4]. Une version de cette sémantique étendue à des graphes munis de poids a été étudiée par [2] qui montre qu'elle satisfait certaines propriétés désirables.

La sémantique **h-categorizer** affecte à chaque argument le *degré d'acceptabilité* suivant :

$$Hbs(a) = \frac{1}{1 + \sum_{b \in Att(a)} Hbs(b)}$$

où $Att(a)$ représente l'ensemble des arguments qui attaquent a .

La Figure 2 montre les degrés d'acceptabilités donnés par $h\text{-categorizer}$ à chaque argument du graphe d'argumentation vu précédemment. Les arguments non attaqués ont un degré d'acceptabilité maximum de 1.

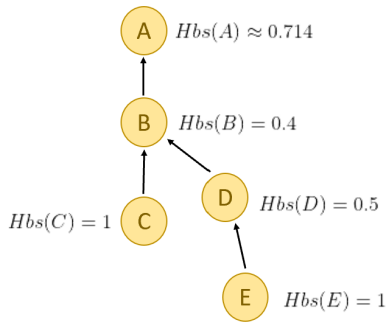


FIGURE 2 – Exemples d'application de la sémantique $h\text{-categorizer}$ à un graphe d'argumentation.

3 Le Modèle

Notre modèle utilise un type d'AAF introduit par [8] : les graphes d'argumentation orientés vers une *issue* (IOAG pour *issue-oriented argumentation graph*). Un argument spécial, l'*issue*, constitue la racine du graphe et tous les autres arguments font partie d'un chemin d'attaque vers cette *issue*. Intuitivement, l'*issue* représente la proposition centrale du débat, et tous les coups des agents ont pour but d'attaquer ou de défendre cette *issue*.

Comme dans le protocole de [8], les agents construisent ainsi un IOAG commun et visible de tous, en publiant des arguments les uns après les autres. Nous nommons cet arbre le *graphe publique scientifique* (PSG pour Public Scientific Graph). Il représente dans notre modélisation une sorte d'"état de l'art", la somme des connaissances publiés par les agents scientifiques étudiant le problème. Le degré d'acceptabilité, calculé par la sémantique, de l'*issue* au sein de ce graphe correspond à une conclusion collective, qui découle de la production commune des agents. On la nomme V_P pour valeur publique.

Chaque agent possède également son propre AAF, qui contient l'*issue* et les arguments auquel l'agent adhère. Ce graphe correspond à la

base de croyance de l'agent, et le degré d'acceptabilité de l'*issue* au sein de ce graphe est appelé **opinion** de l'agent. Les arguments d'un graphe d'opinion ne sont pas nécessairement tous reliés à l'*issue* par une chaîne d'attaques car au cours du jeu, les agents ajoutent à leur graphe d'opinion les arguments qu'ils produisent ainsi que certains arguments publiés par leurs pairs.

L'environnement extérieur est représenté par une valeur WV (pour World Value) comprise entre 0 et 1, qui correspond au degré d'acceptabilité (ou degré de croyance) que les agents devraient affecter à l'*issue* du débat si ils avaient un accès total à la "réalité". Bien sûr, les agents n'ont jamais accès directement à cette valeur ; celle-ci guide leur production d'arguments. Les agents ont également plus de probabilités de générer des arguments qui vont dans le sens de leur propre opinion. Le but de notre modèle est d'étudier les conditions dans lesquels la valeur publique produite par les agents, et les opinions individuelles des agents, convergent ou non vers cette valeur de l'environnement.

Au début du jeu, le PSG et les graphes des agents sont initialisés avec l'*issue* uniquement. Les agents jouent chacun à leur tour, une fois par itération du modèle, dans un ordre aléatoire. A chaque fois qu'il joue, un agent essaie de générer un argument, avec une certaine probabilité : il s'agit d'un simple test aléatoire, cette probabilité correspondant à sa "productivité". Si l'agent réussit, l'argument qu'il produit est généré aléatoirement en prenant en compte l'environnement et l'opinion de l'agent, puis il est publié dans le PSG.

A la fin de chaque tour, les autres agents prennent connaissance du nouvel argument, l'évaluent et peuvent l'ajouter à leur graphe d'opinion.

3.1 Génération d'arguments

A chaque tour, un agent a une certaine probabilité $p_{productivité}$ de générer un nouvel argument. Ce nouvel argument est un contre-argument qui attaque un des arguments du PSG. L'argument attaqué dépend de deux facteurs relatif à l'impact qu'aura cette attaque sur la valeur V_P :

- La **World Value** WV : les arguments qui ont pour impact de rapprocher V_P de WV ont plus de probabilité d'être générés. Un paramètre α nommé précision détermine à quel point la génération d'argument est précise, c'est à dire correspond à la réalité.

• **L'opinion** de l'agent : les arguments qui ont pour impact de rapprocher V_P de l'opinion de l'agent ont plus de probabilité d'être générés par cet agent. Le paramètre β , ou paramètre de biais, détermine à quel point la génération d'argument est biaisée et favorise l'opinion de l'agent.

Formellement à chaque tour, pour chaque argument i du PSG, on calcule la valeur V_P si on ajoute un contre-argument à i . Notons V_P et V_{P_i} les valeurs du PSG sans et avec l'attaque contre i respectivement, et O l'opinion de l'agent.

Alors, on associe à i un poids $w_i = a * b$:

- Si $|\mathcal{WV} - V_P| \leq |\mathcal{WV} - V_{P_i}|$, $a = \alpha$, sinon $a = 1 - \alpha$. On fixe $\alpha \geq 0.5$.
- Si $|O - V_P| \leq |O - V_{P_i}|$, $b = \beta$, sinon $b = 1 - \beta$. De même, $\beta \geq 0.5$.

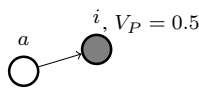
Alors, pour N arguments, la probabilité de générer un contre-argument à l'argument j est :

$$P_j = \frac{e^{\gamma w_j}}{\sum_{i=1}^N e^{\gamma w_i}}$$

Cette fonction permet d'affecter à chaque argument une probabilité proportionnelle au rapport entre son poids et le reste des possibilités. De plus, on peut modifier la façon dont les différences de poids entre arguments affectent les différences de probabilités grâce à γ , le paramètre de diffusion. Après quelques essais sur des graphes simples, nous avons fixé $\gamma = 8$.

Example

Soit un jeu à un agent : dans ce cas, le graphe publique PSG et le graphe d'opinion de l'agent sont identiques, représentés par la figure ci-dessous. $\mathcal{WV} = 0.7$, et les caractéristiques de l'agent sont $\alpha = 0.9$ et $\beta = 0.7$.



L'agent commence par effectuer un test qu'il réussit avec la probabilité $P_{productivity}$. S'il réussit, il peut générer un argument qui attaque l'issue i ou l'argument a . La valeur actuelle de l'état de l'art V_P et de l'opinion de l'agent O sont les mêmes, puisque les graphes sont identiques : $V_P = O = 0.5$. $V_{P_i} \approx 0,33$ et $V_{P_a} \approx 0,67$. Ainsi, attaquer a rapprocherait la valeur du graphe publique de la World Value, contrairement à une attaque envers i . Les deux attaques éloignent cette valeur de l'opinion de l'agent. On a donc $w_i = (1 - \alpha)(1 - \beta) = 0,03$ et $w_a = \alpha(1 - \beta) = 0,27$. Finalement :

- $P_i \approx 0,13$
- $P_a \approx 0,87$

Ainsi, l'agent a plus de chance de générer un argument qui attaque a que i , mais il n'est pas impossible qu'il génère une attaque contre i .

3.2 Evaluation des arguments

Après chaque publication d'arguments, l'agent qui a produit cet argument l'ajoute à son graphe d'opinion, et les autres agents évaluent chacun l'argument pour déterminer si ils l'acceptent (et l'ajoutent à leur propre graphe) ou non. Pour cela, ils effectuent un test de probabilité P_{accept} . Ainsi, chaque agent a la même probabilité d'accepter tous les arguments produits par ses pairs .

3.3 Déroulé d'un jeu

On observe la création d'un arbre d'argumentation orienté vers l'issue (IOAG) dont la forme dépend fortement de la World Value. Lorsque α est haut et β bas, c'est à dire le cas d'agents précis et peu biaisés, on observe que les arbres générés pour une \mathcal{WV} proche de 1 sont souvent constitués d'un attaquant direct de l'issue et de nombreux contre attaquants à cet argument. Inversement, les arbres générés lorsque \mathcal{WV} est proche de 0 présentent généralement de nombreux attaquants directs de l'issue. La Figure 5 présente deux exemples de ces types d'arbres. La taille des noeuds est proportionnelle au degré d'acceptabilité des arguments.

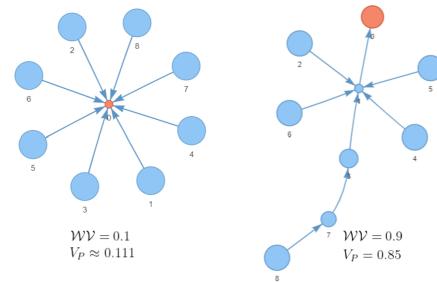


FIGURE 3 – Exemples d'arbres générés par les agents.

1. [8] introduisent un biais de confirmation dans l'évaluation des arguments par les agents (les agents ont plus de chance d'accepter les arguments qui confirment leur opinion). Pour l'instant, nous restreignons notre étude au biais de confirmation appliqué à la génération d'arguments.

4 Simulations

Les questions de recherche qu'on se pose dans cet article sont les suivantes :

1. Est ce qu'une plus grande précision α permet bien au résultat du débat d'être plus proche de la World Value ?
2. Est ce que plus d'agents ou bien plus de *rounds* permettent une convergence plus rapide ?
3. [8] montrent expérimentalement que lorsque la probabilité d'acceptation des arguments augmente, les opinions des agents convergent. Dans leur modèle, les agents ne génèrent pas d'arguments mais débutent le jeu avec un graphe d'opinion dont ils peuvent jouer les arguments. Notre modèle génératif reproduit-il ces résultats ?
4. Quel est l'effet du biais de génération ? Est-il similaire au niveau collectif et individuel ?

Les questions 1 et 2 permettent de valider que notre modèle fonctionne correctement, et les questions 3 et 4 sont des questions ouvertes sur les phénomènes que le modèle fait émerger. Pour répondre à ces questions, nous introduisons les trois métriques suivantes :

L'**Erreur collective (CE)** correspond à la distance entre \mathcal{WV} et V_p . Elle mesure la réussite des agents en tant que groupe construisant une somme de connaissances.

L'**Ecart type des opinions des agents (STD)**, qui mesure la diversité des opinions.

L'**Erreur de la moyenne (EA)** qui correspond à la distance entre la moyenne des opinions des agents et la \mathcal{WV} . On peut voir la moyenne des opinions des agents comme une représentation du consensus. Cette métrique permet de caractériser la réussite individuelle des agents.

4.1 Description des Tests Expérimentaux

Nous avons généré aléatoirement \mathcal{WV} à chaque itération. Le paramètre de productivité $p_{productivity}$ est fixé à 0.5 pour toutes nos expériences. Les valeurs par défaut des paramètres sont : $\alpha = 0.95$, $\beta = 0.6$, $P_{accept} = 0.7$, le nombre de *rounds* $S = 20$ et le nombre d'agents $N = 3$. Chaque résultat correspond à la moyenne de la métrique correspondante sur 150 débats.

Nous présentons les résultats de simulation à 3 agents pour limiter le temps d'expérimentation, tout en nous permettant d'observer des phénomènes intéressants qui diffèrent du cas avec un seul agent. Nous avons observé les mêmes phénomènes lors d'une étude plus qualitative des jeux avec 6 agents.

5 Résultats

5.1 Effet de la précision

On fait varier la précision α en laissant les valeurs par défaut. Les résultats présentés dans la Table 1 et Figure 4 montrent qu'une plus grande précision permet bien de diminuer l'erreur collective et les erreurs individuelles des agents. On observe aussi une légère diminution de l'écart type des opinions des agents. Nous supposons que cela est dû au fait que les agents convergent tous vers une opinion plus proche de la \mathcal{WV} .

	STD	CE	EA
0,5	0,074	0,257	0,260
0,6	0,079	0,181	0,186
0,7	0,068	0,133	0,146
0,8	0,065	0,080	0,108
0,9	0,060	0,051	0,082
1	0,064	0,036	0,068

TABLE 1 – Variation de l'écart type des opinions (STD), l'erreur collective (CE), et l'erreur de la moyenne (EA) en faisant varier la précision α .

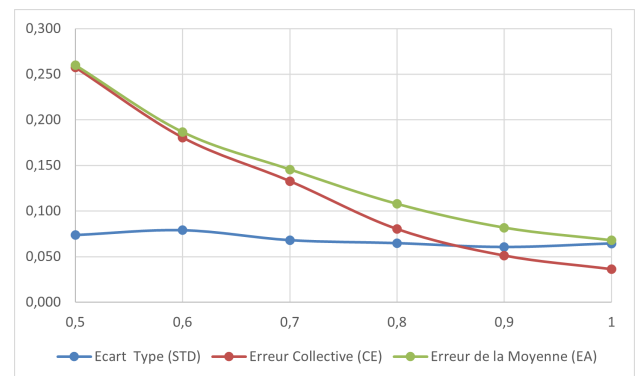


FIGURE 4 – Graphe représentant la variation des trois métriques en faisant varier la précision α .

5.2 Rôle du nombre d'arguments générés

Les règles du protocole nous donnent une approximation du nombre d'argument générés :

P_{accept}	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
Ecart Type	0,154	0,142	0,121	0,105	0,086	0,072	0,060	0,047	0,027	0,000
Erreur Collective	0,025	0,034	0,033	0,039	0,030	0,041	0,038	0,042	0,031	0,047
Erreur de la Moyenne	0,149	0,141	0,121	0,109	0,085	0,076	0,071	0,066	0,043	0,047

TABLE 2 – Variation de l'écart type des opinions (STD), l'erreur collective (CE), et l'erreur de la moyenne (EA) en faisant varier la probabilité d'acceptation P_{accept} .

$N_{arguments} \approx N_{agents} * P_{productivity} * S$, avec S le nombre de *round* du modèle. La Table 3 montre l'erreur collective (CE) en fonction du nombre d'agents et de *rounds* du modèle. Les résultats confirment que la convergence à la WV de la valeur publique est plus rapide lorsque le nombre d'agent augmente, et lorsque le nombre de *rounds* du modèle augmente. On peut en conclure que le nombre d'argument générés influence positivement la convergence. Une légère augmentation de l'erreur pour 6 agents et 50 arguments nous indique qu'il pourrait exister un nombre optimal de nombre d'arguments produits.

Agents/Rounds	10	20	30	50
1	0,094	0,075	0,064	0,051
2	0,054	0,053	0,045	0,029
6	0,041	0,033	0,024	0,026

TABLE 3 – Erreur collective (CE) en fonction du nombre d'agents et de *rounds* du modèle.

5.3 Effet de la probabilité d'acceptation

Nous faisons varier la probabilité d'acceptation des arguments. Les résultats présentés en Table 2 confirment les résultats expérimentaux de [8] : plus la probabilité d'acceptation est haute, plus l'opinion des agents converge. On observe que l'erreur de la moyenne des agents diminue également lorsque P_{accept} augmente. Cela est dû au phénomène identifié plus haut : avoir accès à plus d'arguments permet aux agents de mieux converger, dans le cas où ces agents sont précis et peu biaisés. L'erreur collective augmente légèrement, ce qui peut être dû à l'effet néfaste du biais de confirmation lorsque les communautés communiquent plus, qui est identifié à la section suivante.

5.4 Effet du biais de génération

On fait varier le biais β dans le cas de 3 agents avec les paramètres par défaut, et dans

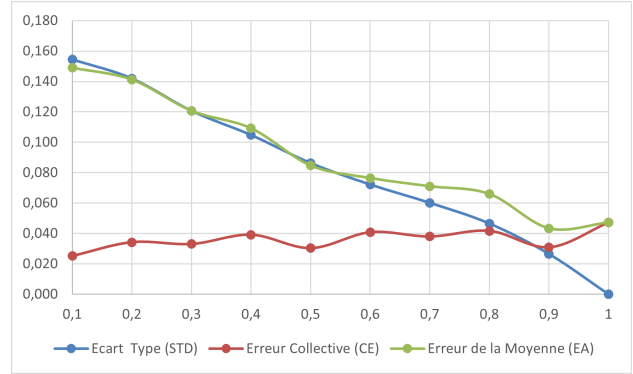


FIGURE 5 – Graphe représentant la variation des trois métriques en fonction de la probabilité d'acceptation P_{accept} .

un deuxième cas où les trois agents n'acceptent aucun argument de leurs pairs : $P_{accept} = 0$.

	β	$P_{accept} = 0$	$P_{accept} = 0.7$
CE	0,5	0,029	0,032
	0,6	0,035	0,043
	0,7	0,039	0,049
	0,8	0,071	0,097
	0,9	0,107	0,156
	0,9999	0,162	0,213
STD	0,5	0,155	0,058
	0,6	0,174	0,064
	0,7	0,174	0,064
	0,8	0,19	0,07
	0,9	0,224	0,08
	0,9999	0,248	0,073
EA	0,5	0,179	0,068
	0,6	0,18	0,073
	0,7	0,194	0,081
	0,8	0,215	0,116
	0,9	0,224	0,174
	0,9999	0,273	0,23

TABLE 4 – Variation de l'écart type des opinions (STD), l'erreur collective (CE), et l'erreur de la moyenne (EA) en faisant varier le biais β , dans le cas avec acceptation $P_{accept} = 0.7$ et sans $P_{accept} = 0$.

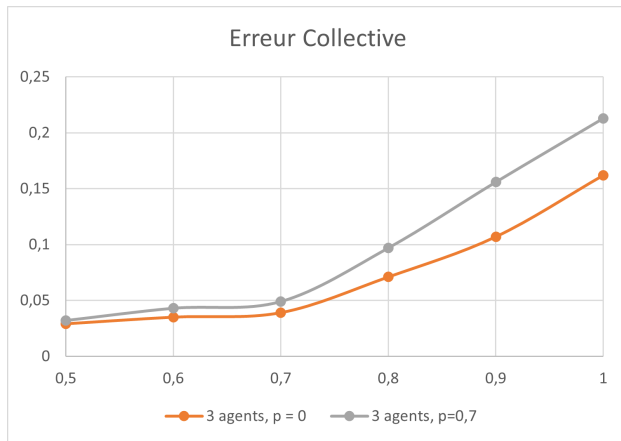


FIGURE 6 – Graphe représentant la variation de l'erreur collective (CE) en fonction du biais β dans le cas avec acceptation $P_{accept} = 0.7$ et sans $P_{accept} = 0$.

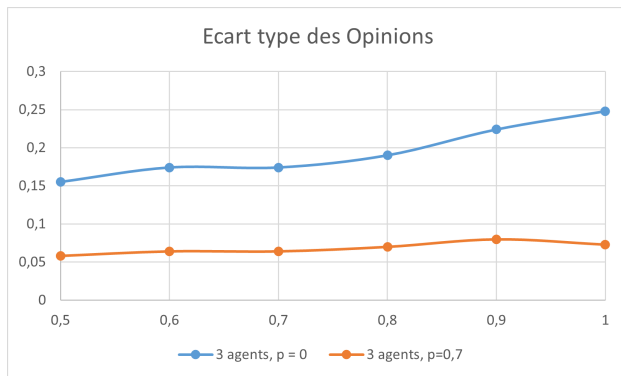


FIGURE 7 – Graphe représentant la variation de l'écart type des opinions (STD) en fonction du biais β dans le cas avec acceptation $P_{accept} = 0.7$ et sans $P_{accept} = 0$.

Les résultats présentés en Table 4 et les Figures 6 et 8 nous montrent que le biais a un effet néfaste sur les succès individuels et collectifs des agents. Cela est cohérent avec la vision classique du biais de confirmation comme un obstacle à la rationalité des raisonnements individuels. Le groupe d'agents à $P_{accept} = 0$ montre une plus grande diversité d'opinion que celui à $P_{accept} = 0.7$, comme illustré en Figure 7, ce qui est cohérent avec les résultats de la section précédente. On observe également que les agents qui n'acceptent pas d'autres arguments que les leurs semblent protégés des effets du biais, et commettent moins d'erreurs que ceux qui acceptent certains arguments de leurs pairs. Ce résultat qui pourrait apparaître contre-intuitif à première vue semble pointer vers un facteur protectif de

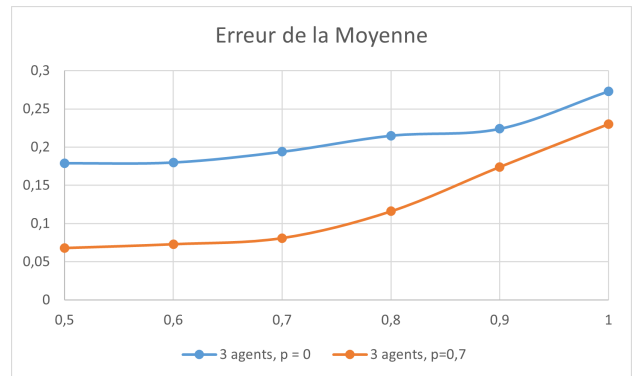


FIGURE 8 – Graphe représentant la variation de l'erreur de la moyenne (EA) en fonction du biais β dans le cas avec acceptation $P_{accept} = 0.7$ et sans $P_{accept} = 0$.

la diversité d'opinion : les agents ayant chacun des opinions différentes, le résultat de leurs efforts collectifs reflète mieux la "réalité", c'est à dire la \mathcal{W} . Au contraire, lorsque les agents apprennent les uns des autres, le résultat collectif est plus biaisé vers leur opinion commune. Cette intuition semble confirmée par le fait que l'erreur de la moyenne des opinions des agents est plus grande dans le cas sans acceptation d'arguments. Ainsi, lorsque les agents biaisés ne communiquent pas, ils font plus d'erreur individuellement mais ils parviennent collectivement à mieux approcher la vérité. Ces résultats sont proches de ceux obtenus par [18] et montrent que la communication peut parfois s'avérer désavantageuse pour une communauté à la recherche de la vérité.

6 Conclusion et Discussion

Les résultats semblent montrer que notre modèle permet bien de modéliser la génération d'arguments influencés par un environnement extérieur à l'aide d'une sémantique graduelle. Nous modélisons de façon satisfaisante les notions de précision par rapport à l'environnement et de biais de confirmation dans la génération d'arguments. A notre connaissance, il s'agit du premier travail de recherche qui utilise les sémantiques graduelles pour représenter la construction d'un débat influencé par une notion de réalité extérieure et la convergence associée.

Nous confirmons les résultats de [8] qui semblent ainsi robuste aux détails du protocole utilisé : partager des arguments permet aux opinions des agents de converger. Cela permet de créer un lien entre les frameworks d'argumen-

tation abstraite et la diffusion d'opinion. Nous observons également que la diversité d'opinion semble protéger les groupes d'agents contre les effets du biais de confirmation. Ce résultat rappelle ceux obtenus par [18] : une communication réduite est parfois bénéfique pour les communautés épistémiques.

Nous aimerions dans un second temps dépasser la limitation à trois agents, pour confirmer les résultats obtenus lorsque le nombre d'agents est plus élevé. Certaines limitations du modèle pourraient être dépassées en utilisant des graphes d'argumentation *bi-directionnels*, c'est à dire contenant des relations de support aussi bien que d'attaque entre arguments. Nous souhaitons également inclure un biais de confirmation lors de l'évaluation des arguments, comme [8] Enfin, nous aimerions étendre le modèle et inclure, en s'inspirant du travail de [14], la diffusion de la production de connaissances scientifiques auprès d'une communauté de profanes.

Références

- [1] Leila AMGOUD et Jonathan BEN-NAIM. "Ranking-Based Semantics for Argumentation Frameworks". In : *Proc. of the 7th International Conference on Scalable Uncertainty Management, (SUM'13)*. 2013, p. 134-147.
- [2] Leila AMGOUD et al. "Acceptability Semantics for Weighted Argumentation Frameworks". In : *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 2017, p. 56-62. doi : 10.24963/ijcai.2017/9. URL : <https://doi.org/10.24963/ijcai.2017/9>.
- [3] Leila AMGOUD et al. "Ranking Arguments With Compensation-Based Semantics". In : *Proc. of the 15th International Conference on Principles of Knowledge Representation and Reasoning, (KR'16)*. 2016, p. 12-21.
- [4] Philippe BESNARD et Anthony HUNTER. "A logic-based theory of deductive arguments". In : *Artificial Intelligence* 128.1-2 (2001), p. 203-235.
- [5] Claudette CAYROL et Marie-Christine LAGASQUIE-SCHIEX. "Graduality in Argumentation". In : *Journal of Artificial Intelligence Research* 23 (2005), p. 245-297.
- [6] Célia da COSTA PEREIRA, Andrea TETTAMANZI et Serena VILLATA. "Changing One's Mind : Erase or Rewind ?" In : *Proc. of the 22nd International Joint Conference on Artificial Intelligence, (IJCAI'11)*. 2011, p. 164-171.
- [7] Phan Minh DUNG. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and N-persons games". In : *Artificial Intelligence* 77 (1995), p. 321-357.
- [8] Louise DUPUIS DE TARLÉ, Elise BONZON et Nicolas MAUDET. "Multiagent Dynamics of Gradual Argumentation Semantics". In : *Proceedings of the 21th International Conference on Autonomous Agents and Multi Agent Systems (AAMAS)* (à para.).
- [9] Alvin I GOLDMAN. *Knowledge in a social world*. Oxford University Press, 1999.
- [10] Davide GROSSI et Sanjay MODGIL. "On the Graded Acceptability of Arguments". In : *Proc. of the 24th International Joint Conference on Artificial Intelligence, (IJCAI'15)*. 2015, p. 868-874.
- [11] João LEITE et João MARTINS. "Social Abstract Argumentation". In : *Proc. of the 22nd International Joint Conference on Artificial Intelligence, (IJCAI'11)*, 2011, p. 2287-2292.
- [12] Paul-Amaury MATT et Francesca TONI. "A Game-Theoretic Measure of Argument Strength for Abstract Argumentation". In : *Proc. of the 11th European Conference on Logics in Artificial Intelligence, (JELIA'08)*. 2008, p. 285-297.
- [13] Hugo MERCIER et Christophe HEINTZ. "Scientists' argumentative reasoning". In : *Topoi* 33.2 (2014), p. 513-524.
- [14] Cailin O'CONNOR et James Owen WEATHERALL. "Scientific polarization". In : *European Journal for Philosophy of Science* 8.3 (2018), p. 855-875.
- [15] Theodore PATKOS, Antonis BIKAKIS et Giorgos FLOURIS. "A Multi-Aspect Evaluation Framework for Comments on the Social Web". In : *Proc. of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*. 2016, p. 593-596.

- [16] Fuan Pu et al. "Argument Ranking with Categoriser Function". In : *Proc. of the 7th International Conference on Knowledge Science, Engineering and Management, (KSEM'14)*. 2014, p. 290-301.
- [17] Fuan Pu et al. "Attacker and Defender Counting Approach for Abstract Argumentation". In : *Proc. of the 37th Annual Meeting of the Cognitive Science Society, (CogSci'15)*. 2015.
- [18] Kevin JS ZOLLMAN. "The communication structure of epistemic communities". In : *Philosophy of science* 74.5 (2007), p. 574-587.

Transmission de connaissances et sélection

Y. Bourahla^a M. Atencia^b J. Euzenat^a
 Yasser.Bourahla@inria.fr atencia@lcc.uma.es Jerome.Euzenat@inria.fr

^aUniv. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France

^bUniversidad de Málaga, 29071 Málaga, Spain

Résumé

Les agents peuvent être utilisés pour simuler l'évolution culturelle et l'évolution culturelle peut être utilisée comme modèle pour les agents artificiels. Des expériences ont montré que la transmission intragénérationnelle des connaissances permet aux agents d'en améliorer la qualité. De plus, sa transmission intergénérationnelle permet de dépasser ce niveau. Ces résultats ont été obtenus dans des conditions particulières : sélection drastique des agents transmettant leurs connaissances, initialisation avec des connaissances correctes ou introduction de bruit lors de la transmission. Afin d'étudier l'impact de ces mesures sur la qualité de la connaissance transmise, nous combinons les paramètres de deux expériences précédentes et relâchons ces conditions. Ce dispositif confirme que la transmission verticale permet d'améliorer la qualité de la connaissance obtenue par transmission horizontale même sans sélection drastique et apprentissage orienté. Il montre également qu'une transmission intragénérationnelle suffisante peut compenser l'absence de sélection parentale.

Mots-clés : *Simulation sociale multi-agents, Evolution culturelle, Transmission des connaissances, Génération d'agents, Evolution culturelles des connaissances*

Abstract

Agents have been used for simulating cultural evolution and cultural evolution can be used as a model for artificial agents. Previous results have shown that intra-generation knowledge transmission allows agents to improve the quality of their knowledge. Moreover, variation generated through inter-generation transmission allows agents to exceed that level. Such results were obtained under specific conditions such as the drastic selection of agents allowed to transmit their knowledge, seeding the process with correct knowledge or introducing artificial noise during transmission. Here, we question the necessity of such measures and study their impact on the quality of transmitted knowledge. For that

purpose, we combine the settings of two previous experiments and relax these conditions. We confirm through this setting that vertical transmission improves on horizontal transmission even without drastic selection and oriented learning. We also show that sufficient intra-generation transmission can compensate for the absence of parent selection.

Keywords: *Multi-agent social simulation, Cultural evolution, Knowledge transmission, Agent generation, Cultural knowledge evolution*

1 Introduction

L'évolution culturelle [12, 19] applique les concepts de l'évolution biologique à la culture d'une société. Ceci a été exploré expérimentalement à l'aide de systèmes multi-agents faisant évoluer le comportement d'une population d'agents [13, 5], leur langage [21] leurs connaissances [14] ou leur culture. L'évolution culturelle repose sur sa capacité à présenter des variations qui peuvent être sélectionnées et transmises. Différents modes de transmission culturelle, inspirés de l'épidémiologie, ont été identifiés : [12, 10] : *transmission verticale* des parents aux enfants, *transmission oblique* des agents de la génération précédente (comme l'éducation) à ceux de la génération suivante, et *transmission horizontale* entre les agents de la même génération. Ici, nous utiliserons également *transmission intergénérationnelle* pour les deux premières et *transmission intragénérationnelle* pour la dernière.

Les systèmes multi-agents dans lesquels les agents s'adaptent suite à leurs interactions [21] peuvent être considérés comme une transmission intragénérationnelle. Cette approche s'est avérée efficace pour développer une culture partagée. Récemment, nous avons étudié les agents qui adaptent leurs connaissances sur l'environnement, exprimées en tant qu'ontologies, par le biais d'interactions sociales afin de parvenir à un accord. Nous avons constaté qu'ils amélioreraient

la précision de leurs ontologies sans nécessairement adopter la même ontologie [8]. Ainsi, la transmission horizontale est capable d'améliorer les connaissances dans une société d'agents.

La transmission verticale, d'une génération à une autre, peut être considérée comme l'occasion de rebattre les cartes. Elle peut introduire plus de variation ou imposer (sélectionner) une culture dominante. Il convient donc de déterminer les rôles respectifs de la transmission intergénérationnelle et intragénérationnelle.

D'autres travaux ont considéré exactement ceci et ont montré que la transmission intergénérationnelle permet aux agents d'améliorer leur connaissance au-delà de ce que la transmission intragénérationnelle seule permet [13]. En revanche, en désactivant la transmission intergénérationnelle, les agents doivent recommencer à améliorer leurs connaissances à chaque génération. Dans un cadre similaire, il a également été constaté que la transmission intragénérationnelle engendre de la variation et que la transmission intergénérationnelle procède à la sélection [1].

Cependant, ces travaux reposent sur de fortes contraintes pour assurer la transmission fidèle de connaissances correctes (en utilisant uniquement la partie de la population la plus performante comme enseignants ou en initialisant les ontologies des agents à partir d'échantillons corrects) ou pour générer de la variation (en ajoutant du bruit dans le processus d'apprentissage).

Ici, nous relâchons ces contraintes pour étudier les effets de la transmission horizontale et verticale de la culture sans sélectionner les agents les plus performants et sans commencer avec des informations correctes. Pour ce faire, l'opportunité de se reproduire et de propager ses connaissances est élargie à l'ensemble de la société d'agents et les connaissances sont initialisées aléatoirement. Cela est susceptible de fournir suffisamment de variation au système.

Pour tester cela, nous avons conçu un nouveau cadre expérimental basé sur ces travaux précédents, en faisant varier les modalités de transmission et les stratégies d'appariement. La transmission intergénérationnelle s'appuie sur l'enseignement comme [1] et la transmission intragénérationnelle sur l'interaction comme [8]. Nous confirmons les résultats des études précédentes sous des hypothèses relâchées : la transmission intergénérationnelle augmente la correction des connaissances obtenues par la transmission intragénérationnelle. De plus, nous montrons que même si la transmission intergénérationnelle

ne sélectionne pas les parents les plus aptes, la transmission intragénérationnelle, avec suffisamment de temps, peut compenser cela, c'est-à-dire que les agents se mettent finalement d'accord pour sélectionner les éléments de connaissance pertinents.

Le reste de l'article est organisé comme suit : dans la section 2, on présente les travaux connexes. La section 3 aborde les limites des systèmes actuels et propose des solutions pour les surmonter. Leur mise en œuvre effective est précisément détaillée dans la Section 4. La section 5 présente les expériences réalisées et la section 6 fournit leurs résultats qui sont discutés dans la section 7.

Cet article est une version abrégée en français de [9].

2 Travaux connexes

Nous présentons d'abord les travaux relatifs aux agents et à l'évolution culturelle en général avant de considérer plus précisément deux expériences sur lesquelles ce travail est basé.

2.1 Agents et évolution culturelle

Les interactions entre agents peuvent servir de moyen d'adaptation. De nombreuses approches multi-agents les exploitent, par exemple, pour améliorer l'interopérabilité [22, 4] ou observer les comportements émergents [17]. Dans certains cas, en interagissant les uns avec les autres, les agents finissent par partager et faire évoluer une culture particulière, par exemple des connaissances sur l'environnement, des façons d'interagir, des normes ou des conventions [20, 15]. Ce phénomène est généralement étudié en observant une population d'agents qui interagissent entre eux selon un protocole bien défini. L'état du système est suivi jusqu'à ce que les agents atteignent un état stable dans lequel leur culture ne change plus. Les caractéristiques de cette culture sont alors évaluées [21].

Ici, nous nous intéressons à l'évolution des connaissances que les agents utilisent pour se comporter. L'évolution culturelle a été appliquée aux connaissances sous la forme d'alignements d'ontologies [14, 2]. Elle a également été appliquée aux ontologies [8] dans lesquelles les agents adaptent leurs définitions de concepts pour prendre des décisions communes. Dans ce contexte, les agents apprennent des autres en interagissant avec eux.

Lorsque les agents parviennent à un état stable, ils atteignent un optimum local en ce qui concerne la qualité de leurs connaissances. Pour sortir de cet état, il faut modifier l'environnement ou les agents.

L'évolution de différentes générations d'agents a été envisagée dans les systèmes multi-agents évolutionnaires [11] dans lesquels les agents peuvent se reproduire et mourir. Cela n'a pas été particulièrement étudié sous l'angle de la culture développée par ces agents. Cela a été proposé dans les algorithmes culturels [18]. Cependant, ces travaux ont été principalement appliqués à des problèmes d'optimisation.

Il a été montré que la transmission de culture permet de découvrir des comportements inaccessibles à l'évolution génétique [16]. La simulation multi-agents de l'évolution culturelle impliquant la transmission intragénérationnelle et intergénérationnelle a été étudiée dans une série d'expériences [13, 5, 1, 3]. En se basant sur les réseaux de neurones pour incarner les connaissances, ils mettent en œuvre la communication de poids de connexion [5], l'apprentissage supervisé (imitation) [13, 1] ou l'instruction directe [3] comme mécanisme de transmission verticale. En utilisant le même mécanisme d'imitation pour la transmission horizontale [1], il a été montré que la transmission de la culture intragénérationnelle fournissait la variation qui permettait au processus évolutif d'être plus efficace que ce qu'il est avec la transmission intergénérationnelle uniquement. Ce travail est cependant basé sur une sélection drastique des enseignants auprès desquels les agents apprennent, tant en transmission oblique qu'horizontale.

Nous discutons plus longuement ci-dessous les travaux développés dans [1] et [8] sur lesquels nous nous appuyons.

2.2 Héritage culturel intergénérationnel

Acerbi et Parisi [1] ont conçu une expérience dans laquelle des agents utilisent un réseau de neurones pour naviguer dans leur environnement. Lorsqu'un agent s'approche d'une source de nourriture comestible, il reçoit une récompense et lorsqu'il s'approche d'une source de nourriture toxique, il reçoit une pénalité. À la naissance, les agents commencent avec des réseaux de neurones pondérés aléatoirement (W). Dans la première partie de leur vie, les agents apprennent leur comportement par des enseignants de la génération précédente : transmission oblique. Ensuite, ils l'apprennent de ceux

de leur génération qui ont le mieux réussi : transmission horizontale. Un point clé est que, dans les deux cas, ils ne sont enseignés que par les 5% meilleurs agents en termes de récompense accumulée. Les enseignants peuvent ajouter du bruit dans leur comportement afin d'engendrer de la variation.

Les résultats montrent que la transmission oblique seule permet aux agents de s'améliorer (mesuré par les récompenses accumulées) au fil des générations, alors que la transmission horizontale ne le fait pas, car les agents apprennent d'agents novices. Ils montrent également que l'introduction de bruit est essentielle pour accélérer l'amélioration, et que la transmission horizontale peut le fournir.

Dans ces expériences, la sélection des 5% agents les plus performants dans les deux modes de transmission (20% pour [13]) est un biais très fort car il donne peu d'opportunité à la population générale de se reproduire. Cela ne se produit pas dans les sociétés humaines.

2.3 Apprentissage social intragénérationnel

Nous avons conçu une expérience dans laquelle des agents vivent dans un environnement peuplé d'objets, identifiés par leurs caractéristiques, pour lesquels ils doivent prendre des décisions [8]. Le but des agents est de se mettre d'accord avec les autres agents sur la décision à prendre. Ils suivent un protocole d'expérimentation en deux étapes (Figure 1, à droite). Premièrement, les agents apprennent à prendre des décisions sur les objets de leur environnement à partir d'un échantillon (S). Le résultat de cette étape est, en particulier, une ontologie simple (O) permettant de classer les objets et de déterminer la décision à prendre. Ensuite, les agents interagissent répétitivement entre eux deux par deux. Ils comparent les décisions qu'ils prendraient concernant les objets. Les interactions réussissent lorsque les agents sont d'accord sur la décision à prendre; elles échouent dans le cas contraire. En cas d'échec, l'un des agents adapte son ontologie afin d'adopter la décision de l'autre sur cet objet.

Les expériences ont montré que les agents sont capables d'atteindre un état dans lequel ils sont toujours d'accord. La qualité de leurs ontologies, mesurée par sa correction¹, augmente en moyenne, mais pas dans tous les cas, et n'atteint pas nécessairement une correction totale.

1. Dans ce papier, nous traduisons le terme *accuracy* [8] par correction. Il s'agit d'un degré de correction.

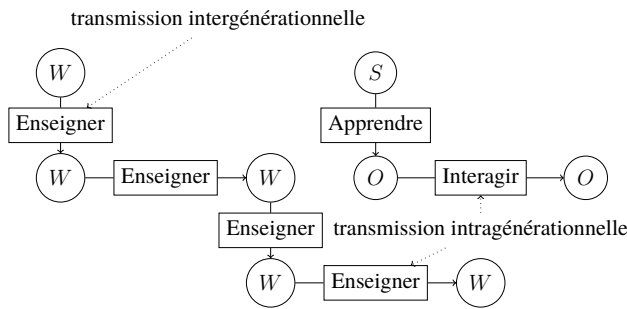


FIGURE 1 – Cadres expérimentaux de [1] (à gauche) et de [8] (à droite). (W =poids des réseaux de neurones, S =échantillon d'entraînement, O =ontologie). Malgré les similarités structurelles, les mécanismes de transmission des connaissances sont différents : les deux mécanismes intergénérationnels sont des apprentissages supervisés, mais la transmission intragénérationnelle est différente.

Enfin, bien que la distance entre les ontologies des agents diminue, les agents ne se retrouvent pas nécessairement avec les mêmes ontologies.

Le critère de réussite des agents est de se mettre d'accord, mais pas nécessairement de prendre la bonne décision. La décision correcte à prendre est inconnue par les agents. Elle n'est considérée que dans deux occasions : (1) initialement, les agents sont formés avec un échantillon correct, mais incomplet ; (2) le choix de l'agent qui adaptera ses connaissances est basé sur les récompenses qu'il a reçues sur un échantillon (différent). Bien que (2) soit légitime, (1) ne l'est pas. Par conséquent, nous le relâchons en utilisant des échantillons étiquetés par des parents ne connaissant pas la décision correcte.

3 Vers une approche intégrée de la transmission des connaissances

Nous discutons d'abord les limites des expériences présentées précédemment. Nous proposons ensuite de les combiner d'une manière qui tienne compte de ces limites. Cela permet de formuler nos affirmations sous forme d'hypothèses.

3.1 Limites

Les deux expériences décrites dans les dernières sous-sections ont le mérite de fournir un modèle de transmission culturelle pour les agents et de montrer qu'elle a un effet positif sur leur rendement. Cependant, elles le font avec des modalités qui semblent fortes et peu naturelles :

- restreindre le rôle des enseignants à une partie très limitée de la population (5%, 20%) va à l'encontre de l'idée d'une culture qui se propage ;
- commencer par des échantillons corrects ne correspond pas à ce qui se passe dans la vie réelle.

Ces deux modalités tendent à imposer la transmission fidèle d'une culture correcte, alors que les avantages de la transmission verticale sont censés provenir de la variation qu'elle procure. C'est au point que [1] introduit un bruit artificiel dans le modèle afin d'augmenter la variation.

Nous conjecturons qu'une capacité moins restreinte de transmettre des connaissances, à la fois entre les générations et au sein d'une même génération, fournit suffisamment de variation pour améliorer la transmission horizontale.

3.2 Combinaison des deux expériences

Pour tester notre conjecture, nous combinons les deux approches de la Figure 1 et relâchons les contraintes imposées. Le cadre résultant, tel qu'illustré dans la Figure 2, est tel que :

transmission intragénérationnelle est mise en œuvre par interaction comme dans [8].

transmission intergénérationnelle est mise en œuvre par l'apprentissage. Comme dans [1], une génération d'agents est remplacée par une autre. Les agents se reproduisent de manière différentielle par rapport à une distribution de probabilité et transmettent leurs connaissances à leurs enfants. Cependant, ils le font avec les modalités de [8].

3.3 Relaxation des contraintes élitistes

Deux mesures sont prises pour relâcher les deux modalités discutées. D'une part, la possibilité pour les agents d'avoir des enfants sera déterminée soit de manière équiprobable, soit par rapport au revenu des agents (dénotant leur succès). D'autre part, l'ontologie initiale pourra être vide, aléatoire, être une fusion des ontologies des parents ou être apprise comme dans la première étape ci-dessus, à partir d'échantillons fournis par les parents (Figure 2). Ainsi, l'échantillon à partir duquel les enfants apprennent peut être incorrect (parce que les parents n'ont pas une connaissance totalement correcte) et incomplet (parce qu'il ne couvre pas tout l'espace objet).

Enfin, aucun bruit artificiel n'est ajouté au comportement des agents. Comme mentionné ci-dessus, le recrutement "égalitaire" des parents et l'inexactitude des échantillons fournis par les parents sont déjà des sources de variation lors de la transmission verticale.

3.4 Hypothèses

Notre objectif est d'étudier principalement deux directions. Premièrement, une fois que les agents ont atteint un accord global, ils n'adaptent plus leurs connaissances. Comme le montre [8], ils peuvent encore s'accorder sur des décisions incorrectes. La transmission intergénérationnelle devrait introduire une variation supplémentaire permettant aux agents de découvrir de nouveaux éléments de connaissance pertinents. En conséquence, comme dans [13, 1], nous faisons l'hypothèse que (H_1) *la transmission verticale permet aux connaissances des nouvelles générations d'être plus correctes que celles de la génération précédente.*

Deuxièmement, selon [8], les agents sont capables d'améliorer l'exactitude de leurs décisions lorsqu'ils adaptent leurs connaissances pour se mettre d'accord. Ceci suggère que la transmission de connaissances intragénérationnelle est capable de sélectionner les éléments de connaissances à préserver. Ainsi, nous faisons l'hypothèse que (H_2), *l'interaction, utilisée pour la transmission intragénérationnelle, peut compenser l'absence de sélection des parents.*

4 Cadre expérimental

Afin de tester les deux hypothèses proposées, nous avons conçu un cadre expérimental suivant les principes ci-dessus avec des mécanismes pour que les agents s'apparient, se reproduisent, meurent, et transmettent leurs connaissances à leurs enfants. Nous décrivons ci-dessous le processus général (Section 4.1) et détaillons ce qui diffère des expériences originales :

- Le cycle de vie des agents (Section 4.2),
- Comment les agents sont sélectionnés pour se reproduire (Section 4.3),
- Comment les agents transmettent leurs connaissances à leurs enfants (Section 4.4).

Plusieurs questions peuvent être soulevées sur la façon dont ces modifications affectent les connaissances des agents. Ici, nous focalisons sur l'impact de la transmission des connaissances sur leur correction.

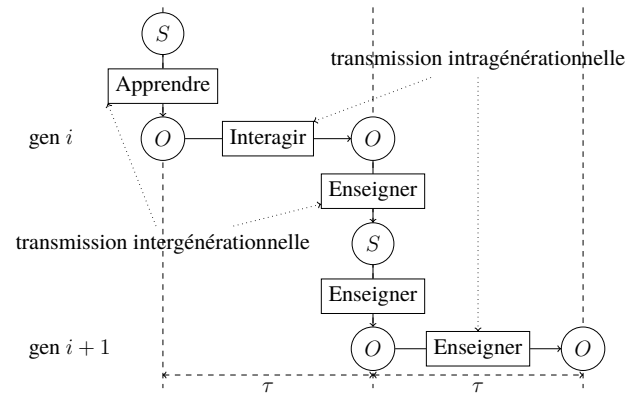


FIGURE 2 – Combinaison des mécanismes de la figure 1 : la transmission horizontale est réalisée par l'interaction des agents ; la transmission verticale utilise des échantillons (S) engendrés par les parents à partir desquels les enfants apprendront leurs ontologies (O).

4.1 Processus global

Nous réutilisons les mêmes environnements et ontologies, procédure d'apprentissage d'ontologie et procédure d'apprentissage social de [8]. Le jeu se déroule avec des objets caractérisés par 4 caractéristiques sur lesquelles une seule décision est valable parmi 6 possibles.

Les expériences sont constituées de périodes divisées en deux parties (Figure 3) :

reproduction dans laquelle la génération $i - 1$ (la moitié de la population) est supprimée (meurt) et la génération $i + 1$ est ajoutée (naît), et les agents de la nouvelle génération acquièrent leurs connaissances initiales ;

interaction dans laquelle les agents utilisent leurs ontologies pour se mettre d'accord sur les décisions concernant les objets et les modifier en cas de désaccord.

Ce processus est illustré dans la figure 2 et décrit précisément ci-dessous.

4.2 Cycle de vie des agents

Un agent naît au début d'une période et meurt à la fin de la période suivante (il vit pendant 2 périodes). Dans la première période, l'agent est considéré comme un enfant. Dans la seconde période, il est considéré comme un adulte. Entre ces deux périodes, les agents peuvent se reproduire selon les modalités précisées dans la section 4.3.

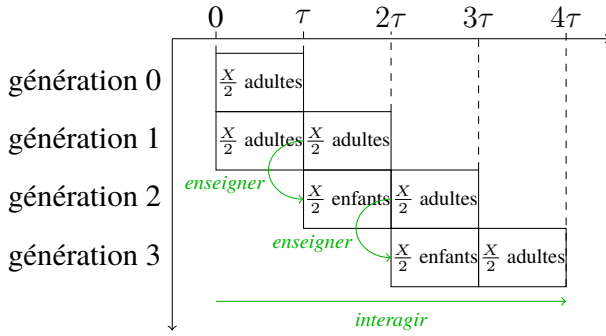


FIGURE 3 – Évolution d'une société au cours de 3 générations d'agents.

Au cours de la première période de leur vie, les agents (enfants) apprennent de leurs parents par le biais d'une transmission de connaissances intergénérationnelle détaillée dans la section 4.4. Durant la deuxième période de leur vie, certains agents deviennent parents et transmettent leurs connaissances à leurs enfants. Ils continuent à interagir avec les autres individus de la société et peuvent adapter leurs connaissances.

À la fin de chaque période, $\frac{X}{2}$ agents de l'ancienne génération meurent et $\frac{X}{2}$ agents naissent. Pour maintenir constant le nombre d'agents dans la population, la population initiale est composée de X agents. La moitié de cette population, considérée comme la génération 0, meurt à la fin de la première période et ne se reproduit pas. La figure 3 montre l'évolution d'une société sur 3 générations.

4.3 Stratégies d'appariement

Pour se reproduire, les agents (adultes) se comportent selon la règle suivante : $v = 2$ parents sont sélectionnés aléatoirement suivant une distribution s pour avoir $c = 1$ enfant. La probabilité de se reproduire suit une distribution s qui peut être :

- Maximale (100%) pour les v agents ayant rassemblé le plus de revenus et minimale (0%) pour les autres agents (*best*),
- Proportionnel au revenu perçu en effectuant leurs tâches (*income*),
- Équiprobable (*random*).

La stratégie maximale est introduite pour simuler la sélection forte [1].

4.4 Transmission des connaissances

Le processus de transmission passe par deux étapes (Figure 3).

Transmission verticale initiale Dans la première étape, chaque agent de la nouvelle génération acquiert des connaissances directement de ses v parents. Ici, nous expérimentons deux types d'ontologies initialement acquises : des ontologies aléatoires (abusivement notées $r = 0$) ou des ontologies apprises des parents en réutilisant la technique de [8]. Plus précisément, $r\%$ de tous les objets ayant des propriétés distinctes (types d'objets) sont sélectionnés aléatoirement. Chaque parent étiquette la moitié de ces objets avec les décisions qu'il prendrait (qui peuvent être incorrectes). Cet ensemble est présenté à l'enfant comme un échantillon d'apprentissage (S) à partir duquel il apprend son ontologie.

Interactions Une fois cette transmission initiale effectuée, les agents interagissent entre eux en suivant le protocole de [8]. Les interactions sont restreintes de telle sorte que les agents sont initialement biaisés vers plus d'interactions avec leurs parents : à sa i^{ime} itération, chaque agent a une probabilité P_i d'interagir avec un de ses parents choisi aléatoirement. La probabilité de restreindre l'interaction de l'agent dépend de i et du taux de réduction de l'interaction restreinte $\epsilon < 1$:

$$P_i = \max(0, 1 - i \times \epsilon)$$

Ainsi, le biais est maximal à la première interaction et diminue au fur et à mesure que les interactions augmentent. Cela imite les agents qui élargissent progressivement leurs cercles sociaux. Dans les expériences, nous utilisons $\epsilon = .01$.

5 Expériences

Nous réalisons deux expériences pour tester chacune des hypothèses introduites dans la section 3.4. Elles utilisent le même mode opératoire, seules quelques variables indépendantes diffèrent. Le tableau 1 résume les valeurs des paramètres considérés dans ces plans d'expérience. Les valeurs de τ se terminent par 1 afin de calculer les mesures à l'itération $\tau - 1$, c'est-à-dire avant que la population ne soit remplacée. Chaque combinaison de valeurs de paramètres est exécutée 10 fois. Pour chaque expérience, nous mesurons la correction (*accuracy*) de l'ontologie comme dans [8]. La taille de la population et le nombre d'itérations sont fixés à 40 et 200000 respectivement.

5.1 Expérience 1 : Effet de la transmission intergénérationnelle sur la correction des ontologies

La première expérience, vise à évaluer les effets de l'introduction de générations d'agents. Ainsi, nous nous concentrons sur les variables affectant la transmission verticale des connaissances : la proportion d'instances couvertes par l'échantillon d'entraînement et la durée de vie de la population.

En conséquence, nous faisons varier le pourcentage de transmission r qui correspond au degré de complétude et d'imperfection du transfert intergénérationnel. Lorsque le pourcentage de transmission est de 0, les agents commencent avec des ontologies aléatoires. Nous faisons également varier la longueur de la période τ qui correspond à la demi-vie des agents. Lorsque la période est plus longue que le nombre d'itérations ($\tau > n$), l'expérience se déroule en une génération (pas de transmission intergénérationnelle).

L'hypothèse H_1 peut donc être reformulée comme suit *L'ajout de la transmission intergénérationnelle entraîne une plus grande correction que la transmission intragénérationnelle seule.*

5.2 Expérience 2 : Interaction entre la sélection et la longueur de l'interaction

Dans cette deuxième expérience, nous testons les politiques moins sélectives. Plus précisément, nous cherchons à savoir si la transmission intragénérationnelle, un mécanisme typique de l'évolution culturelle, peut compenser la réduction ou l'absence de sélection des parents.

Cette expérience se concentre sur (1) la manière dont les parents sont sélectionnés pour la reproduction, et (2) la durée de vie d'une génération, car les agents ont besoin de temps pour se mettre d'accord sur les décisions à adopter. Ainsi, la politique de sélection des parents s prend les valeurs *random*, *income* et *best* et la durée de la période τ varie comme dans la première expérience. Nous réalisons également l'expérience avec des agents qui n'adaptent pas leurs connaissances après l'interaction ($ht = non$), écartant complètement la transmission horizontale.

L'hypothèse H_2 peut donc être testée comme suit *avec une transmission intragénérationnelle suffisante, la correction obtenue avec ou sans sélection est similaire.*

6 Résultats

Dans ce qui suit, les résultats des deux expériences sont présentés.

6.1 La transmission entre générations améliore la correction des ontologies

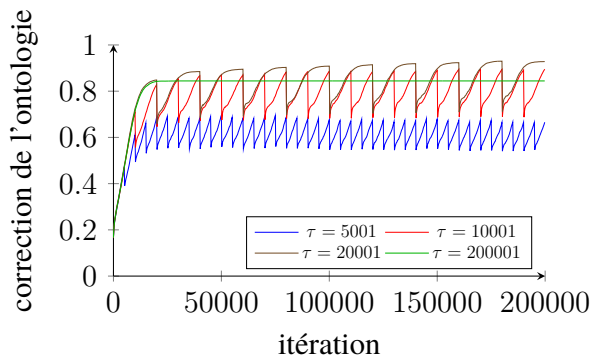
Pour tester l'hypothèse H_1 , nous comparons la correction (*accuracy*) finale moyenne des expériences avec une seule génération ($\tau = 200001$) aux expériences avec plusieurs générations (de différentes durées de période). La figure 4 montre cette évolution. En effectuant un test ANOVA (analyse de la variance) et un test post-hoc tuckey-hsd, la correction à la fin de l'expérience lorsque la transmission intergénérationnelle se produit à des périodes de longueur 10001 et 20001 est significativement plus élevée que lorsqu'elle ne se produit pas ($p \ll 0,01$). Ainsi, l'hypothèse H_1 est acceptée lorsque la période est suffisamment longue ($\tau \geq 10001$). **La transmission intergénérationnelle nécessite de longues périodes d'interaction.** On peut observer, dans les premières itérations de la figure 4, que chaque génération améliore sa correction par rapport à la précédente. En particulier, la correction obtenue à 2τ est strictement supérieure à la correction à τ . Cela confirme que la connaissance des agents avec une transmission verticale est plus correcte qu'avec la transmission horizontale seule. Cependant, lorsque la période d'interaction n'est pas assez longue, les agents n'ont pas le temps de diffuser largement les connaissances pertinentes. La transmission verticale souffre alors de la faible correction des connaissances transmises et la courte période ne permet que de s'en remettre. Ceci explique pourquoi, lorsque la longueur de la période est de 5001, la correction ne s'améliore pas.

Pourcentage de transmission et durée de la période interagissent. La figure 5 compare la correction des agents avec ($r \neq 0$) et sans ($r = 0$) transmission verticale initiale, sous différentes durées de période ($\tau = 5001$ et $\tau = 20001$). Lorsque la période est courte ($\tau = 5001$), un pourcentage de transmission plus élevé ($r = .8$) donne de meilleurs résultats qu'un pourcentage de transmission faible. Au contraire, avec une période longue ($\tau = 20001$), les meilleurs résultats sont obtenus sans transmission verticale initiale ($r = 0$), ceux avec transmission verticale initiale étant très proches les uns des autres.

Ceci s'explique par la capacité de la transmission intragénérationnelle à diffuser des connaissances

Signification	Variable	Expérience 1	Expérience 2
Durée de la période	τ	5001, 10001, 20001, 200001	5001, 10001, 20001
Pourcentage de transmission	r	0 = <i>random</i> , 20, 40, 60, 80, 100	40
Sélection des parents	s	<i>random</i>	<i>income, random, best</i>
Transmission horizontale	ht	<i>oui</i>	<i>oui, non</i>

TABLE 1 – Variables indépendantes et valeurs expérimentales.

FIGURE 4 – Correction moyenne (sur r) par longueur de période (τ).

plus correctes à l'ensemble de la population. Ces connaissances ont une chance d'être transmises même avec un faible r et même en l'absence de transmission verticale initiale ($r = 0$) car elles peuvent être transmises par interaction. Dans ce cas, un faible r fournit la variation permettant d'augmenter encore la correction. Au contraire, si la transmission intragénérationnelle est insuffisante (τ court), une transmission initiale plus fidèle ($r \geq 0.8$) est nécessaire pour préserver la correction. Ceci montre le délicat équilibre à trouver entre r et τ pour assurer l'amélioration des connaissances.

6.2 La transmission intragénérationnelle peut compenser l'absence de sélection

Pour tester l'Hypothèse H_2 , nous montrons d'abord que la sélection des parents sans transmission intragénérationnelle améliore effectivement la correction des connaissances. Ensuite, nous montrons que cet effet n'existe pas avec la transmission intragénérationnelle. Le tableau 2 résume ces résultats. Les résultats présentés ci-dessous sont ceux avec $\tau = 20001$, les mêmes résultats sont obtenus avec 5001 et 10001 (20001 est le cas le moins favorable).

La sélection est efficace. La figure 6 montre en pointillés l'évolution de la correction avec uniquement la transmission intergénérationnelle comparant les politiques de sélection maximale

$ht \setminus s$	<i>random</i>	<i>income</i>	<i>best</i>
<i>no</i>	aucun	moyen	moyen
<i>yes</i>	élevé	élevé	élevé

TABLE 2 – Amélioration de la correction en fonction de la sélection (s) et de la transmission horizontale (ht). En l'absence de transmission horizontale ($ht = non$), la sélection maximale et la sélection basée sur le revenu améliorent la correction finale; avec la transmission horizontale ($ht = oui$), toutes les stratégies fournissent une amélioration plus élevée.

(*best*), basée sur le revenu (*income*) et équiprobable (*random*). En l'absence de transmission intragénérationnelle, le fait d'avoir des parents aléatoires n'améliore pas la correction au fil des périodes, bien que la sélection des parents l'améliore. Le test ANOVA sur la correction finale des trois méthodes de sélection donne une différence significative ($p \ll 0,01$).

La transmission intragénérationnelle compense l'absence de sélection. La figure 6 montre que l'évolution de la correction lorsqu'il y a une transmission intragénérationnelle (lignes continues), est significativement plus élevée que lorsqu'elle n'est pas présente (lignes pointillées). De plus, contrairement à la transmission intergénérationnelle, lorsque la transmission intragénérationnelle est présente, la façon dont les parents sont sélectionnés a peu d'impact sur la correction finale. En présence de transmission intragénérationnelle ($ht = oui$), ANOVA ne retourne pas de différence significative entre les trois méthodes de sélection des parents ($p = 0.34$). Dans ce cas, la table 3 montre que la différence entre l'absence de sélection (*random*) et les politiques de sélection maximale et basée sur le revenu est proche de 0, bien qu'elle soit significativement plus importante sans transmission intragénérationnelle. Ainsi, nous acceptons l'hypothèse H_2 : la transmission intragénérationnelle compense l'absence de sélection.

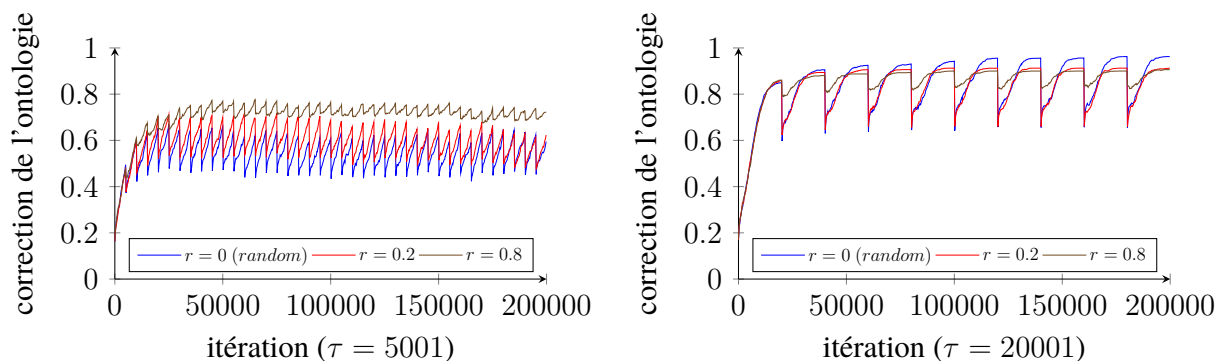


FIGURE 5 – Correction moyenne avec (rouge et marron) ou sans (bleu) transmission verticale initiale.

$ht \setminus s$	<i>income</i>	<i>best</i>
<i>non</i>	-0.115(35)	-0.165(35)
<i>oui</i>	-0.01(4)	0.005(35)

TABLE 3 – Intervalles de confiance à 95% de la différence moyenne entre la sélection équiprobable et les autres méthodes de sélection avec ($ht = oui$) et sans ($ht = non$) transmission intragénérationnelle.

7 Discussion

Ces résultats peuvent être lus comme une confirmation, en utilisant un cadre expérimental différent et des contraintes plus souples, les résultats de [1] sur l’efficacité de la transmission verticale pour améliorer la connaissance des agents. Ils peuvent également être considérés comme une réfutation de l’idée qu’une forte sélection des parents est un facteur important pour observer de tels effets. En effet, plus une culture est partagée dans une population, moins la sélection est importante. Les résultats obtenus montrent que la diffusion de connaissances de qualité nécessite du temps. Si les agents ont une durée de vie courte et qu’il n’y a pas de sélection, alors les connaissances ne s’amélioreront pas car les plus correctes auront peu de chance d’être transmises à la génération suivante. Mais s’ils ont suffisamment de temps pour diffuser des connaissances de qualité, alors celles-ci s’amélioreront au fil des générations sans sélection des parents.

Contrairement aux gènes, même si les parents ne fournissent pas les meilleurs atouts culturels, les enfants sont capables de les acquérir auprès de leurs pairs ou d’autres sources. [1] a montré que (1) la transmission intragénérationnelle peut introduire une variation dans la culture et (2) sa sélection se produit dans la trans-

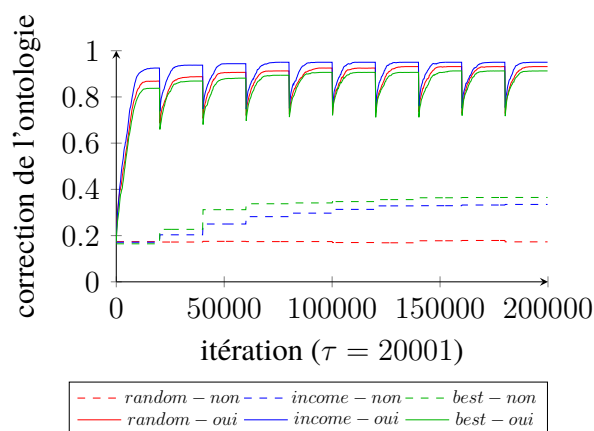


FIGURE 6 – Correction moyenne par mode de sélection avec (ligne continue) et sans (ligne discontinue) transmission horizontale.

mission intergénérationnelle. À l’inverse, nous avons montré comment (1) la transmission intergénérationnelle peut introduire les variations (qui permettent aux agents d’améliorer encore leur correction comme le montre la section 6.1) et (2) la transmission intragénérationnelle peut sélectionner les connaissances qui se répandent dans la population des agents (section 6.2). Ces résultats montrent la robustesse de l’évolution culturelle dans laquelle les deux modes de transmission peuvent s’équilibrer.

Disponibilité des données

Toutes les expériences ont été réalisées dans l’environnement logiciel *Lazy lavender*². Les paramètres, les résultats et les carnets d’analyse des données sont disponibles sur [6] et [7].

2. <https://gforge.inria.fr/moex/lazylav>

8 Remerciements

Ce travail a été partiellement soutenu par MIAI @ Grenoble Alpes (ANR-19-P3IA-0003).

Références

- [1] ACERBI, A., AND PARISI, D. Cultural transmission between and within generations. *Journal of Artificial Societies and Social Simulation* 9, 1 (2006), 1–16.
- [2] ANSLOW, M., AND ROVATOS, M. Aligning experientially grounded ontologies using language games. In *Actes 4e International Workshop on Graph Structures for Knowledge Representation and Reasoning* (2015), pp. 15–31.
- [3] ANZOLA, D., AND RODRÍGUEZ-CÁRDENAS, D. A model of cultural transmission by direct instruction : An exercise on replication and extension. *Cognitive Systems Research* 52 (2018), 450–465.
- [4] ATENCIA, M., AND SCHORLEMMER, M. An interaction-based approach to semantic alignment. *Journal of Web Semantics* 12 (2012), 131–147.
- [5] BORENSTEIN, E., AND RUPPIN, E. Enhancing autonomous agents evolution with learning by imitation. *Artificial Intelligence and Simulation of Behaviour* (2003).
- [6] BOURAHLA, Y. 20210601-DOTG : Knowledge is transmitted between agents and across generations, 2021. <https://sake.re/20210601-DOTG>.
- [7] BOURAHLA, Y. 20210927-DOTG : Knowledge is transmitted between agents and across generations with different parent selection methods, 2021. <https://sake.re/20210927-DOTG>.
- [8] BOURAHLA, Y., ATENCIA, M., AND EUZENAT, J. Knowledge improvement and diversity under interaction-driven adaptation of learned ontologies. In *Actes 20e AAMAS* (2021), pp. 242–250.
- [9] BOURAHLA, Y., ATENCIA, M., AND EUZENAT, J. Knowledge transmission and improvement across generations do not need strong selection. In *Actes 21e AAMAS* (2022). to appear.
- [10] BOYD, R., AND RICHERSON, P. *Culture and the evolutionary process*. The University of Chicago Press, Chicago, 1985.
- [11] BYRSKI, A., DREZEWSKI, R., SIWIK, L., AND KISIEL-DOROHINICKI, M. Evolutionary multi-agent systems. *The Knowledge Engineering Review* 30, 2 (2015), 171–186.
- [12] CAVALLI-SFORZA, L. L., AND FELDMAN, M. W. *Cultural transmission and evolution : a quantitative approach*. Princeton University Press, New Jersey, 1981.
- [13] DENARO, D., AND PARISI, D. Cultural evolution in a population of neural networks. In *Actes 8th Italian Workshop on Neural Nets* (1996), pp. 100–111.
- [14] EUZENAT, J. Interaction-based ontology alignment repair with expansion and relaxation. In *Actes 26e IJCAI* (2017), pp. 185–191.
- [15] GERARD, S., AND SINGH, M. Evolving protocols and agents in multiagent systems. In *Actes 12e AAMAS* (2013), pp. 997–1004.
- [16] JOLLEY, B., BORG, J. M., AND CHANNON, A. Analysis of social learning strategies when discovering and maintaining behaviours inaccessible to incremental genetic evolution. In *Actes International Conference on Simulation of Adaptive Behavior* (2016), pp. 293–304.
- [17] NOLFI, S. Emergence of communication in embodied agents : Co-adapting communicative and non-communicative behaviours. *Connection Science* 17, 3–4 (2005), 231–248.
- [18] REYNOLDS, R. An introduction to cultural algorithms. In *Actes 3e Annual Conference on Evolutionary Programming* (1994), pp. 131–139.
- [19] RICHERSON, P., AND BOYD, R. *Not By Genes Alone : How Culture Transformed Human Evolution*. The University of Chicago Press, Chicago, 2005.
- [20] SEN, S., AND AIRIAU, S. Emergence of norms through social learning. In *Actes 20e IJCAI* (2007), pp. 1507–1512.
- [21] STEELS, L. *Experiments in cultural language evolution*. John Benjamins Publishing Company, Amsterdam, 2012.
- [22] VAN DIGGELEN, J., BEUN, R., DIGNUM, F., VAN EIJK, R., AND MEYER, J. Anemone : An effective minimal ontology negotiation environment. In *Actes 5e AAMAS* (2006), pp. 899–906.

Session 4 : Émergence, auto-organisation

P-ADRIP : un système multi-agent auto-organisateur pour la prévision du trafic routier

Ha-Nhi Ngo^{a,b} Elsy Kaddoum^a Marie-Pierre Gleizes^a
ha-nhi.ngo@irit.fr elsy.kaddoum@irit.fr marie-pierre.gleizes@irit.fr

Jonathan Bonnet^b Anaïs Goursolle^b
jonathan.bonnet@continental-corporation.com anais.goursolle@continental-corporation.com

^aIRIT- Université de Toulouse, CNRS, Toulouse INP, UT3, UT1, UT2

^bContinental Digital Services France, Toulouse, France

Résumé

La prévision du trafic suscite de plus en plus d'intérêt dans les recherches académiques et industrielles. Les modèles basés sur les séries temporelles modélisent la dépendance linéaire mais décrivent difficilement les propriétés non linéaires et complexes des données du trafic. Récemment, de nombreuses méthodes de prévision du trafic basées sur des approches d'apprentissage automatique et d'apprentissage profond ont été proposées. Cependant, ces modèles coûteux en temps de calcul, nécessitent des volumes de données importants et leurs résultats soulèvent des questions de fiabilité et de faisabilité. En effet, la prévision du trafic est une tâche très difficile en raison des corrélations spatiales complexes dans le réseau routier, de la dépendance temporelle et de la difficulté de la prédiction à long terme. Pour s'adresser aux défis mentionnés, nous proposons le système P-ADRIP (Prediction subsystem - Adaptive multi-agent system for DRIVING behaviors Prediction) basé sur les systèmes multi-agents. P-ADRIP vise à fournir une prévision dynamique et temps réel du trafic. Les expériences menées montrent la performance remarquable de P-ADRIP par rapport aux méthodes de prédiction connues dans l'état de l'art.

1 Introduction

La prévision précise et en temps réel du trafic est aujourd'hui essentielle pour le contrôle du trafic urbain, la sécurité et les fonctions de guidage du Système de Transport Intelligents (STI). Elle est largement appliquée à divers services de transport afin de fournir de meilleures recommandations de voyage, d'atténuer les embouteillages et les collisions dangereuses en queue du bouchon, d'économiser l'énergie consommée

et d'améliorer l'efficacité du trafic. Ainsi, la prédiction précise devient indispensable pour de nombreux sous-systèmes de STI, notamment les systèmes avancés d'aide à la conduite (ADAS), les systèmes avancés d'information aux voyageurs (ATIS), *etc.*

Cette tâche est difficile en raison de la complexité des dépendances spatio-temporelles et de la difficulté de la prédiction à long terme. D'une part, les données de trafic montrent qu'une forte dynamique temporelle entraîne une haute dépendance des valeurs prédites par rapport aux données historiques de trafic, i.e une dépendance temporelle. La dépendance temporelle est caractérisée par différentes propriétés des séries temporelles telles que : la non-stationnarité, la saisonnalité, la non-linéarité, *etc.* D'autre part, la dynamique du trafic présente également une corrélation spatiale complexe.

Récemment, les Systèmes Multi-Agents (SMA) ont été appliqués avec succès à de nombreux problèmes dans divers domaines, notamment dans le transport intelligent [1] ou l'apprentissage par démonstrations pour des robots ambiants [2], grâce à leur capacité à modéliser des problèmes complexes. Comme le problème de la prévision du trafic peut être considéré comme un problème complexe en raison du nombre de facteurs qui interviennent, un système basé sur un SMA est une approche prometteuse à envisager.

Cet article présente un système basé sur un SMA [3] pour prédire la dynamique du trafic, appelé P-ADRIP (Prediction subsystem - Adaptive multi-agent system for DRIVING behaviors Prediction). P-ADRIP met en oeuvre une stratégie de prédiction en temps réel et à long terme qui garantit que la prédiction de la dynamique du trafic est fournie et mise à jour tout au long de la vie du système.

2 État de l'art

Les recherches actuelles portent principalement sur les paramètres fondamentaux du flux de trafic tels que : la vitesse moyenne, le volume ou la densité avec différents horizons de prédiction. Ces travaux sont principalement classés en deux catégories : les modèles paramétriques et les modèles non-paramétriques. Pour chaque catégorie, nous présentons ci-dessous les méthodes les plus utilisées [4].

2.1 Modèles paramétriques

Les modèles paramétriques sont basés sur les modèles de séries temporelles. Les modèles ARMA (Auto-Regressive Moving Average) sont bien adaptés pour la prédiction du trafic à court terme [5]. Certaines variantes d'ARMA sont développées pour résoudre des problèmes de non-stationnarité, saisonnalité ou de dépendance spatio-temporelle.

Les modèles ARMA supposent que les informations futures du trafic ne dépendent linéairement que des valeurs précédentes (Auto-Régressif - AR) et de bruits aléatoires (Moyenne Mobile - MA) [6]. ARIMA (Auto-Regressive Integrated Moving Average), qui est une extension du modèle ARMA peut traiter des non-stationnarités en considérant les différences entre les observations consécutives. ARIMA a été appliqué de façon prometteuse dans [6] pour étudier la prédiction du temps de trajet dans les artères d'une ville. Dans [7], les auteurs ont appliqué un sous-ensemble d'ARIMA pour la prévision à court terme du volume de trafic. Le problème de saisonnalité a été traité avec SARIMA Seasonal (Auto-Regressive Integrated Moving Average). Dans [8], la base théorique de la prédiction de l'état du trafic à l'aide de processus SARIMA est présentée. Le modèle SARIMA est également appliqué pour prévoir le flux du trafic [9] combiné à un modèle de lissage exponentiel. Selon les résultats obtenus, l'intégration des composantes saisonnières améliore la performance du modèle. Le modèle VARMA (Vector AutoRegressive Moving Average) est un modèle multivarié qui a été appliqué dans [10] pour considérer l'impact des mesures des segments voisins sur le segment actuel. Les résultats ont montré une amélioration significative de la performance de prédiction lors de l'utilisation d'un modèle de série temporelle multivariée dans les cas d'un grand réseau routier et d'un nombre élevé de détecteurs de boucles.

Malgré l'amélioration des modèles paramétriques pour traiter les propriétés spécifiques des données de trafic, leur principal inconvénient est qu'ils ne peuvent pas résoudre les problèmes non linéaires. Ils obtiennent de bonnes performances lorsque le trafic présente des variations régulières, mais l'erreur de prédiction devient significative lorsque des changements de situations se produisent.

2.2 Modèles non paramétriques

Les modèles non paramétriques utilisent des données historiques pour entraîner des modèles et calibrer des coefficients qui expriment l'impact des variables sur les valeurs futures. Parmi les nombreux modèles non paramétriques développés, nous nous concentrons sur les modèles similaires à P-ADRIIP, comme le modèle des K-plus proches voisins (KNN), et qui répondent aux défis importants de l'analyse des séries temporelles.

KNN cherche les k historiques états de trafic les plus proches de l'état actuel du trafic et prédit les états futurs en observant les états suivants des k voisins. Dans [11], KNN a été appliqué pour la première fois à la prédiction du flux de trafic. Une version améliorée de KNN pour la prédiction du trafic a été introduite dans [12]. La grande précision obtenue montre la faisabilité de la méthode proposée pour la prédiction du flux de trafic à court terme sans contraintes de données. Cependant, le choix de la valeur optimale des paramètres du modèle pose la question de la robustesse. Le type simple de réseau de neurones (NN) appelé **le réseau de neurones à propagation avant (FFNN)** est appliqué dans [13] pour estimer le flux de trafic. Les auteurs de [14] ont utilisé un NN avec plusieurs couches cachées pour prédire la vitesse du trafic. Les résultats obtenus montrent que l'erreur quadratique moyenne diminue de 14% par rapport à une seule couche cachée. Les réseaux neuronaux sont flexibles et permettent d'intégrer les variables environnementales et d'autres informations sur le trafic ou les paramètres du trafic de la route voisine ou du capteur. Cependant, la capture des dépendances complexes et à long terme des séries temporelles de données de trafic est un grand défi pour les NN en raison de la complexité croissante du modèle. Les réseaux neuronaux récurrents (RNN) et leurs variantes, telles que le réseau **Long short-term memory (LSTM)** et le réseau **Gated Recurrent Unit (GRU)**, constituent un autre type de réseau neuronal capable de traiter la dépendance à long terme des séries

temporelles. Les expériences dans [15], [16] ont montré que la performance de LSTM et GRU surpasse la performance d'ARIMA, NN et RNN traditionnels.

2.3 Discussion

L'état de l'art ci-dessus souligne les points suivants : premièrement, la complexité de la prédiction du trafic à long terme est affirmée par la grande dépendance temporelle et spatiale. Deuxièmement, la plupart des méthodes existantes ne font pas évoluer leur modèle tout au long de la vie pour s'adapter à de nouvelles perceptions. Troisièmement, le passage à l'échelle est difficile de par l'exigence élevée de calcul. Il est nécessaire d'associer à ces méthodes, un mécanisme de stockage et de manipulation efficaces. Finalement, les recherches existantes se concentrent sur la prédiction des paramètres fondamentaux du trafic tels que : la vitesse moyenne ou la densité moyenne du trafic dans une fenêtre temporelle (ex.5 minutes). Cependant, ces paramètres ne sont pas suffisants pour représenter finement la dynamique du trafic ou les comportements de conduite sur un segment de route. Par exemple, nous ne pouvons pas déduire la variation de la vitesse des véhicules sur un segment de route entier.

Les limitations des approches actuelles incitent à aborder la prévision du trafic par SMA. En effet, les SMA ont montré leur efficacité dans la résolution de problèmes complexes dans des environnements dynamiques et non linéaires en se basant sur la distribution des tâches entre les agents autonomes et coopératifs. De plus, les agents peuvent adapter localement leurs décisions pour s'adapter à des situations spécifiques grâce aux capacités d'auto-organisation. D'ailleurs, la décentralisation des SMA permet de paralléliser le calcul au niveau local, entraînant la réduction du temps de traitement pour les applications en grande échelle. Grâce à ces propriétés, nous considérons que les SMA sont pertinents pour représenter la dynamique du trafic. Dans la suite, nous proposons P-ADRIP - un SMA pour la prédiction du trafic.

3 Le système multi-agent ADRIP

3.1 Architecture d'ADRIP

ADRIP (Adaptive multi-agent system for Driving behaviors Prediction) vise à traiter le problème de la prévision dynamique du trafic qui

peut être décrit comme :

- Un ensemble de véhicules $V = v_1; v_2; \dots; v_m$. Chaque véhicule suit un itinéraire I constitué d'une séquence de segments de route notés $I = \{rds_1, \dots, rds_N\}$.
- Un ensemble de segments de route déterminés selon le réseau routier dans Open Street Map (OSM) ou des scénarios routiers simulés. Chaque segment de route est caractérisé par une position de départ et une position d'arrivée qui peuvent être localisées par des dispositifs GPS.

ADRIP est décomposé en deux sous-systèmes (L-ADRIP et P-ADRIP) qui fonctionnent en parallèle et en temps réel :

- L-ADRIP [17] : le sous-système d'apprentissage qui permet de classer dynamiquement **les comportements de conduite** des véhicules sur chaque segment de route, il est basé sur le flux de données perçu à partir des véhicules.
- P-ADRIP : le sous-système de prédiction qui est l'objet de cet article, prédit les comportements de conduite sur chaque segment de route pour un horizon de prédiction donné en utilisant la base des comportements de conduite appris.

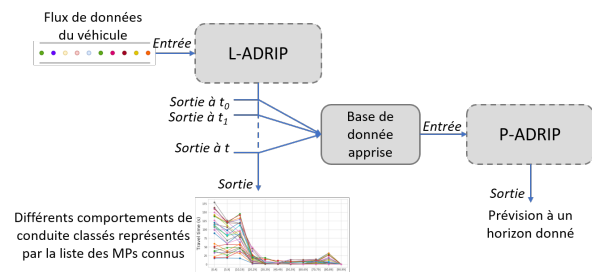


FIGURE 1 – Illustration d'ADRIP

L'architecture globale d'ADRIP est présentée dans la figure 1. Tout au long de son activité, le sous-système L-ADRIP fournit et met à jour la base de données locale de chaque segment de route. Cette base de données contient différents comportements de conduite appelés **profils de mobilité (MP)**. Le MP est défini comme la distribution du temps de passage sur le segment par rapport à différentes plages de vitesse. Chaque MP a des intervalles de temps associés appelés **plages d'utilisation (RU)** pendant lesquels les véhicules se sont déplacés à une certaine vitesse. Chaque MP a une liste de RU indiquant les moments où les véhicules d'une flotte se sont déplacés sur ce segment de route avec ce MP. La base de données apprise de L-ADRIP est composée

des différentes bases de données locales apprises pour chaque segment. Ce mécanisme d'apprentissage est réalisé par un SMA adaptatif décrit dans [17].

Le sous-système P-ADRIP perçoit la situation actuelle et utilise des bases de données apprises pour prédire les comportements de conduite futurs sur l'ensemble du réseau pour un horizon de temps donné. Les prédictions de P-ADRIP sont mises à jour à chaque fois que des changements se produisent dans la situation actuelle ou dans la base de données apprise. Dans P-ADRIP, la prévision du trafic sur un réseau routier est effectuée par un SMA composé d'agents segment (SA). Chaque agent SA est associé à un segment de route du réseau routier et prédit le trafic sur ce segment.

3.2 Formalisation du problème de prédiction locale

Pour envisager la prévision du trafic sur un réseau routier, ADRIP distribue la tâche globale au niveau local. Le problème de prédiction locale est défini comme la prédiction du trafic sur chaque segment de route du réseau par un agent SA. Pour cela, deux aspects majeurs sont considérés : la *corrélacion spatiale* et la *dépendance temporelle*.

Premièrement, la corrélation spatiale est prise en compte puisque la dynamique du trafic d'un segment de route est influencée par celle des segments de route voisins, en particulier la dynamique du trafic des segments de route en amont et en aval. Dans ADRIP, les agents SA communiquent avec leurs voisins pour demander les informations nécessaires à leur prédiction locale. Les informations des SA voisins constituent une partie de l'entrée de l'algorithme de prédiction défini comme une **Configuration**.

Definition 1 La configuration à l'instant T du point de vue de l'agent SA est l'ensemble des MP et de leurs RU correspondants à T des agents SA voisins.

Deuxièmement, la dépendance temporelle est intégrée en analysant la base de données obtenue par l'apprentissage de L-ADRIP. La figure 2 montre la description de la base de données d'un agent SA obtenue par L-ADRIP. Les $SA_1, \dots, SA_i, \dots, SA_N$ représentent l'ensemble des agents SA associés à l'ensemble des segments de route. Après le pro-

cessus d'apprentissage, chaque SA_i peut observer la liste des différents comportements de conduite représentés par la liste des MP : $\{MP_{i1}, \dots, MP_{ij}, \dots, MP_{iLi}\}$ avec L_i étant le nombre de MP de SA_i . Chaque $MP_{i,j}$ possède une liste de RU : $\{RU_{i,j,1}, \dots, RU_{i,j,k}, \dots, RU_{i,j,K_{i,j}}\}$ avec $K_{i,j}$ le nombre de RU de $MP_{i,j}$.

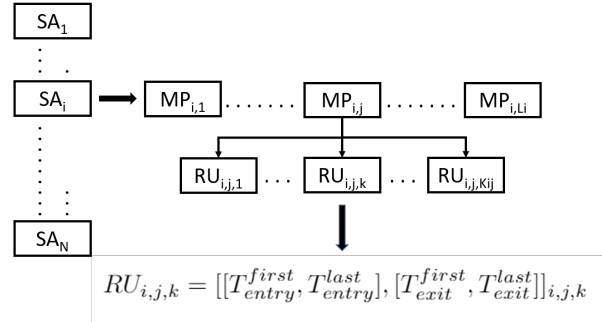


FIGURE 2 – Sortie de L-ADRIP

4 Les agents segment de P-ADRIP

Le comportement d'un agent SA est décomposé en deux parties : le comportement nominal et le comportement coopératif. Le comportement nominal consiste en l'ensemble des fonctions qui sont exécutées localement par les agents SA pour poursuivre leur but individuel. Le comportement coopératif définit l'ensemble des solutions proposées par les agents SA pour régler des situations spécifiques qui perturbent le comportement nominal appelées **des situations de non-coopération**.

4.1 Comportement nominal

Le **comportement nominal** d'un agent SA dans P-ADRIP consiste à calculer l'enchaînement prédictif des différents changements de MP pour un horizon de prévision donné (20 minutes par exemple). À partir de l'instant courant (T_s), l'agent SA lance l'algorithme de prédiction dont les principales étapes sont illustrées à la figure 2 pour prédire le prochain MP et le timestamp auquel le changement de profil (MP) va se produire. Ce timestamp est également le début du prochain MP prédit. Si ce timestamp n'atteint pas l'horizon de prévision demandé, l'agent SA lance à nouveau l'algorithme de prédiction en utilisant comme entrées le dernier MP calculé et son timestamp associé. Ce processus est répété jusqu'à ce que l'agent SA atteigne au moins l'horizon de prévision demandé.

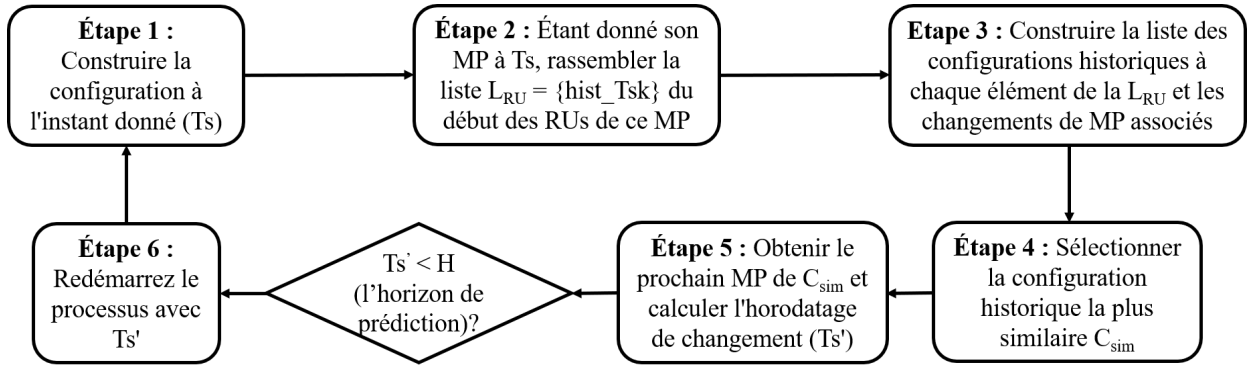


FIGURE 3 – Étapes principales du processus de prédiction d’un agent SA

A l’étape 4, l’agent SA compare la configuration à T_s et les configurations historiques pour sélectionner la plus similaire. Ainsi, pour chaque configuration historique calculée à l’étape 3, l’agent SA la compare à la configuration à T_s et calcule deux critères :

1. Le premier critère est le nombre d’agents SA voisins avec des MPs différents parmi les deux configurations.
2. Le deuxième critère se réfère à la comparaison de l’écart temporel entre le début du MP des SAs voisins et le début du MP du SA considéré. En effet, puisque le changement de dynamique du trafic sur un segment de route peut être entraîné par des changements de dynamique du trafic sur ses voisins, ce critère vise à évaluer la différence du temps de propagation de la dynamique entre deux configurations. La figure 3 illustre le calcul de ce critère.

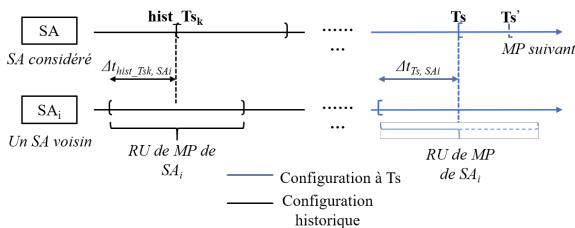


FIGURE 4 – Illustration du calcul de l’écart temporel du second critère

Pour chaque SA_i appartenant à l’ensemble des agents SA voisins, $\Delta t_{T_s,SA_i}$ est la distance temporelle entre le début du MP sur SA (T_s) et le début du MP de SA_i observé à T_s . De même, $\Delta t_{hist_T_{s_k},SA_i}$ est la distance temporelle entre le début de MP de SA à un RU historique ($hist_T_{s_k}$)

et le début du MP de SA_i observé à $hist_T_{s_k}$. Ainsi, le deuxième critère s’exprime comme :

$$c_2(Conf_{T_s}, Conf_{hist}) = \sum_{i=1}^M |\Delta t_{T_s,SA_i} - \Delta t_{Hist_T_{s_k},SA_i}| \tag{1}$$

où M est le nombre d’agents SA voisins.

La configuration historique la plus similaire avec la configuration à T_s est celle qui minimise les deux critères. À l’étape 5, le changement de MP associé à la configuration historique la plus similaire constitue le prochain MP prédictif de SA à T_s' . T_s' est calculé en décalant T_s vers l’avant pour un intervalle de temps égal à la taille du RU associé à la configuration choisie. Enfin, si T_s' atteint l’horizon de prédiction, l’agent SA arrête l’algorithme, sinon, il relance le processus calculant la prédiction à T_s' et stocke le prochain MP dans la chaîne des différents changements prédictifs de MP.

4.2 Comportement coopératif

Au cours de leur comportement nominal, les agents SA rencontrent des situations spécifiques qui perturbent leurs performances. Les comportements coopératifs sont définis pour faire face à ces situations spécifiques comme :

- **Incompétence** un agent SA ne peut pas construire la configuration complète requise à l’étape 1 car certains voisins n’ont pas encore estimé leur prédiction à l’instant donné, et ne peuvent donc pas l’envoyer à l’agent SA.
Solution proposée : un agent SA attend que les agents SA voisins qui ont un horizon de prédiction plus court que

le sien lancent leur processus de prédiction avant. Lorsque les prédictions de ces agents SA sont mises à jour et que l’horizon de prédiction du SA devient plus court que celui de ses agents SA voisins, il peut construire la configuration complète requise et commencer son processus de prédiction. Il s’agit d’auto-organisation qui permet de réorganiser automatiquement et de manière adaptative les agents en fonction des différentes perceptions.

- **Improductivité** un agent SA perçoit un nouveau MP et l’agent SA ne peut pas trouver des successeurs de ce MP car il n’a jamais été perçu ou appris par L-ADRIP.

Solution proposée P-ADRIP reste sur ce nouveau MP pour la prédiction jusqu’à l’horizon de prédiction. En parallèle, le processus d’apprentissage dynamique de l’agent SA (réalisé par L-ADRIP) intègre ce nouveau MP dans sa base de données locale et continue à apprendre des MP perçus. Si l’agent SA détecte que le MP prévu n’est pas celui appris par L-ADRIP, le mécanisme d’auto-correction de SA corrige la prédiction. Ce mécanisme est réalisé par une fonction de contrôle de P-ADRIP qui sera étudiée dans la prochaine étape du projet.

5 Expérimentations

Dans cette section, nous présentons les expérimentations pour valider le comportement de P-ADRIP et le comparer avec des modèles connus de prédiction de trafic.

5.1 Génération de données

Les expériences sont réalisées sur la plateforme GAMA (GIS Agent-based Modelling Architecture) [18]. GAMA permet de charger des réseaux routiers à partir de fichiers de forme ou de fichiers d’OSM (Open Street Map) pour réaliser la simulation sur un réseau routier réel. Pour cette étude, le scénario se situe sur le campus de l’Université Toulouse III Paul Sabatier à Toulouse et comprend 63 segments de route (cf. figure 5). Les caractéristiques des agents SA sont extraites de l’OSM [19]. Le comportement des véhicules est défini selon le modèle proposé par [20] nommée Advanced Driving Skill. Le nombre de véhicules à chaque instant de la simulation est fixé à 200 ce qui permet de présen-

ter une diversité de dynamiques de trafic. Dès qu’un véhicule termine sa trajectoire, il est remplacé par un nouveau afin que le nombre total de véhicules pendant la simulation soit toujours de 200. La position de démarrage de chaque véhicule et sa trajectoire sont choisies aléatoirement. Ainsi, bien que le nombre total d’agents soit fixe, le flux de trafic sur chaque segment de route varie dans le temps, entraînant des dynamiques de conduite différentes. Le trafic a été simulé sur ce scénario pendant 3 heures pour l’échantillon d’apprentissage et ensuite pendant 1 heure pour l’échantillon de test, totalisant 9520 trajectoires de véhicules.

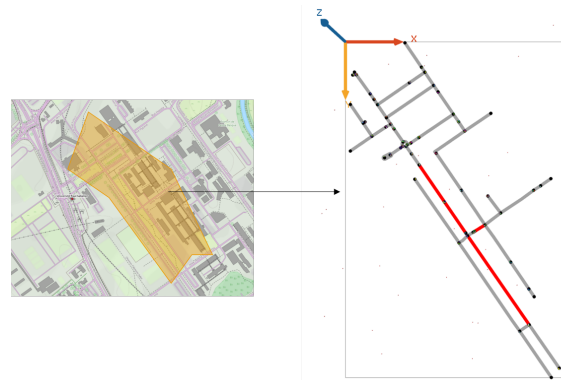


FIGURE 5 – Le scénario de l’OSM (à gauche) et sa projection dans GAMA

5.2 Prétraitement des données

À partir de la simulation, nous obtenons les données de la trajectoire du véhicule contenant la vitesse à chaque position GPS identifiée par la longitude et la latitude. Les méthodes présentées dans l’état de l’art sont testées en utilisant les données formant la série temporelle de la vitesse moyenne sur les segments de route à chaque f minutes (f est la fréquence de la série temporelle). Ainsi, nous faisons la moyenne des vitesses de tous les véhicules sur chaque segment de route toutes les 10 secondes pour obtenir des séries temporelles de vitesse moyenne à une fréquence de 10s. De plus, les données collectées doivent également être normalisées à l’aide de *MinMaxScaler* pour éviter de dégrader les performances des modèles d’apprentissage lors de l’optimisation du gradient pour FFNN, LSTM, GRU et pour assurer un poids égal à chaque variable dans la mesure de la distance pour KNN. En outre, la fonction de transformation *StandardScaler* est appliquée pour les modèles VARMA et ARIMA afin de garantir l’échelle cohérente des données dans différentes séries tem-

porelles. Ces fonctions sont implémentées dans la bibliothèque *sklearn* de Python.

Pour ADRIP, les données de trajectoire des véhicules sont segmentées en fonction du réseau routier et ces données segmentées sont transformées en MP. Le MP est la distribution du temps de parcours sur 7 plages de vitesse qui sont [0%, 5%], [5%, 10%], [10%,20%], [20%, 30%], [30%, 40%], [40%, 60%], [60%, 200%] de la limite de vitesse du segment de route. La dernière plage de vitesse est étendue arbitrairement à 200% de la vitesse limitée afin de couvrir certains dépassements de vitesse qui se produisent parfois. Nous observons la vitesse du véhicule au cours du temps et comptons le temps de déplacement correspondant à chaque plage de vitesse. Ces MP sont construits localement par le véhicule puis sont communiqués et traités par L-ADRIP pour apprendre les profils historiques du segment considéré.

5.3 Paramètres

Pour **ARIMA**, les paramètres sont (30, 0, 1) présentés dans l'article [21] sauf le nombre de valeurs précédentes utilisées que nous avons fixé à 30 pour s'adapter à notre plan d'expérience. Le modèle est implémenté en utilisant la librairie Python *statsmodel*.

Pour **VARMA**, le modèle est appliqué avec un nombre de lags fixé à 30 et est mis en oeuvre à l'aide de la librairie Python *statsmodel*.

Pour **KNN**, la valeur optimale de k (nombre de voisins les plus proches) est fixée à 18 selon [11].

Pour **FFNN**, le réseau a 2 couches cachées, chaque couche contient 256 unités. Les paramètres du réseau sont issus de [21].

Pour les modèles **LSTM**, **GRU**, les cellules du modèle sont implémentées en utilisant la librairie *keras.layers.recurrent*. Ces modèles ont 2 couches cachées avec 64 unités et 1 couche dense avec 30 unités.

5.4 Critères d'évaluation

Pour comparer la performance des méthodes considérées pour la prédiction de la vitesse moyenne, 2 métriques sont adoptées : MAE et RMSE (Racine de l'erreur quadratique moyenne).

1. Le code Python se trouve à l'adresse suivante : <https://github.com/xiaochus/TrafficFlowPrediction>.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| ; RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (2)$$

où y_i est la valeur réelle, \hat{y}_i est la valeur prédite et N est le nombre de points de données.

De plus, pour évaluer la performance de ADRIP pour la prédiction des MP, nous définissons la métrique de la **différence de MP**.

$$MPDiff(MP^k, MP^l) = (|MP_i^k - MP_i^l|)_{i=1, \dots, N} \quad (3)$$

où N est le nombre de plages de vitesse, MP_i^k and MP_i^l sont les valeurs du temps de déplacement correspondant à la plage de la i ème vitesse.

5.5 Résultats et analyse

Nous comparons d'abord ADRIP avec les méthodes choisies pour la prédiction de la vitesse moyenne pour les 5 prochaines minutes. Les méthodes comparées prennent en entrée la vitesse moyenne des 5 dernières minutes. Bien que l'horizon de prédiction ne soit que de 5 minutes, notre expérience peut être considérée comme un test à long terme en ce qui concerne la fenêtre de données à 10s puisque cet horizon correspond aux 30 prochains points de données. Contrairement aux tests réalisés dans d'autres articles, nous visons à obtenir toutes les valeurs prédictives entre le timestamp actuel et les 5 prochaines minutes. Ainsi, la variable de réponse est un vecteur de taille 30.

Nous sélectionnons 4 segments de route (montrés en rouge sur la figure 5) du scénario de test. Ces segments possèdent la plus grande diversité de dynamiques de trafic. En effet, ces segments de route sont intéressants pour l'évaluation des méthodes de prévision du trafic car ils montrent l'aptitude des méthodes à capturer l'évolution complexe des données de trafic. Le tableau I résume les résultats obtenus. Notez que la prédiction obtenue par ADRIP est le MP, ainsi, les vitesses moyennes prédites à chaque 10s sont déduites en utilisant l'expression suivante :

$$\bar{v}_t = \frac{l}{\sum_{i=1}^N MP_t^i} \quad (4)$$

où \bar{v}_t est la vitesse moyenne à l'instant t , l est la longueur du segment de route, N est le nombre

	Métrique	VARMA	ARIMA	KNN	FFNN	LSTM	GRU	ADRIP
Moyen sur 4 segments	MAE	1.51	1.51	1.50	1.59	1.59	1.72	1.31
	RMSE	2.42	2.36	2.40	2.54	2.52	2.69	1.77
1er segment	MAE	1.05	1.09	0.96	1.02	0.95	0.96	1.00
	RMSE	1.80	1.68	1.62	1.72	1.59	1.72	1.41
2ème segment	MAE	1.10	1.16	1.08	1.13	1.03	1.05	0.91
	RMSE	2.15	2.10	2.09	2.11	2.05	2.05	1.42
3ème segment	MAE	2.44	2.41	2.58	2.80	2.95	3.49	1.80
	RMSE	3.82	3.77	4.05	4.42	4.57	5.28	2.40
4ème segment	MAE	1.43	1.40	1.38	1.41	1.42	1.38	1.52
	RMSE	1.91	1.90	1.82	1.89	1.88	1.82	1.83

TABLE 1 – Evaluation des méthodes de l'état de l'art et ADRIP sur les 4 segments de route

de plages de vitesse dans le MP et MP_t^i est le temps de déplacement du véhicule dans la plage de la i ème vitesse du MP à l'instant t . La somme $\sum_{i=1}^N MP_t^i$ donne le temps de déplacement total lorsque les véhicules se déplacent avec le MP_t .

Premièrement, les erreurs moyennes sur 4 segments de route montrent que P-ADRIP obtient la meilleure performance moyenne pour les deux paramètres d'évaluation, ce qui témoigne de l'efficacité de la modélisation de la dépendance spatio-temporelle. Deuxièmement, KNN - un modèle non paramétrique simple - est légèrement meilleur que les modèles paramétriques (ARIMA et VARMA), ce qui souligne l'importance de la prise en compte de la dépendance non linéaire. Troisièmement, FFNN et les modèles basés sur RNN (LSTM, GRU) donnent de moins bonnes performances que KNN et les modèles linéaires. Cela peut venir de la dégradation causée par la complexité du modèle ou à l'insuffisance de l'ensemble des données d'apprentissage. Cette remarque renforce l'avantage de P-ADRIP comme une solution efficace pour la dépendance spatio-temporelle complexe dans la prévision du trafic avec une taille raisonnable de l'ensemble de données.

Ensuite, en considérant le détail de la performance du modèle sur chaque segment de route, nous observons les remarques suivantes. Pour les 2ème et 3ème segments de route, P-ADRIP surpasse les méthodes de l'état de l'art pour les deux métriques d'évaluation, ce qui est cohérent avec l'interprétation des résultats moyens. Cependant, pour les 1er et 4ème segments de route, P-ADRIP donne parfois un RMSE plus faible mais un MAE plus élevé par rapport aux autres modèles. Puisque RMSE pénalise plus sévèrement les grandes erreurs que MAE, nous pouvons déduire de ce phénomène que P-ADRIP fait plus de petites erreurs mais moins de grandes erreurs que les autres méthodes.

Ensuite, nous évaluons la précision de prédiction de P-ADRIP pour différents horizons. La figure 6 montre l'erreur de prédiction moyenne pour chaque plage de vitesse sur les 4 segments de route choisis pour les horizons à 2 minutes, 3 minutes, 4 minutes et 5 minutes. Nous observons que P-ADRIP n'augmente pas fortement l'erreur de prédiction lorsque l'on étend l'horizon de prédiction. Ce résultat permet de confirmer la capacité d'ADRIP pour les prévisions de trafic à long terme.

6 Conclusions et perspectives

Dans cet article, nous proposons le SMA P-ADRIP qui fournit des prédictions de trafic sur tout un réseau routier à partir de profils de mobilités appris en temps réel sur ses segments. Avec L-ADRIP introduit dans [17], P-ADRIP complète la fonctionnalité globale d'ADRIP. P-ADRIP montre sa capacité à traiter les problèmes de dépendance spatio-temporelle de la prévision du trafic en se basant sur deux mécanismes principaux. Premièrement, la dépendance temporelle est étudiée en analysant la base de données historique locale apprise pour chaque segment de route. Deuxièmement, l'analyse de la dépendance spatiale est basée sur l'échange d'informations avec les segments routiers voisins.

P-ADRIP a été comparé aux méthodes les plus utilisées sur la plateforme GAMA. Le test de comparaison pour la prédiction de la vitesse moyenne montre que P-ADRIP est meilleur dans le cas de forte variation de trafic mais moins efficace dans le cas de variations de trafic réduites. L'erreur de prédiction dépend du choix du seuil de similarité, pour lequel nous effectuons une étude de sensibilité. De plus, ADRIP peut fournir la prédiction des profils de mobilité qui est une représentation plus fine de la dynamique du trafic que la vitesse moyenne. Le test de prédic-

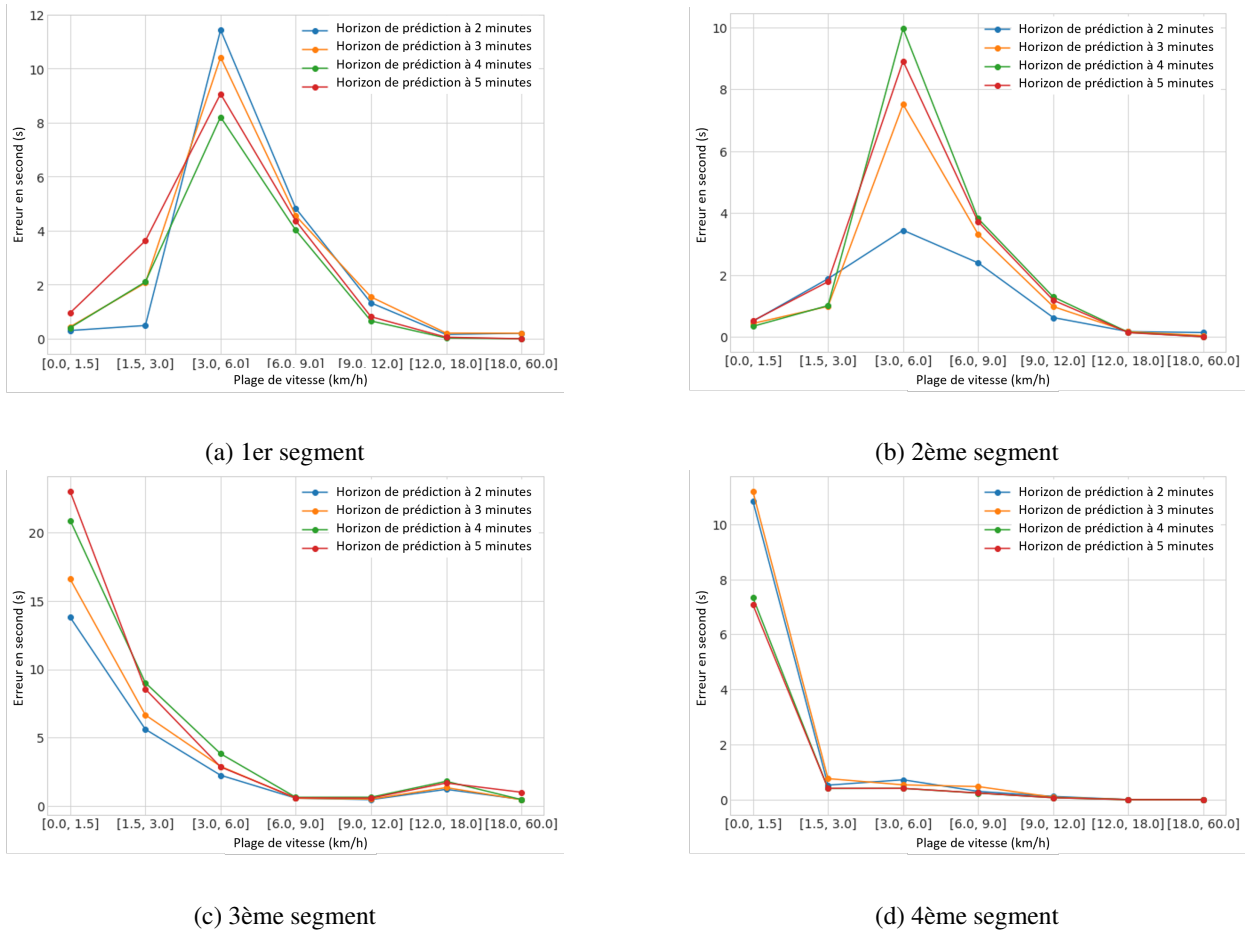


FIGURE 6 – Erreur de prédiction moyenne par plage de vitesse avec les différents horizons de prédiction

tion avec différents horizons montre la capacité d’ADRIP pour la prévision à long-terme.

Par rapport aux méthodes existantes, notre solution présente de multiples avantages : (1) intègre de nombreuses propriétés des données de trafic, telles que la non linéarité et la dépendance spatio-temporelle sans augmenter la complexité du modèle grâce à des décisions décentralisées, (2) inclut des modèles sans paramètres, (3) permet la prédiction dynamique et en temps réel, (4) adapte la décision en fonction de l’évolution de l’environnement et (5) demande un temps de calcul et une capacité de calcul réduits grâce au mécanisme distribué.

Si les simulations ont permis de maîtriser les conditions d’évaluation des performances de plusieurs modèles, nous souhaitons dans l’immédiat appliquer P-ADRIP sur des ensembles réels de données à grande échelle afin de renforcer son efficacité et sa fiabilité. Ensuite, nous intégrerons un mécanisme de contrôle pour P-ADRIP qui permet automatiquement de détec-

ter, corriger l’erreur de prédiction et la mettre à jour en temps réel.

Références

- [1] Maxime Guériau, Frédéric Armetta, Salima Hassas, Romain Billot, and Nour-Eddin El Faouzi. A constructivist approach for a self-adaptive decision-making system : application to road traffic control. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 670–677. IEEE, 2016.
- [2] Nicolas Verstaevel, Jérémy Boes, Julien Nigon, Dorian d’Amico, and Marie-Pierre Gleizes. Lifelong machine learning with adaptive multi-agent systems. In *9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, volume 2, pages pp–275, 2017.
- [3] Giovanna Di Marzo Serugendo, Marie-Pierre Gleizes, and Anthony Karageorgos.

- Self-organising software : From natural to artificial adaptation, 2011.
- [4] Attila M Nagy and Vilmos Simon. Survey on traffic prediction in smart cities. *Pervasive and Mobile Computing*, 50 :148–163, 2018.
- [5] Brian L Smith, Billy M Williams, and R Keith Oswald. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C : Emerging Technologies*, 10(4) :303–321, 2002.
- [6] Daniel Billings and Jiann-Shiou Yang. Application of the arima models to urban roadway travel time prediction—a case study. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2529–2534. IEEE, 2006.
- [7] Sangsoo Lee and Daniel B Fambro. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record*, 1678(1) :179–188, 1999.
- [8] Billy M Williams and Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process : Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6) :664–672, 2003.
- [9] Billy M Williams, Priya K Durvasula, and Donald E Brown. Urban freeway traffic flow prediction : application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Record*, 1644(1) :132–141, 1998.
- [10] Yiannis Kamarianakis and Poulicos Prastacos. Forecasting traffic flow conditions in an urban network : Comparison of multivariate and univariate approaches. *Transportation Research Record*, 1857(1) :74–84, 2003.
- [11] Lun Zhang, Qiuchen Liu, Wenchen Yang, Nai Wei, and Decun Dong. An improved k-nearest neighbor model for short-term traffic flow prediction. *Procedia-Social and Behavioral Sciences*, 96 :653–662, 2013.
- [12] Pinlong Cai, Yunpeng Wang, Guangquan Lu, Peng Chen, Chuan Ding, and Jianping Sun. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C : Emerging Technologies*, 62 :21–34, 2016.
- [13] Shiliang Sun, Rongqing Huang, and Ya Gao. Network-scale traffic modeling and forecasting with graphical lasso and neural networks. *Journal of Transportation Engineering*, 138(11) :1358–1367, 2012.
- [14] Nicholas G Polson and Vadim O Sokolov. Deep learning for short-term traffic flow prediction. *Transportation Research Part C : Emerging Technologies*, 79 :1–17, 2017.
- [15] Rui Fu, Zuo Zhang, and Li Li. Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328. IEEE, 2016.
- [16] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network : a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2) :68–75, 2017.
- [17] Ha-Nhi Ngo, Elsy Kaddoum, Marie-Pierre Gleizes, Jonathan Bonnet, and Goursolle Anaïs. Life-long learning system of driving behaviors from vehicle data streams. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1132–1139. IEEE, 2021.
- [18] 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 *International conference on principles and practice of multi-agent systems*, pages 117–131. Springer, 2013.
- [19] Mordechai Haklay and Patrick Weber. Openstreetmap : User-generated street maps. *IEEE Pervasive computing*, 7(4) :12–18, 2008.
- [20] Patrick Taillandier. Traffic simulation with the gama platform. In *Eighth International Workshop on Agents in Traffic and Transportation*, pages 8–p, 2014.
- [21] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network : Data-driven traffic forecasting. *arXiv preprint arXiv :1707.01926*, 2017.

Estimation de données environnementales manquantes sans déploiement de capteurs supplémentaires : le système HybridIoT

Davide Andrea Guastella^a
davide.guastella@icar.cnr.it

Valérie Camps^b
valerie.camps@irit.fr

Marie-Pierre Gleizes^b
marie-pierre.gleizes@irit.fr

^aCNR Conseil National des Recherches Italien, ICAR,
Palerme, Italie

^bInstitut de Recherche en Informatique de Toulouse,
Université Toulouse III - Paul Sabatier,
31062 Toulouse Cedex 9, France

Résumé

La ville intelligente s'intéresse à l'amélioration de la qualité de vie de ses habitants. De nombreux capteurs ad hoc nécessitent alors d'être déployés pour connaître l'état de l'environnement dans lequel les activités humaines se déroulent. Si ces capteurs sont souvent bon marché, leurs coûts d'installation et de maintenance augmentent rapidement avec leur nombre. La problématique adressée dans ce papier consiste à estimer des informations environnementales où des capteurs physiques ne sont pas disponibles, pour limiter les coûts engendrés par l'installation de capteurs supplémentaires.

Cet article présente le système HybridIoT, qui permet d'estimer les valeurs environnementales manquantes dans des réseaux de capteurs à grande échelle. Notre contribution est triple : la définition d'une approche pour estimer des valeurs manquantes dans des environnements à grande échelle, l'utilisation des systèmes multi-agents pour adresser simultanément les propriétés d'ouverture et d'hétérogénéité, et enfin le déploiement du système dans le contexte du GIS neOCampus.

Mots-clés : *Ville intelligente, systèmes multi-agents coopératifs, estimation de données manquantes*

Abstract

The smart city aims at improving the quality of life of its citizens. Many sensors have to be deployed to monitor the state of the environment in which human activities take place. Despite these sensors being often cheap, their installation and maintenance costs increase rapidly with their number. In this paper, we address the problem of estimating environmental information where physical sensors are not available, to limit the costs related to the installation of additional sensors.

This paper presents the HybridIoT system for estimating missing environmental values in large-scale sensor networks. Our contribution is three-fold : the definition of an approach to estimate missing values in large-scale environments, the use of the multi-agent approach that addresses simultaneously the properties of openness and heterogeneity, and finally the deployment of the system in the context of the neOCampus operation.

Keywords: *Smart City, Cooperative Multi-Agent Systems, Missing Data Estimation*

1 Introduction

La ville intelligente répond, par des moyens technologiques, à des problèmes d'optimisation de ressources afin de permettre de meilleures interactions entre citoyens. Elle s'intéresse à plusieurs aspects de la société urbaine ; nous nous focalisons ici sur l'aspect technologique, à savoir l'utilisation de technologies récentes pour analyser les données de l'environnement urbain afin d'améliorer les services offerts aux citoyens, mais aussi leur qualité de vie.

Pour atteindre ces objectifs d'amélioration, il est nécessaire de pouvoir observer de façon ponctuelle et continue l'environnement par le biais de capteurs, sachant que l'idéal serait de parvenir à un compromis entre la précision des informations observées et les coûts d'installation et de maintenance pour les collectivités. Afin de réduire les coûts supportés par les collectivités, nous proposons de leur fournir des moyens technologiques permettant de combler le manque de capteurs, autrement dit de leur fournir des estimations de données environnementales dans des points non couverts par des capteurs sans coût supplémentaire. Cette proposition est l'objectif

du système HybridIoT dont nous allons présenter les principes. Concrètement, HybridIoT permet de définir une infrastructure IoT *hybride*, dans laquelle des capteurs physiques ainsi que virtuels fonctionnent conjointement et de manière transparente pour l'utilisateur du système, afin de fournir des informations en des points précis de l'environnement. L'utilisation des capteurs virtuels permet de fournir des informations à des endroits non couverts par des capteurs physiques sans installation de nouveau capteur.

Après avoir positionné ce travail par rapport aux principales techniques d'estimation de valeurs manquantes dans différents domaines d'application (section 2), la section 3 décrit une solution qui utilise conjointement plusieurs techniques classiques pour estimer des valeurs manquantes dans des environnements à grande échelle. La section 4 se focalise sur la présentation du système HybridIoT, qui permet d'estimer des valeurs environnementales manquantes dans des zones non couvertes par des capteurs tout en adressant simultanément les propriétés d'ouverture, d'hétérogénéité et de passage à l'échelle. La section 5 présente le déploiement en cours d'HybridIoT dans un contexte réel avant de conclure et de dresser quelques perspectives en section 6.

2 Contexte et positionnement

Le GIS neOCampus, supporté par l'UT3, joue un rôle important dans la conception de technologies qui pourraient être employées dans les villes intelligentes, en proposant un terrain d'expérimentations dans un contexte de campus intelligent [15]. Par leur taille, leur nombre d'utilisateurs et la diversité de leurs activités, les campus universitaires peuvent être considérés comme des petites villes. C'est la raison pour laquelle de plus en plus de chercheurs considèrent les campus universitaires comme des lieux privilégiés pour expérimenter des services et des techniques innovantes pour les villes intelligentes, en construisant ce que l'on appelle un "smart campus".

Le campus de l'UT3 s'étend sur environ 150 hectares et contient plus de 407 000 m² de bâtiments, avec 36 000 utilisateurs qui le fréquentent. Dans neOCampus, le campus est considéré comme une ville intelligente où plusieurs milliers de données sont fournies par des capteurs hétérogènes placés à l'intérieur et à l'extérieur des bâtiments (CO₂, consommation d'énergie et

de fluides, humidité, luminosité, présence humaine, ...).

Dans un tel contexte, le système HybridIoT tente de lever trois verrous : la prise en compte de la dynamique imprévisible de l'environnement, la décentralisation du calcul et l'hétérogénéité des entités présentes dans l'environnement. Pour cela, la solution proposée adresse simultanément trois propriétés : l'ouverture, l'hétérogénéité et le passage à l'échelle. L'ouverture permet au système de continuer à fonctionner sans aucune reconfiguration, même lors de l'apparition ou disparition de dispositifs. L'hétérogénéité permet l'intégration d'informations hétérogènes provenant de différentes sources de données. Le passage à l'échelle indique la possibilité de déployer le système dans des contextes urbains tels que les villes ou les régions. Nous avons passé en revue les principales techniques de l'état de l'art traitant de l'estimation de valeurs manquantes dans différents domaines d'application et les avons évaluées par rapport à leur capacité d'adresser les trois propriétés souhaitées (Tableau 1).

Nous avons utilisé quatre indicateurs pour décrire les points forts et les points faibles de chaque méthode décrite : (++) une propriété a été discutée et les auteurs présentent une solution pour la traiter, (+) une propriété a été discutée et traitée mais les auteurs n'ont pas fourni une description détaillée de la solution, (-) la propriété a été mentionnée mais pas traitée, (--) la propriété n'a été ni mentionnée ni traitée.

Comme le montre le tableau 1, la propriété d'ouverture n'est pas adressée par les techniques de l'état de l'art car l'ajout d'un grand nombre de capteurs en temps réel pourrait compromettre le fonctionnement du système, ou rendre des reconfigurations nécessaires. Nous pouvons également remarquer que la propriété d'hétérogénéité n'est que partiellement adressée car l'intégration d'informations dont le type n'est pas connu *a priori* nécessiterait une modification de la procédure d'estimation. Enfin, concernant le passage à l'échelle, les solutions proposées n'explicitent pas clairement si leur fonctionnement est correct ou pas lorsque le nombre de capteurs augmente.

3 Estimation par classification hiérarchique et convolution normalisée

L'objectif de cette étape est de concevoir une première solution pour estimer des valeurs environnementales manquantes dans des points où les

TABLE 1 – Comparaison des solutions de l'état de l'art pour l'estimation des informations manquantes.

Technologie	Auteurs	Domaine	Ouverture	Hétérogénéité	Passage à l'échelle
Régression	Hasenfratz et al. (2015) [1]	Environnement	--	++	+
	Seal et al. (2012) [2]		--	--	+
	Shan et al. (2016) [3]	Trafic urbain	--	++	+
	Tomaras et al. (2018) [4]	Maison intelligente et bâtiments	--	+	+
	Spencer et al. (2018) [5]		--	-	--
Réseaux Neuronaux (RN)	Kumar et al. (2013) [6]	Trafic urbain	--	++	+
	Yu et al. (2005) [7]	Environnement	--	-	--
	Ma et al. (2020) [8]		--	++	+
	Pisa et al. (2019) [9]		--	++	--
	Aliberti et al. (2018) [10]	Maison intelligente et bâtiments	--	--	--
Régression+ RN	Zhu et al. (2015) [11]	Environnement	--	++	+
	Mateo et al. (2013) [12]	Maison intelligente et bâtiments	--	-	--
Gradient Boost	Zhang et Haghani (2015) [13]	Trafic urbain	--	--	--
Combiné	Oprea et Bâra (2019) [14]	Maison intelligente et bâtiments	--	++	--

capteurs physiques ne sont pas disponibles. Elle s'appuie sur l'utilisation conjointe de plusieurs techniques standards telles que les diagrammes de Voronoï, la classification hiérarchique et la convolution normalisée (figure 1) [16].

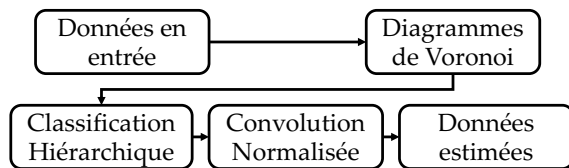


FIGURE 1 – Étapes principales de la méthode d'estimation basée sur des techniques standards.

Les diagrammes de Voronoï, habituellement utilisés pour créer efficacement des cartes de distances dans des images binaires [17] sont utilisés ici pour délimiter l'estimation de données à l'intérieur de régions spécifiques, autrement dit pour déterminer des régions/voisinages dans lequel(le)s les capteurs perçoivent des informations ayant une dynamique similaire. La classification hiérarchique est utilisée pour regrouper les capteurs qui perçoivent des informations similaires. Enfin, la convolution normalisée (habituellement utilisée pour reconstruire des données incomplètes ou incertaines [18]) est ici utilisée pour estimer les informations manquantes.

Plus concrètement, les régions de deux ou plusieurs capteurs sont fusionnées si leurs zones de Voronoï sont adjacentes et sont regroupées par le processus de classification hiérarchique. Enfin, pour un point donné, la convolution normalisée permet d'estimer les informations environnementales manquantes en utilisant les capteurs situés dans la région de Voronoï correspondante.

Cette première étape a permis d'obtenir une base de résultats qui ont été comparés à ceux obtenus par la méthode multi-agent présentée dans la section suivante.

4 Le système HybridIoT

L'objectif de cette étape est de concevoir une solution multi-agent pour estimer des valeurs environnementales dans des zones non pourvues de capteurs physiques, qui soit capable d'adresser les propriétés d'ouverture et d'hétérogénéité.

Le système HybridIoT se compose de deux types d'agents distincts : l'*Agent Capteur Réel (Real Sensor Agent, RSA)* qui représente toute instrumentation physique capable de fournir des informations précises sur l'environnement (comme la température, l'humidité, ...), et l'*Agent Contexte Ambient (Ambient Context Agent, ACA)* qui est chargé d'estimer, en un point spécifique de l'environnement, la valeur qu'aurait perçue un capteur réel s'il était situé à ce même point. Un ACA peut être associé à un capteur physique ; son rôle est alors de fournir un mécanisme de résilience pour produire des estimations lorsque le capteur physique situé au même endroit est momentanément dans l'impossibilité de percevoir l'environnement (panne...).

HybridIoT propose d'estimer des informations environnementales selon trois mécanismes distincts (Figure 2) :

- **Estimation endogène à partir d'un historique** 🕒 : l'estimation se fonde sur les informations précédemment acquises par les capteurs ;

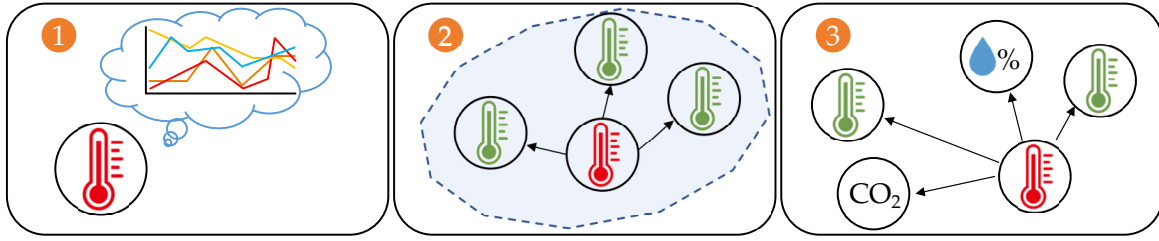


FIGURE 2 – Techniques d'estimation d'HybridIoT : estimation endogène à partir d'un historique (❶), estimation endogène à l'aide de capteurs voisins homogènes (❷) et estimation exogène (❸).

- **Estimation endogène à l'aide de capteurs voisins homogènes** (❷) : l'estimation utilise les informations de même type (et avec la même unité) acquises par les capteurs situés à proximité de l'endroit où on souhaite connaître une valeur environnementale ;
- **Estimation exogène** (❸) : l'estimation est réalisée en intégrant des informations hétérogènes (de types différents, avec des échelles différentes).

Nous allons à présent décrire chacun de ces mécanismes d'estimation. Leur description est accompagnée de résultats permettant de mettre en avant la valeur ajoutée d'HybridIoT. Ces résultats ont été obtenus en utilisant un jeu de données disponible sur le Web et comprenant des informations hétérogènes (notamment température, vitesse du vent, rayonnement solaire, humidité) acquises par 80 stations météorologiques durant 196 jours, à partir de septembre 2017 dans la région Émilie-Romagne (Italie) [18].

4.1 Estimation endogène à partir d'un historique

L'estimation endogène à partir d'un historique (Figure 2 (❶)) s'effectue en deux étapes : l'évaluation coopérative des poids à associer aux informations de l'historique, puis l'estimation des données manquantes en utilisant les poids précédemment évalués. Avant de détailler chacune de ces étapes nous allons poser quelques définitions.

Définition 1 (Fenêtre contextuelle) Une ACW (Ambient Context Window) C_t regroupe des informations environnementales homogènes perçues durant un intervalle de temps discret $T = [t - \delta, t]$, $t - \delta < t$. Une ACW contient $|C_t| = |T|$ informations homogènes, une pour chaque instant temporel. Chaque ACW C_t est associé à un indice t qui correspond à l'instant t durant lequel l'information a été perçue.

Définition 2 (Distance entre ACWs) La distance entre deux ACWs, calculée afin de les comparer, est définie comme la différence en valeur absolue des valeurs des deux ACWs, divisée par le nombre de valeurs γ des deux ACWs. Plus la différence est faible, plus les deux ACWs sont similaires. La distance entre deux ACWs C_t et C_k est définie par la formule suivante :

$$d(C_t, C_k) = \frac{\sum_{\ell \in [1, \gamma]} |E_\ell^t - E_\ell^k|}{\gamma} \quad (1)$$

où $\gamma = |C_t| = |C_k|$, ℓ est un indice dans la plage $[1, \gamma]$, E_ℓ^t et E_ℓ^k sont respectivement les valeurs en position ℓ dans C_t et C_k .

Supposons qu'un ACA_i doit estimer une information manquante à l'instant t parce que son capteur associé n'est plus disponible. L' ACA_i cherche parmi ses ACWs, celles qui sont les plus similaires à celle contenant l'information à estimer. Plus concrètement, l' ACA_i utilise les distances entre les ACWs pour calculer un poids w_t qui est ajouté à la dernière information perçue à l'instant $t - 1$. Le poids w_t est calculé comme la moyenne des différences des deux dernières valeurs de chaque ACW $C_k \in \xi$ en utilisant la distance $d(C_k, C_t)$, $\forall C_k \in \xi$, où ξ est l'ensemble des ACWs les plus similaires à ACA_i . Le poids w_t est calculé grâce à la formule 2 :

$$w_t = \frac{\sum_{C_k \in \xi} (E_\ell^k - E_{\ell-1}^k) \cdot d(C_t, C_k)}{\sum_{C_k \in \xi} d(C_t, C_k)} \quad (2)$$

où C_t est l'ACW contenant l'information à estimer à l'instant t , $C_k \in \xi$ est la k -ième ACW la plus similaire à C_t et E_ℓ^k et $E_{\ell-1}^k$ sont respectivement la ℓ^e et la $(\ell - 1)^e$ valeur de l'ACW $C_k \in \xi$, à savoir les deux dernières entrées de contexte de C_k .

L'estimation de la valeur manquante est alors calculée à l'aide de la formule :

$$E_t^j = E_{t-1}^j + w_t \quad (3)$$

où E_t^j et E_{t-1}^j sont respectivement les estimations de l'information manquante aux instants t et $t - 1$, et $E_{t-1}^j \in C_j$ est la dernière information perçue par ACA_i .

Enfin, l' ACA_i évalue une nouvelle ACW pour l'information estimée à l'instant t [19].

Un mécanisme de coopération entre ACAs a été défini pour combler un éventuel manque de données. Ce processus de coopération est initié lorsqu'un ACA_i n'est pas capable de poursuivre son objectif (i.e. de fournir des informations environnementales parce que le capteur physique auquel il est associé est en panne ou parce qu'il possède trop peu d'informations dans son historique d'ACWs). Dans une telle situation, l' ACA_i choisit les agents avec qui il coopère selon deux critères : soit il choisit de coopérer avec les agents les plus proches physiquement, soit avec les agents qui suivent une dynamique d'information similaire à la sienne. Le choix du critère à utiliser est confié à l'utilisateur du système.

Une fois les agents avec qui coopérer choisis par l' ACA_i , ce dernier leur indique quelles ACWs de leur historique ils vont utiliser pour calculer les poids qui seront utilisés pour réaliser son estimation (celles qui, chez eux, dans le passé, étaient similaires au contexte courant donc dont les informations perçues suivaient une dynamique similaire). Une fois les ACWs connues, chaque agent fournit à l' ACA_i des poids calculés à l'aide de la formule 2. L' ACA_i calcule autant d'estimations que de poids reçus par les agents avec qui il coopère. Toutes les valeurs sont enfin pondérées afin de fournir une valeur d'estimation [18].

TABLE 2 – Erreur moyenne absolue obtenue par estimation endogène à partir d'un historique.

	Erreur moyenne absolue	
	Irradiation solaire (W/m ²)	Vitesse du vent (m/s)
Agents les plus confiants	0.15851	0.15888
Agents les plus proches	0.16265	0.16141
	Température (°C)	Humidité (%)
Agents les plus confiants	0.060	0.16320
Agents les plus proches	0.081	0.16240

Le tableau 2 montre les résultats obtenus en utilisant l'estimation endogène sur l'ensemble du jeu de données météorologiques. Nous avons évalué notre proposition sur quatre ensembles d'informations de types différents pour mettre en évidence la généralité d'HybridIoT : notre

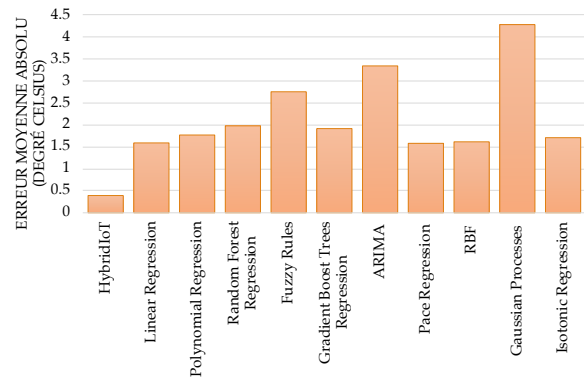


FIGURE 3 – Comparaison des résultats obtenus par HybridIoT et par les techniques disponibles dans KNIME.

proposition permet d'estimer avec précision différents types d'informations sans configuration particulière. L'erreur obtenue, calculée comme la différence entre les estimations réalisées et les valeurs réelles, est significativement faible, ce qui prouve la pertinence de la solution proposée.

La figure 3 compare les résultats obtenus par HybridIoT et ceux obtenus par des techniques d'estimation disponibles dans le logiciel KNIME [20] en utilisant les paramètres par défaut pour ces techniques et en utilisant le même jeu de données pour tous les tests. On peut constater qu'HybridIoT fournit des estimations avec une erreur moyenne plus faible, par rapport aux valeurs réelles, tout en adressant simultanément les propriétés d'ouverture, d'hétérogénéité et de passage à l'échelle. L'ouverture a été testée en simulant l'indisponibilité de plusieurs capteurs ; nous avons alors constaté que le système continue à fonctionner sans qu'aucune reconfiguration ne soit nécessaire. L'hétérogénéité a été testée en utilisant plusieurs jeux de données de types différents, sans préciser le type de données manipulées. Le passage à l'échelle a lui été évalué en utilisant de nombreux capteurs déployés à l'échelle régionale

4.2 Estimation endogène à l'aide de capteurs voisins homogènes

Dans ce type d'estimation, l' ACA_i estime les valeurs manquantes en coopérant avec des agents (soit des RSA, soit des ACA) percevant le même type d'information que lui et situés à l'intérieur de sa zone de confiance.

Définition 3 (Zone de confiance) Une zone de confiance est un polygone à n côtés associé à un

ACA et situé autour de lui. Un ACA utilise sa zone de confiance pour regrouper les capteurs (ACAs ou RSAs) dont les informations perçues suivent une dynamique similaire (c'est-à-dire ayant des ACW dont la distance est faible). Un ACA peut modifier instantanément la forme de sa zone de confiance pour ne garder dans la région que les capteurs qui fournissent des informations similaires aux siennes.

Lorsque celui-ci est incapable de percevoir une information à partir de l'observation directe de son environnement, il coopère avec les agents situés dans sa zone de confiance. Ces derniers lui envoient leurs perceptions de l'environnement. Pour chaque paire d'agents dans la zone de confiance, l'ACA_i calcule un unique **champ de données** représentant une estimation pour la valeur manquante.

Définition 4 (Champ de données) *Un champ de données Γ entre deux agents (soit ACAs soit des RSAs) est un champ vectoriel dans l'espace euclidien. Chaque point est associé à un vecteur qui est orienté vers l'agent qui fournit la valeur la plus élevée. L'estimation se base alors sur la valeur du gradient entre les données perçues par les capteurs.*

Une fois que les champs de données ont été calculés, l'ACA_i calcule l'estimation de la valeur manquante à l'instant t .

$$E_t^j = \frac{\sum_{p \in \text{keyset}(\Theta)} \Gamma(p) \cdot \Theta(p)}{\sum_{p \in \text{keyset}(\Theta)} \Theta(p)}. \quad (4)$$

où E_t^j est l'estimation de l'information manquante à l'instant t , $\Gamma(p)$ est le champ de données calculé par la paire d'agents p , Θ est un dictionnaire contenant des paires d'agents comme clé et leur colinéarité par rapport à ACA_i comme valeur. Nous définissons la colinéarité comme la distance entre le point où ACA_i est situé et la ligne entre deux agents situés dans sa zone de confiance. $\text{keyset}(\Theta)$ contient les paires d'agents à l'intérieur de la zone de confiance, $\Theta(p)$ est la colinéarité entre les agents de la paire p et ACA_i. L'équation (4) renvoie une estimation de la valeur manquante en pondérant les champs de données des paires d'agents évaluées avec les valeurs de colinéarité des agents par rapport à ACA_i. L'idée est que les agents ayant une valeur de colinéarité plus élevée (c'est-à-dire étant plus alignés) fournissent des informations plus précises car le gradient correspondant fournit des

valeurs précises si l'ACA_i est situé dans la même direction que le gradient et à proximité des deux capteurs.

Enfin, la forme de la zone de confiance de l'ACA_i est modifiée pour ne garder à l'intérieur de celle-ci que les agents qui ont fourni des valeurs cohérentes (notion définie par la suite). Pour modifier la forme de la zone de confiance, l'ACA_i trie la liste des valeurs des champs de données reçues selon l'ordre croissant puis calcule le champ de données médian $\tilde{\gamma}$. L'ACA_i calcule ensuite la valeur δ comme étant la distance entre les champs de données de la première paire d'agents ($\Gamma(p_1)$ pour p_1) et de la troisième ($\Gamma(p_3)$ pour p_3). Nous avons choisi la première et la troisième paires d'agents de manière empirique, après plusieurs expérimentations effectuées en environnements fermés (intérieur), mais aussi en environnements ouverts (extérieurs) tels que des bâtiments universitaires [21]. Nous utilisons deux valeurs de seuil th^\pm pour partitionner les champs de données.

$$th^\pm \leftarrow \Gamma(p_1) \pm \delta \times \omega \quad (5)$$

où la constante ω a une valeur fixée expérimentalement à 2.5.

L'ACA_i élargit ensuite sa zone de confiance en direction des agents fournissant des champs de données cohérents (qui fournissent des valeurs de champs de données dans l'intervalle $[th^-, th^+]$), et réduit la zone de confiance pour exclure les autres agents. Il s'agit d'un processus local entre les ACAs situés dans une zone de confiance spécifique.

Nous comparons les résultats avec ceux obtenus en utilisant le pipeline de techniques classiques (cf. section 3), sur le même jeu de données. Les résultats sont présentés dans la Figure 4. Nous avons regroupé les résultats des capteurs par groupes de 20 pour faciliter la visualisation des résultats.

Nous pouvons voir sur la figure 4 qu'HybridIoT fournit une plage d'erreur sensiblement plus large que le pipeline de techniques standards. Cette différence est due à la dynamique de la zone de confiance des ACA qui est inexistante dans le pipeline de techniques standards. Cette dynamique au niveau des zones de confiance est un avantage pour HybridIoT car ces zones ont une taille qui s'ajuste au fur et à mesure des estimations pour progressivement exclure les informations biaisées et par conséquent diminuer les erreurs d'estimations. Au delà de cet avantage, HybridIoT permet d'adresser des problèmes plus complexes (ouverture, hétérogénéité et passage

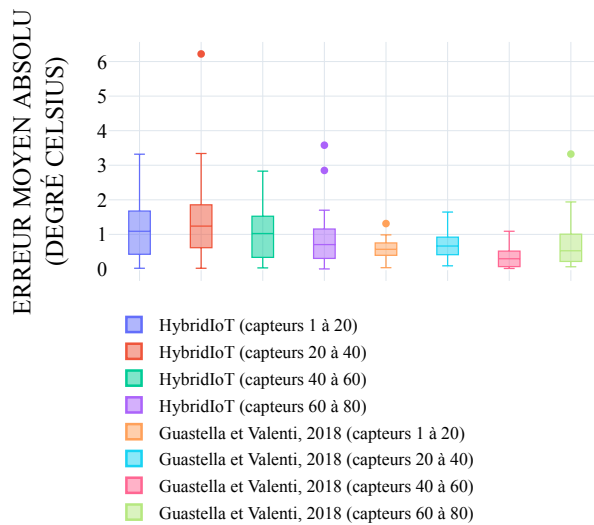


FIGURE 4 – Résultats obtenus par HybridIoT en utilisant la zone de confiance et les résultats obtenus par le pipeline des méthodes standards (Voronoi+classification hiérarchique).

à l'échelle) que le pipeline de techniques standards.

4.3 Estimation exogène

Dans ce type d'estimation, l'ACA_i ne perçoit pas d'agents dans son voisinage capables de capter des informations de même type (et même échelle) que lui (Figure 2 [3]). L'idée proposée dans ce cas, est de considérer que les valeurs perçues par différents capteurs peuvent suivre une dynamique semblable dans des conditions environnementales similaires. Sur cette base, les ACAs sont capables d'estimer les valeurs manquantes en exploitant un large éventail de données hétérogènes.

Plus concrètement, l'ACA_i utilise l'estimation exogène à l'instant t si (i) il n'y a pas d'agents homogènes à l'intérieur de sa zone de confiance et (ii) certains agents percevant des informations hétérogènes ou homogènes sont disponibles au-delà de la zone de confiance de l'ACA_i. Le mécanisme d'estimation exogène comporte quatre étapes :

1. **Estimation par les données historiques** : les ACAs estiment les valeurs manquantes de manière endogène, puis coopèrent avec d'autres agents (soit des ACAs, soit des RSAs, le cas échéant) percevant des informations hétérogènes et situées au-delà de la zone de confiance

des ACAs. Il s'agit du mécanisme présenté en section 3.

2. **Détermination de l'ensemble des agents avec qui coopérer** : les ACA déterminent l'ensemble des agents avec qui coopérer pour estimer les informations manquantes, en utilisant un des deux critères mentionnés auparavant : soit un ACA choisit de coopérer avec les agents les plus proches, soit avec les agents qui suivent une dynamique d'information similaire à la sienne.
3. **Évaluation coopérative de l'ensemble des poids** : les agents qui coopèrent avec ACA_i calculent chacun un poids (comme expliqué pour l'estimation endogène, à partir d'un historique, en utilisant leurs ACWs dont les indices sont indiqués par ACA_i).
4. **Calcul de l'estimation** : l'ACA_i construit un histogramme H à partir des estimations calculées en utilisant les poids obtenus à l'étape précédente. Soit f une fonction qui renvoie la moyenne des valeurs contenues dans une case H_k de l'histogramme H telle que la différence entre H_k et l'estimation endogène E_{end} soit minimisée :

$$f(H_k) = |\overline{H}_k - E_{end}| \quad (6)$$

où \overline{H}_k est la moyenne des valeurs de la k -ième case de l'histogramme H . L'ACA_i calcule la valeur estimée E_{exo} à l'aide de l'équation :

$$E_{exo} = \frac{\operatorname{argmin}_k f(H_k) + E_{end}}{2}. \quad (7)$$

Le tableau 3 résume les résultats d'estimation de valeurs de température, calculées à partir d'informations hétérogènes issues du jeu de données météorologiques. Ces résultats montrent que les ACAs fournissent des estimations précises grâce à des informations hétérogènes sans utiliser de techniques ou configurations particulières. L'erreur résultant de l'estimation exogène est faible même si certains ACAs perçoivent des valeurs de températures qui ont des types ou échelles différentes.

5 Déploiement dans un contexte réel

L'objectif de cette étape est de déployer HybridIoT sur le campus de l'Université Paul Sabatier (UT3), en utilisant des capteurs IoT réels

TABLE 3 – Erreur absolue moyenne et écart-type (en degré Celsius) obtenus par l’estimation exogène.

	Erreur absolue moyenne (°C)	
	Agents les plus confiants	Agents les plus proches
Temp. à partir de irradiation solaire	0.02716	0.03796
Temp. depuis humidité	0.02654	0.03679
Temp. à partir de vitesse du vent	0.02531	0.03914
Temp. à partir de toutes les infos	0.07	0.05935
	Écart-type (°C)	
	Agents les plus confiants	Agents les plus proches
Temp. à partir de irradiation solaire	1.639136	1.645494
Temp. depuis humidité	1.646173	1.647901
Temp. à partir de vitesse du vent	1.64716	1.651358
Temp. à partir de toutes les infos	1.65435	1.62805

possédés par les utilisateurs du campus (RSA) ainsi que des capteurs virtuels (ACA). Le développement front-end d’HybridIoT consiste à développer une application mobile multiplateforme, fonctionnant sous Android et IOS, qui joue le rôle d’intermédiaire entre l’estimation et les capteurs embarqués (si disponibles). Le développement back-end d’HybridIoT concerne l’infrastructure nécessaire (matérielle et logicielle) pour supporter le calcul de l’estimation de manière décentralisée ainsi que le stockage des informations acquises par les capteurs (physiques ou virtuels). L’application mobile propose trois fonctionnalités : l’acquisition d’informations à partir des capteurs embarqués, l’affichage de statistiques sur les informations perçues et l’installation de nouveaux capteurs ACA virtuels. Les deux premières fonctionnalités ne sont accessibles qu’aux utilisateurs, tandis que l’installation des capteurs ACA virtuels est réservée au personnel technique du campus.

La figure 5 montre deux captures d’écran de l’application développée sous Android. Actuellement, l’application permet la visualisation de l’emplacement des capteurs dans le campus ainsi que les valeurs environnementales perçues par les capteurs physiques sous forme de graphique.

Du point de vue du back-end, nous devons avoir une infrastructure pour supporter HybridIoT mais aussi fournir des informations instantanées à tout moment et partout. Cette infrastructure est supportée par la plateforme neOCampus

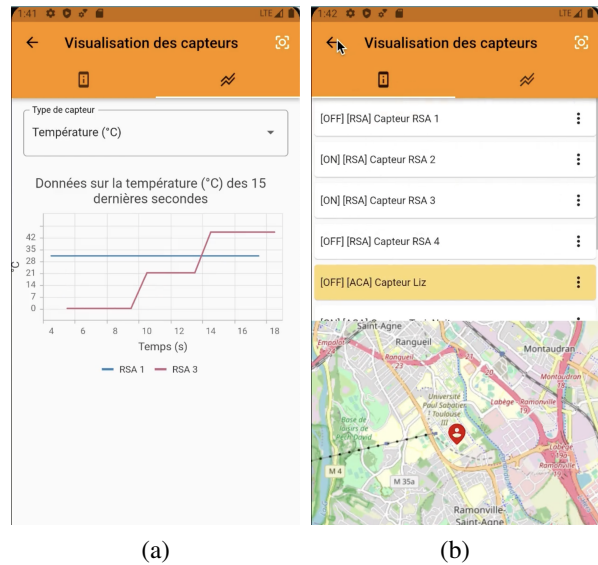


FIGURE 5 – HybridIoT pour Android : statistiques des valeurs de température sur les 15 dernières secondes fig. 5a), emplacement des capteurs et leur identifiant (fig. 5b).

de l’UT3. Cette plateforme comporte trois équipements principaux : (i) 16 cœurs, 64 Go de RAM et 2.7 To de stockage haut débit, (ii) un serveur de stockage de 45 To partagé avec la plateforme CloudMIP, et (iii) un réseau IoT dédié, filaire et sans fil.

Le serveur comprend un broker MQTT qui permet à tous les capteurs/actionneurs/applications ambiants d’être interconnectés. Les capteurs/actionneurs sont reliés au réseau IoT par des dispositifs tels que Raspberry Pi et ESP8266. Ces appareils sont authentifiés par un mécanisme hébergé sur le serveur, qui inclue l’application sensOCampus (<http://sensocampus.univ-tlse3.fr>) ainsi qu’une base de données spécifique. Un réseau LoRaWAN sans fil est également déployé à l’échelle du campus afin de permettre aux dispositifs LoRa d’envoyer les données des capteurs dans un rayon de plus de 2 km. Les données sont ensuite acheminées vers le serveur principal du campus [15].

Nous développons actuellement une version d’HybridIoT qui sera hébergée sur le serveur de neOCampus. L’objectif de cette version est triple :

- limiter la portée opérationnelle d’HybridIoT au campus universitaire. Afin d’éviter des résultats biaisés causés par des appareils qui ne sont pas situés sur le campus, seuls les capteurs mobiles se trouvant à l’intérieur du campus doivent en-

- voyer des informations au serveur.
- permettre l'utilisation de données acquises par des capteurs pouvant supporter différentes fréquences d'acquisition. Le protocole actuel doit être modifié car les données sont échangées entre les capteurs et le serveur via le protocole MQTT.
- proposer un calcul décentralisé *embarqué*, afin de réduire la charge de calcul du serveur et permettre à un utilisateur de réaliser l'estimation directement depuis son smartphone plutôt que sur le serveur.

Le fonctionnement d'HybridIoT peut alors se résumer ainsi : les agents (RSA et ACA) acquièrent des données (à partir des capteurs physiques disponibles) qui sont envoyées au serveur neOCampus via MQTT. Le serveur stocke ces informations, estime les valeurs manquantes pour les ACA enregistrés, puis renvoie l'estimation à l'utilisateur. Les administrateurs du système doivent alors déterminer la qualité des estimations par rapport aux valeurs réelles, afin de décider si l'installation d'un nouveau capteur physique est nécessaire ou pas ; si tel n'est pas le cas, les ACA sont utilisés.

Nous devons également préserver la confidentialité des données manipulées et s'assurer de la pertinence de ces données pour éviter l'insertion de bruit dans le processus d'estimation.

6 Conclusion et travaux futurs

Cet article présente le système HybridIoT qui permet d'estimer des valeurs environnementales dans des endroits non couverts par des capteurs. Cette présentation se compose de 3 parties complémentaires : (i) une méthode d'estimation basée sur l'utilisation conjointe de plusieurs techniques standards, à des fins de comparaison, (ii) une méthode d'estimation à base d'agents pour adresser simultanément les propriétés d'ouverture, d'hétérogénéité et de passage à l'échelle et (iii) le déploiement d'HybridIoT sur le campus de l'UT3. Cette dernière étape, en cours de réalisation, s'appuie sur l'infrastructure neOCampus, fournissant un serveur puissant pour stocker les données et effectuer les calculs. Elle est réalisée en collaboration avec l'entreprise MODIS, impliquée dans le développement du prototype dans le cadre d'un mécénat avec l'UT3.

HybridIoT a été évalué sur un jeu de données météorologiques et comparé avec des techniques de l'état de l'art. Les résultats obtenus montrent la pertinence de notre proposition pour estimer des valeurs environnementales manquantes dans

des environnements à grande échelle utilisant un nombre limité de capteurs. Comparée aux techniques de l'état de l'art, HybridIoT permet d'adresser simultanément les propriétés d'ouverture, d'hétérogénéité et de passage à l'échelle. Nos objectifs à court terme sont (i) d'évaluer combien de capteurs physiques peuvent être remplacés par des ACA sans compromettre la qualité des estimations, et (ii) de déterminer si l'installation de capteurs physiques est nécessaire dans les zones de l'environnement où les estimations ne sont pas cohérentes avec les valeurs réelles. Nous allons également poursuivre le déploiement d'HybridIoT sur le campus de l'UT3 afin d'atteindre un niveau de maturité technologique (TRL) de 6-7.

7 Remerciements

Ce travail de recherche a été réalisé dans le cadre du GIS neOCampus (www.neocampus.org) de l'Université Toulouse III - Paul Sabatier. Il vient de bénéficier (2022) d'un financement de la part de Toulouse Tech Transfer pour poursuivre le développement d'HybridIoT et son déploiement sur le campus de l'UT3 afin d'atteindre un niveau de maturité technologique (TRL) de 6-7.

Références

- [1] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, T. Arn, J. Beutel, and L. Thiele, "Deriving high-resolution urban air pollution maps using mobile sensor nodes," *Pervasive and Mobile Computing*, vol. 16, pp. 268 – 285, 2015.
- [2] V. Seal, A. Raha, S. Maity, S. Kumar Mitra, A. Mukherjee, and M. K. Naskar, "A Simple Flood Forecasting Scheme Using Wireless Sensor Networks," *International Journal of Ad hoc, Sensor & Ubiquitous Computing (AIRCC)*, vol. 3, Feb. 2012.
- [3] Z. Shan, Y. Xia, P. Hou, and J. He, "Fusing Incomplete Multisensor Heterogeneous Data to Estimate Urban Traffic," *IEEE MultiMedia*, vol. 23, no. 3, pp. 56–63, 2016.
- [4] D. Tomaras, V. Kalogeraki, N. Zygouras, N. Panagiotou, and D. Gunopulos, "Evaluating the Health State of Urban Areas Using Multi-source Heterogeneous Data," in *2018 IEEE 19th Int. Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 14–22, 2018.

1. www.toulouse-tech-transfer.com

- [5] B. Spencer, O. Alfandi, and F. Al-Obeidat, "A Refinement of Lasso Regression Applied to Temperature Forecasting," *Procedia Computer Science (Elsevier)*, vol. 130, pp. 728–735, 2018.
- [6] K. Kumar, M. Parida, and V. Katiyar, "Short Term Traffic Flow Prediction for a Non Urban Highway Using Artificial Neural Network," *Procedia - Social and Behavioral Sciences*, vol. 104, pp. 755 – 764, 2013.
- [7] L. Yu, N. Wang, and X. Meng, "Real-time forest fire detection with wireless sensor networks," in *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005.*, vol. 2, pp. 1214–1217, 2005.
- [8] Y. Ma, S. Liu, G. Xue, and D. Gong, "Soft Sensor with Deep Learning for Functional Region Detection in Urban Environments," *Sensors*, vol. 20, p. 3348, June 2020.
- [9] I. Pisa, I. Santín, J. Vicario, A. Morell, and R. Vilanova, "ANN-Based Soft Sensor to Predict Effluent Violations in Wastewater Treatment Plants," *Sensors*, vol. 19, p. 1280, Mar. 2019.
- [10] A. Aliberti, F. M. Ugliotti, L. Bottaccioli, G. Cirrincione, A. Osello, E. Macii, E. Patti, and A. Acquaviva, "Indoor Air-Temperature Forecast for Energy-Efficient Management in Smart Buildings," in *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, pp. 1–6, 2018.
- [11] J. Y. Zhu, C. Sun, and V. O. Li, "Granger-Causality-based air quality estimation with spatio-temporal (S-T) heterogeneous big data," in *Computer Communications Workshops (INFOCOM WKSHPS)*, (Hong Kong), pp. 612–617, IEEE, 2015.
- [12] F. Mateo, J. J. Carrasco, A. Sellami, M. Millán-Giraldo, M. Domínguez, and E. Soria-Olivas, "Machine learning methods to forecast temperature in buildings," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1061 – 1068, 2013.
- [13] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transportation Research Part C : Emerging Technologies*, vol. 58, pp. 308 – 324, 2015.
- [14] S.-V. Oprea and A. Bâra, "Machine Learning Algorithms for Short-Term Load Forecast in Residential Buildings Using Smart Meters, Sensors and Big Data Solutions," *IEEE Access*, vol. 7, pp. 177874–177889, 2019.
- [15] M.-P. Gleizes, J. Boes, B. Lartigue, and F. Thiébolt, "neOCampus : A Demonstrator of Connected, Innovative, Intelligent and Sustainable Campus," in *Intelligent Interactive Multimedia Systems and Services 2017*, vol. 76, pp. 482–491, Springer, 2018.
- [16] D. A. Guastella and C. Valenti, "Estimating Missing Information by Cluster Analysis and Normalized Convolution," in *4th Int. Forum on Research and Technology for Society and Industry (RTSI)*, pp. 1–6, 2018. doi : 10.1109/RTSI.2018.8548454.
- [17] D. Lo Castro, D. Tegolo, and C. Valenti, "A visual framework to create photorealistic retinal vessels for diagnosis purposes," *Journal of Biomedical Informatics*, vol. 108, p. 103490, 2020.
- [18] D. A. Guastella, V. Camps, and M.-P. Gleizes, "A Cooperative Multi-Agent System for Crowd Sensing Based Estimation in Smart Cities," *IEEE Access*, vol. 8, pp. 183051–183070, 2020. doi : 10.1109/ACCESS.2020.3028967.
- [19] D. A. Guastella, V. Camps, and M.-P. Gleizes, "Estimating missing environmental information by contextual data cooperation," in *PRIMA 2019 : Principles and Practice of Multi-Agent Systems*, pp. 523–531, Springer, 2019.
- [20] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel, "KNIME - the konstanz information miner : Version 2.0 and beyond," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 26–31, 2009. doi : 10.1145/1656274.1656280.
- [21] D. A. Guastella, N. Verstaevel, C. Valenti, B. Arshad, and J. Barthélemy, "Evaluating Correlations in IoT Sensors for Smart Buildings," in *13th Int. Conference on Agents and Artificial Intelligence (ICAART)*, pp. 224–231, SCITEPRESS, 2021.

Session 5 : Simulation, modélisation

Dairy Health Manager : un simulateur multi-agents flexible pour l'étude des maladies des animaux d'élevage

Philippe Gontier^a
philippe.gontier@oniris-nantes.fr

Nathalie Bareille^a
nathalie.bareille@oniris-nantes.fr

Sébastien Picault^a
sebastien.picault@inrae.fr

^aINRAE, Oniris, BIOEPAR, 44300, Nantes, France

Résumé

La simulation est une approche pertinente pour comparer ex ante divers scénarios d'apparition, de développement et de traitement de maladies et pour identifier des leviers de maîtrise efficaces. Dans le cas de l'élevage, des simulateurs ad hoc ont été régulièrement élaborés pour quantifier l'effet des maladies sur les performances animales et leurs conséquences économiques. La plateforme DHM (Dairy Health Manager) propose un cadre multi-agents pour renforcer la performance, la généricité et l'évolutivité de tels modèles et faciliter les échanges avec des scientifiques non modélisateurs en sciences animales. Dans cet article, nous illustrons la modularité de DHM à travers des Agents Vaches Laitières (AVL) et les Agent Délégués Quartier (ADQ) évoluant au sein de leur environnement que représentent les différents lots du troupeau. Le développement d'une première application aux cétooses offre de nombreuses perspectives.

Mots-clés : Simulation de troupeaux, Maladies des bovins laitiers, Modélisation multi-niveaux

Abstract

The simulation is a pertinent approach to compare the dynamics and treatment scenarios for various diseases and to identify the most effective control measures. In animal production, simulators have been regularly developed to assess the effect of animal diseases on animal performances (e.g. milk production) and their economic consequences. The DHM platform (Dairy Health Manager) offers a multi-agent framework to enhance the performance, genericity and upgradability of such models and facilitate exchanges with non-modelling scientists in animal health. We

illustrate the modularity of DHM through Dairy Cow Agents and Quarter Delegate Agents evolving within their environment that represents the different batches of the herd. The development of a first application to ketosis opens multiple perspectives.

Keywords: Herd simulation, Dairy cattle diseases, Multi-level modelling

1 Introduction

La rentabilité économique des mesures collectives de maîtrise de la santé est un axe majeur des recherches menées dans le domaine de l'agriculture. Pour être en mesure d'analyser les effets économiques de scénarios de maîtrise de la santé animale au niveau du troupeau, toutes choses égales par ailleurs, le recours à un simulateur d'élevage bovin laitier prenant en compte les effets individuels et leurs interrelations est très pertinent. À cette fin, ce simulateur doit représenter les fonctions biologiques impactées par les maladies les plus fréquentes et les fonctions ayant un impact fort sur la rentabilité de l'activité.

Les résultats obtenus en simulation doivent être cohérents avec ceux qui seraient observés sur le terrain. Cela passe par une modélisation précise des décisions prises par l'éleveur et de leur impact biologique sur chacun des animaux, compte tenu de leurs caractéristiques propres. Du fait de la diversité des caractères individuels à modéliser pour chaque fonction biologique, le système à représenter doit être centré sur l'individu.

Si des simulateurs de troupeaux laitiers inté-

grant la gestion individuelle et une dimension économique existent en santé animale, leur développement a généralement été conduit pour répondre à des problématiques ciblées et sans anticiper l'évolution des besoins, la transposition à des contextes différents ou l'évolution des connaissances scientifiques. Pourtant, la réponse à apporter aux professionnels de l'élevage pour mettre au point leur stratégie de maîtrise doit permettre d'intégrer de manière réactive le progrès des connaissances en sciences animales. Elle doit prévoir d'emblée l'évolutivité propre à intégrer de nouvelles pratiques d'élevage (réelles ou expérimentales) ou liées à la modélisation de nouvelles maladies. Cette réactivité implique un dialogue permanent et interdisciplinaire entre biologistes et informaticiens quant aux besoins fonctionnels induits par de nouvelles questions scientifiques.

L'objectif de cet article est de présenter l'approche méthodologique mise en œuvre à partir d'une vision macroscopique des champs disciplinaires et de décrire la modélisation du Système Multi-Agents (SMA) mis en pratique pour la production du logiciel de simulation « *Dairy Health Manager* » DHM (Gontier and Bareille 2021) avec comme première application la prise en compte de la cétose, maladie métabolique des bovins laitiers.

2 Besoins et travaux proches

2.1 Besoins de simulation

Le développement d'un simulateur de troupeau laitier évolutif a été entamé à BIOEPAR en 2015. Son objectif était de permettre à un expérimentateur d'évaluer des stratégies de maîtrise des maladies de production en fournissant des bilans économiques. La solution à mettre en œuvre devait donc prendre en compte la dynamique d'individus aux caractéristiques biologiques multiples et de leurs interactions, ce qui orientait naturellement vers une approche SMA. Les questions scientifiques sous-jacentes visent à comprendre et agir sur les

déterminants et la transmission des maladies infectieuses des animaux d'élevage par une approche multidisciplinaire et multi-échelles. La modélisation de ces questions devait donc intégrer des contraintes d'évolutivité et s'appuyer sur une structuration modulaire répartissant la complexité des différents domaines à traiter, à savoir la reproduction, la production laitière, l'alimentation, la génétique des animaux, la gestion de la population et les maladies (de production ou transmissibles). De plus, la solution devait être suffisamment générique pour intégrer de nouveaux concepts de manière réactive. Enfin, la stochasticité des modèles et la variété des scénarios à explorer, conduisant à de nombreuses répétitions, nécessitait de garantir les meilleures performances techniques possibles tout en préservant la reproductibilité des simulations.

2.2 Simulateurs de troupeaux laitiers

Le développement de simulateurs de troupeaux bovins laitiers intégrant l'impact économique des maladies de production n'est pas nouveau. ECOMAST (« *ECONomy MASTitis* ») a été développé à BIOEPAR pour des besoins spécifiques à partir de 1997. Son but était d'appuyer une thèse relative à l'efficacité économique de la maîtrise des infections intra mammaires en élevage bovin laitier (Hortet 2000). Il a ensuite fait l'objet d'évolutions au gré des questions scientifiques qui se sont succédé, par exemple en intégrant un sous-modèle génétique à l'occasion d'une thèse sur l'évaluation de l'intérêt du croisement entre des races laitières (Dezetter 2015). D'un point de vue technique, ECOMAST est écrit en Visual Basic et s'appuie sur une base de données Microsoft Access. Les produits issus de cette technologie vieillissante étant peu performants et difficilement maintenables, ECOMAST n'était plus en mesure d'évoluer au gré du progrès des connaissances scientifiques, de la disponibilité croissante de données réelles d'exploitation, ni pour intégrer de nouvelles questions de recherche.

Parmi les autres outils de simulation d'intérêt figure le simulateur danois SIMHERD (Østergaard et al. 2005) qui se base sur une gestion des individus et met en œuvre un large spectre de fonctions (production, reproduction, santé, comptabilité, alimentation). Si la prise en compte de ces sujets représentait un intérêt certain pour notre projet, la spécificité danoise, notamment en matière de conduite de troupeau, rendait difficile son réemploi. D'autres simulateurs comme DAFOSYM (Rotz et al. 1999) ou encore celui développé par Simone Kraatz (Kraatz 2012) proposaient des fonctionnalités intéressantes, mais leur axe fonctionnel était orienté vers les questions de nutrition et de maladies métaboliques uniquement.

2.3 Modèles épidémiologiques

Les modèles épidémiologiques mécanistes sont destinés à comprendre et prédire la propagation d'agents pathogènes, et à comparer des scénarios de maîtrise. Ils peuvent opérer à différentes échelles (lot d'animaux, troupeau, région, filière de production), ainsi que la dynamique de leurs vecteurs le cas échéant. Leur intérêt pour l'appui à la décision en santé (Ezanno et al. 2020) est désormais bien connu depuis la pandémie de Covid-19. Certains de ces modèles mettent en œuvre une gestion individuelle associée à des performances animales.

En revanche, ils sont généralement sujets à trois limitations. D'abord, ils n'intègrent pas la production de bilans économiques, pourtant nécessaires dans le cas des maladies de production. Ensuite, ces modèles se sont généralement focalisés sur la question scientifique traitant des voies de transmission et des leviers pour les réduire, ils intègrent peu les effets combinés de caractéristiques multidisciplinaires. En effet, la nature infectieuse des maladies traitées et la spécificité des leviers de maîtrise associés sont assez éloignées des problématiques rencontrées avec les maladies de production.

Enfin, la plupart de ces modèles sont développés en façon très spécifique à une maladie cible et manquent de généralité. Ce n'est que très récemment que se sont développées des approches ouvertes et génériques telles que des frameworks orientés objets comme Broadwick (O'Hare et al. 2016) ou un couplage entre un simulateur générique et un langage dédié à la modélisation épidémiologique tel qu'EMULSION (Picault et al. 2019) ou Kendrick (Mai Anh Bui et al. 2019). Néanmoins, l'intégration dans ces outils de modes de conduite d'élevage complexes reste encore une question de recherche (Sicard, Andraud, and Picault 2021).

2.4 Plateformes multi-agents

Une fois le cadre SMA identifié comme pertinent, l'utilisation d'une plateforme multi-agents générique a été étudiée. Repast HPC (Collier and North 2012) propose d'excellentes performances techniques obtenues notamment grâce à la parallélisation. Il permet d'agir finement sur l'architecture du système modélisé et ses caractéristiques opérationnelles, et offre un ordonnancement efficace pour la simulation des scénarios souhaités. Toutefois l'emploi de cette solution en soutien à la question scientifique d'un chercheur non-modélisateur aurait imposé le développement d'une interface logicielle qui facilite les interactions. Ce type d'interface est offert par la plateforme GAMA (Grignard et al. 2013), mais les performances techniques affichées par cette solution s'avéraient insuffisantes au regard du besoin.

Les limitations rencontrées sur les simulateurs de troupeaux existants, le manque de détails économiques et zootechniques dans les modèles épidémiologiques, et les besoins de performance et de contrôle sur le moteur, nous ont conduit à développer *de novo* un nouveau système de simulation. Le choix méthodologique qui a été fait pour répondre aux besoins consiste donc d'une part à s'appuyer sur de bonnes pratiques de génie

logiciel pour un développement performant et ouvert, et d'autre part à mobiliser UML (Rumbaugh, Booch, and Jacobson 2000) comme support d'échange entre disciplines, d'expression et de validation des hypothèses, ainsi que préconisé par Bersini (Bersini 2012) pour le développement de simulations multi-agents.

3 Le simulateur DHM

Le simulateur DHM vise à fournir un laboratoire virtuel permettant d'étudier des questions de recherche sur la santé des bovins laitiers. Il doit permettre d'expérimenter des scénarios diversifiés en termes d'échelle de temps, paramètres d'élevage, durée de simulation, nombre de répétitions, modèles comptables et protocoles. De plus, son architecture doit permettre à moyen terme une déclinaison pour d'autres conduites d'élevage, d'autres types de troupeaux voire d'autres espèces. Sa production devait donc s'inspirer des méthodes agiles pour privilégier l'interaction avec les scientifiques non-modélisateurs afin de pouvoir identifier les domaines attachés aux besoins fonctionnels potentiels et les intégrer dans une architecture multi-agents cohérente et extensible.

UML a donc été mobilisé pour définir les éléments constitutifs de l'élevage bovin laitier en y décrivant les agents, leur environnement ainsi que les règles de leurs interactions, en mettant en œuvre les *design patterns* (Gamma et al. 1994) idoines, le tout garantissant leur évolutivité descriptive.

3.1 Les agents et leur environnement

Les stratégies de maîtrise en santé animale et leur évaluation économique mobilisent un éventail disciplinaire très vaste, combinant différentes échelles (intra- ou inter-individuelles, groupes, troupeaux) et ciblant différentes espèces. En tant qu'outil de recherche, DHM est par ailleurs amené à incorporer des travaux de maturité variable. Une étude préliminaire des domaines

impliqués a donc été menée dès le lancement du projet. Des interviews auprès d'acteurs de différentes disciplines (épidémiologistes, modélisateurs, économistes, zootechniciens, parasitologues) ont permis d'identifier les notions pertinentes autour de la santé des animaux et de les positionner dans un méta-modèle multi-agents, avec l'animal pour élément central.

Pour DHM en particulier, l'agent central de la simulation de troupeau laitier a été déterminé comme étant la vache laitière : Agent Vache Laitière (AVL). Cet agent s'inscrit dans une hiérarchie de classes intégrant différents niveaux d'abstraction pour garantir la genericité et l'évolutivité en vue de faciliter des extensions ultérieures (figure 1).

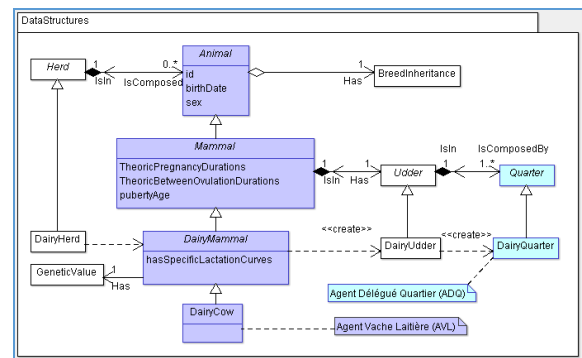


FIG. 1 - Définition statique des agents

Les agents dérivés du concept « *Mammal* » encapsulent un environnement « mamelle » (« *Udder* ») qui peut héberger des quartiers (« *Quarter* »). Cette modélisation multi-niveaux permet de déléguer une partie des fonctions biologiques à une échelle intra-individuelle tout en gagnant en modularité.

Les mammifères élevés spécifiquement pour la production laitière (« *DairyMammal* ») présentent une valeur génétique dédiée (« *GeneticValue* ») et disposent d'une mamelle dotée d'une capacité de production journalière (« *DairyUdder* »). C'est à ce niveau que se situent les « Agents Délégués Quartiers » (ADQ) qui ont leurs propres états et interactions. L'AVL « *DairyCow* » héberge quatre ADQ qui représentent les quatre trayons de la mamelle. Cette dernière constitue un environnement qui permet les

interactions entre l'AVL et les ADQ ou entre les ADQ, ce qui joue un rôle clé dans la propagation d'agents pathogènes responsables de maladies de production telles que les mammites. De même, les AVL interagissent au sein de l'environnement encapsulé par le troupeau. Ce dernier peut s'accroître par la naissance ou l'achat d'AVL qui évoluent ensuite de façon autonome selon leurs propres caractéristiques, en contribuant à l'activité productive de l'exploitation laitière et à la charge comptable associée (e.g. frais de santé, de reproduction et d'alimentation).

3.2 Comportements

Les actions des individus concernent les fonctions biologiques et comportementales, les effets de la conduite d'élevage (performances de production et de reproduction, santé, valeur génétique propre) et leurs éventuelles conséquences collectives (interactions liées à la reproduction et à la santé entre agents situés dans le même environnement). Tous ces comportements ont été modélisés par des machines à états.

Pour l'AVL, sont pris en compte l'avancement de son âge, sa situation reproductive (ovulation, gestation, vêlage...), son état de production laitière (environnement, tarissement...), l'apparition ou l'évolution d'une maladie, la prise de traitements médicaux, sa consommation d'aliment ou même sa mort. Pour l'ADQ, l'activité concerne principalement la quantité et la composition du lait produit, et l'état d'avancement des éventuelles maladies. Les mammites, dont les cas les plus graves peuvent aller jusqu'à occasionner l'arrêt de leur activité, impliquent des interactions avec les autres ADQ de la mamelle. La figure 2 illustre les relations impliquées dans la modélisation du processus biologique de reproduction, piloté par une gestion d'états. Les ovulations peuvent être détectées par l'éleveur qui procède à une insémination, et conduit en cas de succès à une gestation.

Certaines actions confiées aux agents peuvent dépasser leur responsabilité du fait de la complexité de la réponse à apporter. C'est le cas par exemple du calcul de la quantité de lait produite quotidiennement par un AVL, qui dépend pour partie de sa propre capacité du moment ou de celle de ses ADQ, mais également d'influences externes (nombre de traites quotidiennes, objectifs de production, période de l'année). Dans ce cas, les agents s'en remettent à des mécanismes de contrôle présents dans chacun des sous-modèles concernés (i.e. associés à chaque machine à états), ce qui facilite la modularité et la maintenance du modèle.

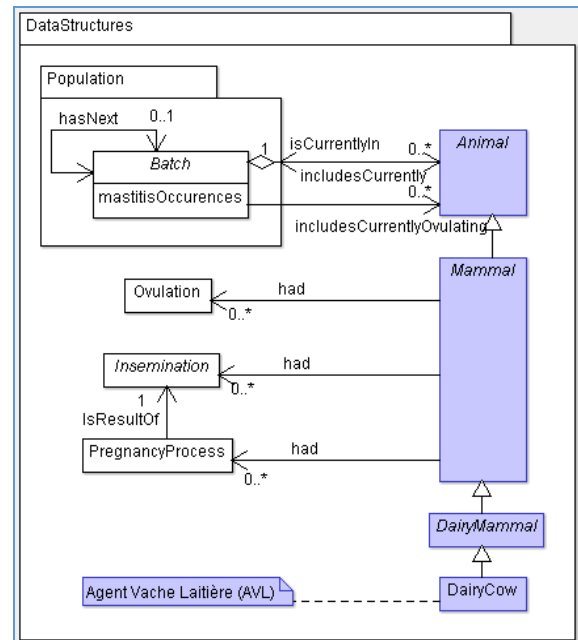


FIG. 2 - Définition des principales associations dynamiques de l'AVL

En ce qui concerne les interactions entre AVL, elles sont actuellement mises en œuvre dans trois domaines précis : le vêlage, l'effet de groupe des chaleurs et la contagiosité de certaines maladies, avec pour environnement le lot dans lequel ils se trouvent (« *Batch* », figure 2). Le vêlage est l'interaction la plus forte entre deux individus, car il s'agit pour un AVL mère d'instancier un ou plusieurs autres AVL veaux, avec transmission d'une partie de son patrimoine génétique. Pour ce qui est de l'effet de groupe, l'expression

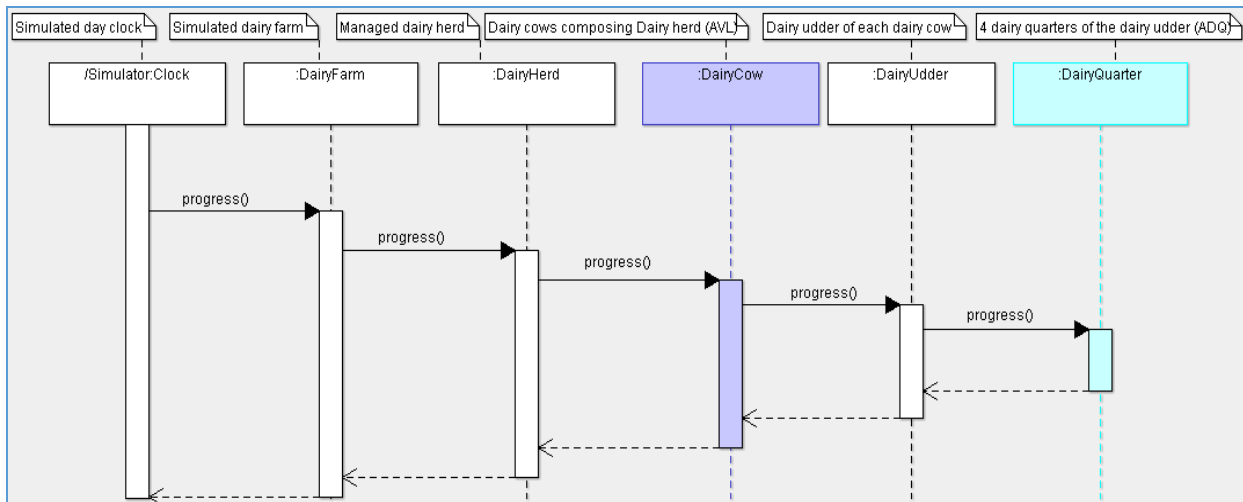


FIG. 3 – Enchaînement du signal de progression jusqu'à et à travers l'agent

comportementale de l'ovulation d'une vache (l'œstrus) peut favoriser celle de ses congénères dans la même situation et présentes dans le même environnement, ce qui peut faciliter sa détection par l'éleveur. Pour cela l'agent AVL dépose dans son environnement « lot » l'information relative à son entrée en ovulation.

Enfin pour la contagiosité, la probabilité qu'une vache soit atteinte par la maladie peut être accentuée par la prévalence de la maladie dans son environnement que constitue également son lot. C'est le cas des mammites dont l'incidence augmente si d'autres vaches du lot sont également concernées. Cette interaction existe aussi au niveau de la mamelle puisque l'état sanitaire de chaque ADQ peut également exercer une influence sur la probabilité d'apparition d'une mammite sur les ADQ voisins.

3.3 Parallélisation et ordonnancement

L'ordonnancement du simulateur a été conçu pour garantir la reproductibilité des expériences. Les modèles manipulés étant stochastiques, chaque répétition d'une expérience est associée à une graine du générateur pseudo-aléatoire. Les exécutions peuvent alors être parallélisées sans préjudice pour la reproductibilité des résultats. En ce qui concerne la dynamique de simulation, le

système modélisé est caractérisé par une échelle temporelle qui rythme les actions des agents de l'élevage à simuler. Les comportements de l'AVL et de l'ADQ sont rythmés par l'horloge selon un signal qui se propage d'agent en agent (figure 3). Ce signal se déclenche une fois par jour simulé pour que les agents réalisent en toute autonomie les actions de progression liées à leur état, en tenant compte des circonstances.

3.4 Mise en œuvre

DHM a été implémenté en C++ dans une architecture orientée objets de façon à faciliter la mise en œuvre des concepts SMA. UML a été employé avec les utilisateurs pour définir la spécification des agents et leurs comportements. Afin de maximiser les performances du simulateur, la parallélisation des répétitions a été utilisée en s'appuyant sur la sérialisation binaire pour le transfert de données et en exploitant la technologie « *OpenMP* » (OpenMP 2021). Le bénéfice en temps de calcul a pu être constaté lors de la première exploitation du simulateur sur les serveurs de calcul du Centre de Calcul Intensif des Pays de la Loire (CC IPL 2021).

Le déploiement proposé sous la forme d'un composant ré-employable (bibliothèque dynamique multiplateforme) offre une réponse à l'objectif visé d'une mise en œuvre

diversifiée (application standalone, web, ...).

4 Une première application : MO3SAN

DHM est un simulateur destiné à s'étendre, en accueillant au fil du temps et de manière itérative diverses questions nouvelles. Le projet global santé en Montbéliarde, MO3SAN, a été le premier à mettre en œuvre DHM. Son objectif est de développer des méthodes et des indicateurs nouveaux pour mettre à disposition des éleveurs laitiers des références et moyens de maîtrise des principales maladies bovines par l'amélioration des pratiques et la génétique (MO3 2019). DHM a été choisi comme support pour la simulation des scénarios devant intégrer en particulier les règles de gestion relatives à la cétose, maladie métabolique des bovins laitiers (Jutel 2021). L'intégration des fonctionnalités nécessaires pour y répondre a été facilitée par l'ouverture rendue possible avec la généricité de la modélisation. Elle est passée par la mise en œuvre d'un processus qui a permis *in fine* la production des résultats attendus. Ce processus visait dans un premier temps à déterminer comment des facteurs de risque associés à la prévalence d'une maladie dans le troupeau pouvaient influencer sur l'état de santé des animaux, et par voie de conséquence sur leurs performances de production déterminant le bilan économique de l'élevage (figure 4). Cette démarche relativement classique a permis de définir le besoin fonctionnel en vue de la simulation.

Nous avons décrit les différents niveaux de gravité des cétozes que pouvait subir une vache et les conséquences individuelles sur chacun des aspects la concernant (reproduction, production, létalité, ...). La dynamique associée, les niveaux d'incidence et leurs modulations éventuelles, la détectabilité et les traitements médicaux prodigués ont pu être mis en œuvre de façon individuelle. Des leviers de conduite d'élevage ont été créés sous la forme de

paramètres d'entrée pour permettre à l'utilisateur de manipuler le troupeau selon le scénario désiré en relation avec la maladie. Ils concernent le niveau général de prévention des cétozes de l'élevage, la mise en œuvre de moyens complémentaires de détection et la description des effets des traitements médicaux prodigués (délai d'action, effet sur la production laitière et la reproduction, probabilité de succès et de réduction du risque d'aggravation). Egalement paramétrables selon les scénarios, les données tarifaires liées aux activités et événements du modèle cétose ont été introduites dans le modèle comptable, contribuant à la constitution du bilan économique.

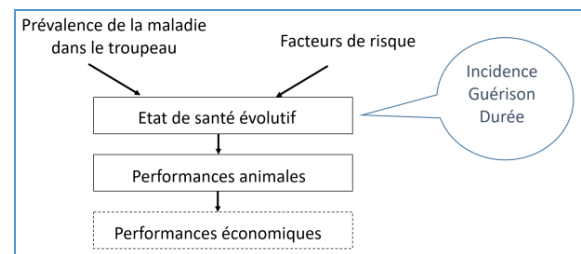


FIG. 4 - Impact des agents pathogènes et des facteurs de risque sur les performances économiques

Cette modélisation à base d'agents a permis de reproduire les conditions réalistes de la maîtrise des maladies de production au sein du troupeau et de fournir des résultats en conséquence. Pour autant, l'intégration de ce nouveau modèle de maladie dans le simulateur a conduit à un réajustement de certaines constantes générales. Par exemple, la prise en compte du taux de mortalité propre à la cétose a nécessité que soit recalculé celui appliqué jusqu'alors, qui traduisait la mortalité générale du troupeau quelle qu'en soit l'origine. Le principe adopté a été d'opérer un calibrage logiciel de la mortalité en fonction des résultats obtenus à l'issue de la simulation du scénario par défaut (intégrant nouvellement la cétose), de manière à ce que ces résultats respectent les données d'observation et la littérature.

Les simulations réalisées pour MO3SAN ont mis en œuvre trois types différents d'élevage.

Leur spécificité résidait principalement au niveau de la dimension du troupeau et son taux de renouvellement, des modalités de mise à la reproduction, de la pratique du pâturage (influant les taux d'incidence des mammites), des rations alimentaires, des niveaux de production et enfin des prix pratiqués. Pour chacun de ces élevages, deux niveaux de fréquence combinés des mammites et des cétooses ont été déterminés en modulant le niveau de prévention ce qui a donc conduit à paramétrer 12 élevages différents. Chaque scénario a donné lieu à 500 répétitions stochastiques (Jutel 2021) pour obtenir une distribution stable des sorties.

Avant que ne soit lancée la production des résultats de simulation, il a été nécessaire de réaliser un calibrage fonctionnel des données moyennes d'entrée. En effet, certains paramètres permettent de positionner un objectif pour l'expérimentateur (quantité de lait à produire, rémunération, ...), mais du fait de la multiplicité de facteurs impliqués, un écart à l'attendu peut être constaté, nécessitant un ajustement.

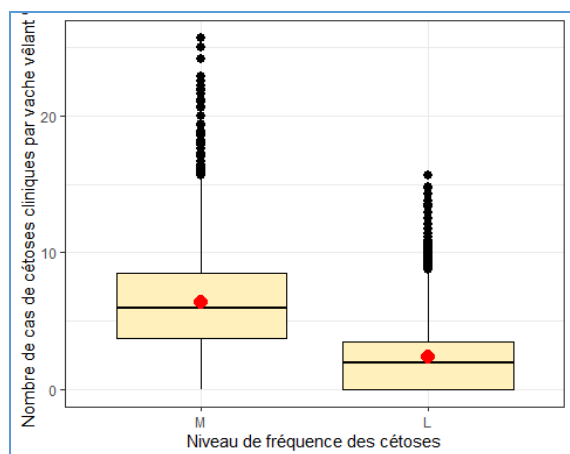


FIG. 5 - Variabilité du nombre de cas de cétooses cliniques par vache vêlant en fonction du niveau de fréquence des cétooses

Une fois les données d'entrée calibrées, les élevages ont été simulés dans les conditions prévues, occasionnant 6000 répétitions tous scénarios confondus et un nombre total de plus d'un milliard de sollicitations d'AVL, et de près de deux milliards en ce qui concerne les ADQ d'AVL adultes. La durée totale de la

simulation a été de l'ordre de 10 minutes, ce qui a représenté 1/10^{ème} de seconde en moyenne pour une répétition.

Les résultats obtenus ont permis de comparer l'effet des différents scénarios dans chacun des domaines modélisés (production laitière et contrôles laitiers, gestion de la population, performance de reproduction, états de santé mammites et cétooses, et enfin bilan comptable des charges et produits de l'exploitation). La figure 5 est un exemple de représentation du niveau de fréquence des cas de cétooses occasionnés par la conjonction ramenée à l'individu des divers facteurs d'influence (génétique, rang de lactation, niveau de production laitière, durée de la lactation précédente, présence d'autres maladies comme les mammites, actions de prévention). La comparaison entre le nombre relevé de cas de cétoose subi par chacune des vaches pour le scénario moyen (M) avec celui mettant en œuvre une conduite de troupeau plus préventive (L) traduit d'un point de vue fonctionnel l'intérêt d'une bonne pratique. Plus généralement, l'analyse des effets comptables des différentes pratiques préventives a permis de démontrer leur intérêt économique (voir tableau 1).

TAB. 1 - Perte de marge brute de l'activité de production laitière (€/1000 litres livrés)

Elevage	Grand-Ouest	AOP Comté	Massif Central
Effet des cétooses	-2.2	-6.5	-9.0
Effet des mammites	-6.8	-7.6	-11.1

5 Conclusion et perspectives

Le besoin de disposer d'un simulateur de troupeau laitier permettant à un expérimentateur d'évaluer des stratégies de maîtrise des maladies de production en fournissant des bilans économiques a conduit au développement d'un nouveau logiciel : DHM.

L'évolutivité et la performance de l'outil produit ont démontré la pertinence du choix ayant consisté à réaliser en propre une archi-

teature de type SMA. Grâce à la gestion individuelle des vaches laitières modélisées sous forme d'agents, les effets sur les performances des animaux en la présence d'agents pathogènes combinée avec des facteurs de risques dépendant de chaque situation ont pu être évalués tant au niveau individuel que collectif. Ainsi, par sa prise en compte fine des interactions multifactorielles et complexes, ce SMA produit des résultats aussi fidèles à des situations réelles que réalistes pour des scénarios expérimentaux.

L'approche méthodologique adoptée consistant à intégrer d'emblée une dimension générique à la modélisation pour faciliter l'intégration itérative et en circuit court des réponses aux besoins fonctionnels exprimés a posteriori a fait la preuve de son intérêt. Tout cela participe à la recherche menée avec des approches pluridisciplinaires et multi-échelles dans le domaine de la santé des animaux d'élevage.

En ce qui concerne l'état de la connaissance scientifique, référence de la modélisation fonctionnelle, la base décrite dans la littérature est dans la plupart des cas obtenue dans un cadre restreint (une race, une région, ...) ou à l'inverse à une large échelle (performances animales publiées annuellement, publications agrégées, ...). La modélisation de ces différents grains de connaissance peut être problématique pour une implémentation dans DHM qui nécessite de détenir l'exhaustivité des règles de gestion à appliquer pour tous les contextes modélisés. L'accès à des données précises d'observation ou à minima les avis d'expert peuvent aider à contrecarrer cet inconvénient.

Si DHM est actuellement pleinement exploitable, du travail reste à réaliser, notamment des travaux d'analyses de sensibilité qui permettront d'identifier les leviers de conduite les plus influents sur les résultats clé afin d'en consolider la modélisation.

Pour ce qui est de la recherche en épidémiologie, une étude pourrait être lancée pour évaluer l'intérêt d'une connexion avec le langage

de modélisation épidémiologique du simulateur EMULSION (Picault et al. 2019). Cela pourrait faciliter la prise en compte par DHM de maladies contagieuses pouvant être rencontrées au sein du troupeau, qui auraient fait l'objet de modélisations antérieures moins précises en termes de conduite d'élevage.

Enfin, pour ce qui est des applications pratiques, DHM va être mis en œuvre dans le cadre de projets menés en partenariat public et privé, et pour un appui aux politiques publiques.

Remerciements

Nous remercions Christine Fourichon, ancienne directrice de BIOEPAR pour le partage de sa longue expérience en simulation de la santé animale, Arnaud Rault et Vianney Sicard de l'équipe Dynamo pour leur lecture avisée, Claire Jutel première utilisatrice de DHM, le CCIPL (CCIPL 2021) pour les ressources de calcul mises gracieusement à notre disposition et enfin Marc Jaouen pour ses précieux enseignements.

Références

- Bersini, Hugues. 2012. "UML for ABM." *Journal of Artificial Societies and Social Simulation* 15 (1): 9. <https://doi.org/10.18564/jasss.1897>.
- CCIPL. 2021. "Le Centre de Calcul Intensif Des Pays de La Loire." 2021. <https://ccipl.univ-nantes.fr/>.
- Collier, Nicholson, and Michael North. 2012. "Parallel Agent-Based Simulation with Repast for High Performance Computing." *SIMULATION* 89 (10): 1215–35. <https://doi.org/10.1177/0037549712462620>.
- Dezetter, C. 2015. "Evaluation of the Interest of Crossbreeding between Dairy Cattle Breeds."
- Ezanno, P., M. Andraud, G. Beaunée, T. Hoch, S. Krebs, A. Rault, S. Touzeau, E. Vergu, and S. Widgren. 2020. "How Mechanistic Modelling Supports Decision Making for the Control of Enzootic Infectious Diseases." *Epidemics* 32 (May): 100398. <https://doi.org/10.1016/j.epidem.2020.100398>.
- Gamma, E, R Helm, R Johnson, and J Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series.*
- Gontier, Philippe, and Nathalie Bareille. 2021. "Dairy

- Health Manager.” 2021. <http://www.dairyhealthmanager.fr>.
- Grignard, Arnaud, Patrick Taillandier, Benoit Gaudou, Duc An Vo, Nghi Quang Huynh, and Alexis Drogoul. 2013. “GAMA 1.6: Advancing the Art of Complex Agent-Based Modeling and Simulation.” In *PRIMA 2013: Principles and Practice of Multi-Agent Systems*, edited by Guido Boella, Edith Elkind, Bastin Tony Roy Savarimuthu, Frank Dignum, and Martin K Purvis, 117–31. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Hortet, Philippe. 2000. “Evaluation Ex-Ante de l'efficacité Économique Des Programmes de Maîtrise Des Infections Intramammaires En Élevage Bovin Laitier.”
- Jutel, Claire. 2021. “Estimation de l'impact Technico-Économique Des Mammites et Des Cétoses Par Simulation Dans Trois Systèmes d'élevages En Race Montbéliarde.”
- Kraatz, Simone. 2012. “Energy Intensity in Livestock Operations - Modeling of Dairy Farming Systems in Germany.” *Agricultural Systems* 110: 90–106. <https://doi.org/10.1016/j.agsy.2012.03.007>.
- Mai Anh Bui, T., Nick Papoulias, Serge Stinckwich, Mikal Ziane, and Benjamin Roche. 2019. “The Kendrick Modelling Platform: Language Abstractions and Tools for Epidemiology.” *BMC Bioinformatics* 20 (1). <https://doi.org/10.1186/s12859-019-2843-0>.
- MO3. 2019. “Projet Global Santé En Montbéliarde.” 2019. <http://www.mo3.fr/2019/04/03/mo3san-projet-global-sante-en-montbeliarde/>.
- O'Hare, Anthony, Samantha J. Lycett, Thomas Doherty, Liliana C.M. Salvador, and Rowland R. Kao. 2016. “Broadwick: A Framework for Computational Epidemiology.” *BMC Bioinformatics* 17 (1): 1–6. <https://doi.org/10.1186/s12859-016-0903-2>.
- OpenMP. 2021. “The OpenMP API Specification for Parallel Programming.” 2021. <https://www.openmp.org/>.
- Østergaard, S., M. G.G. Chagunda, N. C. Friggens, T. W. Bendedgaard, and I. C. Klaas. 2005. “A Stochastic Model Simulating Pathogen-Specific Mastitis Control in a Dairy Herd.” *Journal of Dairy Science* 88 (12): 4243–57. [https://doi.org/10.3168/jds.S0022-0302\(05\)73111-8](https://doi.org/10.3168/jds.S0022-0302(05)73111-8).
- Picault, Sébastien, Yu Lin Huang, Vianney Sicard, Sandie Arnoux, Gaël Beaunée, and Pauline Ezanno. 2019. “EMULSION: Transparent and Flexible Multiscale Stochastic Models in Human, Animal and Plant Epidemiology.” *PLoS Computational Biology* 15 (9): 1–13. <https://doi.org/10.1371/journal.pcbi.1007342>.
- Rotz, C. A., D. R. Mertens, D. R. Buckmaster, M. S. Allen, and J. H. Harrison. 1999. “A Dairy Herd Model for Use in Whole Farm Simulations.” *Journal of Dairy Science* 82 (12): 2826–40. [https://doi.org/10.3168/jds.S0022-0302\(99\)75541-4](https://doi.org/10.3168/jds.S0022-0302(99)75541-4).
- Rumbaugh, James, Grady Booch, and Ivar Jacobson. 2000. *Le Guide de l'utilisateur UML*. Eyrolles.
- Sicard, Vianney, Mathieu Andraud, and Sébastien Picault. 2021. “L'organisation Comme Design Pattern Dans Les Systèmes Multi-Agents Multi-Niveaux.” In *Actes Des 29e Journées Francophones Sur Les Systèmes Multi-Agents (JFSMA '21)*, 73–82. Cépadauès.

Méta-modèle d'agent pour la génération de comportements variables induits par des modèles cognitifs modulaires

T. de Blauwe^a
tristan.de-blauwe@hds.utc.fr

N. Sabouret^b
nicolas.sabouret@universite-paris-saclay.fr

D. Lourdeaux^a
domitile.lourdeaux@hds.utc.fr

^a Alliance Sorbonne Université,
UTC Heudiasyc, CNRS, UMR 7253
Compiègne, France

^b Laboratoire Interdisciplinaire des Sciences du Numérique,
LISN, CNRS, UMR 9015,
Université Paris-Saclay, France

Résumé

Cet article se place dans le contexte du développement d'agents virtuels pour la formation professionnelle. Nous proposons un méta-modèle permettant d'intégrer dans les modèles d'agents une plus grande variété de comportements cognitifs, sociaux ou affectifs. La définition d'un agent se fait de manière modulaire à partir d'opérations de bases définissant le modèle d'agent et d'un ensemble de processus cognitifs qui complètent ou modifient ces opérations. Nous présentons une première évaluation avec un modèle d'agent et trois modèles cognitifs. Nous montrons que les modèles cognitifs impactent de manière intelligible le comportement de l'agent.

Mots-clés : Méta-Modèle d'agent, Agent cognitif, modèles cognitifs, comportements

Abstract

This article is set in the context of virtual characters's development for training. We propose a meta-model allowing to integrate in the agent models a wider variety of cognitive, social or affective behaviours. The definition of an agent is done in a modular way from basic operations defining the agent model and from a set of cognitive processes that complete or modify these operations. We present a first experimentation with one agent model and three cognitive models. We show that the cognitive models impact the behaviour of the agent in an intelligible way.

Keywords: Agent meta-model, cognitive agent, cognitive models, behaviours

1 Introduction

Les Personnages Virtuels Autonomes (PVAs) sont des agents évoluant au sein d'un Envi-

ronnement Virtuel (EV). Ils interagissent avec leur environnement, ainsi qu'avec l'utilisateur, généralement représenté par un avatar. Une application de ces PVAs consiste à simuler des opérateurs humains, de divers métiers, dans un contexte d'entraînement professionnel [1, 2, 3]. Les agents doivent alors exprimer des comportements complexes, i.e cognitifs, émotionnels et/ou sociaux, afin de reproduire les comportements observés. Le comportement des PVAs est observable aux travers des actions qu'ils réalisent sur les éléments de l'environnement.

Ces actions sont le produit d'un modèle d'agent, défini par un ensemble d'opérations c'est-à-dire de mécanismes de traitements d'informations, allant de la perception à la réalisation d'actions, suite à une prise de décision, en passant par la communication avec les autres agents [4].

Pour mettre en oeuvre différents comportements, les informaticiens implémentent, au sein d'un modèle d'agent, un ou plusieurs modèles cognitifs, généralement inspirés des Sciences Humaines et Sociales (SHS). Ces modèles définissent la manière dont les opérations calculent les actions et les communications à effectuer en fonction des perceptions. Par exemple, l'architecture BEN [5] intègre plusieurs modèles cognitifs au sein d'une architecture d'agents BDI. Elle lie des modèles cognitifs, comme le modèle OCC [6] pour les émotions, au modèle de personnalité OCEAN [7] afin d'obtenir des comportements cognitifs, affectifs et sociaux. Une autre architecture comme EMA [8] intègre une composante émotionnelle avec la théorie de l'évaluation cognitive des émotions proposée par Smith et Lazarus [9], au sein de l'architecture d'agents SOAR [10].

Cependant, dans un contexte de simulation, les comportements et les modèles cognitifs des

agents dépendent fortement du domaine d'application et des objectifs de la simulation (qu'il s'agisse d'un scénario de formation, d'un jeu ou d'un outil d'aide à la décision). Concevoir une architecture de PVAs générique est donc difficile. Tout d'abord, on ne connaît pas a priori les modèles cognitifs à intégrer et à combiner, ainsi que les comportements à générer. Qui plus est, les différentes combinaisons de modèles impactent le comportement à différents niveaux, mais le résultat doit rester intelligible. Nous entendons par là que le modélisateur veut retrouver dans la simulation, les comportements métiers observés, c'est-à-dire les séquences d'actions qui lui semblent plausibles. Prenons un exemple concret : en situation de crise, les médecins secouristes peuvent être amenés à s'éloigner légèrement de la réglementation. Par exemple, dans une séquence d'actions de soin d'une blessure, ils peuvent utiliser un chiffon pour gagner de précieuses secondes au lieu de mettre des gants pour faire un point de compression. L'apparition d'un tel comportement, suite à l'ajout d'un modèle cognitif qui décrit cette possibilité de déviation, serait un exemple de résultat intelligible.

Ainsi, notre objectif dans cet article est de présenter un méta-modèle qui répond à cette double difficulté : permettre l'intégration de modèles cognitifs dans un modèle d'agent, avec un minimum d'hypothèses sur le modèle d'agent et sur le modèle cognitif, tout en s'assurant que les comportements produits restent intelligibles.

Le plan de l'article est le suivant. Dans la section 2, nous décrivons l'état de l'art sur les approches et modèles cognitifs utilisés en simulation. Dans la section 3, nous présentons le méta-modèle. Dans la section 4, nous appliquons notre approche sur un projet de formation médicale. Dans la section 5, nous concluons sur les avantages et inconvénients de notre approche, ainsi que les perspectives envisageables.

2 État de l'art

Nous sommes intéressés par l'opérationnalisation de modèles cognitifs au sein d'un modèle d'agent afin d'obtenir des comportements intelligibles. Nous présentons d'abord ce qui caractérise ces modèles cognitifs et nous montrons en quoi leur opérationnalisation est difficile. Nous détaillons ensuite quelques modèles cognitifs qui seront utilisés dans la suite de l'article. Enfin, nous présentons quelques modèles d'agent classiques en simulation multi-agent et la manière dont ils ont intégré des modèles cognitifs.

2.1 Modèles cognitifs

Selon le dictionnaire de psychologie de l'Association Américaine de Psychologie¹ (APA), un modèle cognitif est une représentation simplifiée et théorique d'un ensemble de processus cognitifs/mentaux amenant à des phénomènes observables et qui fournit des explications sur la cause. Selon la même source, un processus cognitif est n'importe quelle fonction impliquée dans des activités telles que l'attention, la perception, l'apprentissage et la résolution de problèmes.

La littérature en SHS propose un grand nombre de modèles cognitifs selon les processus à modéliser. Dans le domaine des processus socio-affectifs, par exemple, on peut citer la théorie de l'évaluation cognitive des émotions proposée par Smith et Lazarus [9], le modèle OCEAN [7] pour la personnalité ou en encore le modèle OCC [6] très souvent utilisé en informatique pour opérationnaliser l'évaluation cognitive [6].

L'étude des travaux en informatique, qui implémentent de tels modèles, met en évidence la difficulté de généraliser l'opérationnalisation d'un modèle cognitif donné. Ainsi, le même modèle OCC peut être implémenté de différentes manières selon les modèles d'agent utilisés (par exemple, un modèle BDI [5] ou un modèle à base de règles [11]). En effet, un modèle peut impacter différentes opérations au sein du modèle d'agent, que ce soit pour la perception, pour la délibération, pour la mise à jour des données, pour la sélection d'action ou pour la communication. Il est donc difficile de proposer une opérationnalisation sur une seule de ces dimensions, car la manière dont on implémente le modèle cognitif pour une opération de perception peut impacter la manière dont il faut altérer le processus de décision. Cela est d'autant plus vrai que l'on souhaite obtenir un résultat intelligible, c'est-à-dire dans lequel l'utilisateur perçoit une cohérence entre les différentes étapes du modèle d'agent.

2.2 Modèles cognitifs et travail d'équipe

Nous nous intéressons plus particulièrement aux comportements d'équipes en situation de crise. Nous présentons dans cette section quelques modèles cognitifs qui seront utilisés pour illustrer notre approche.

Le premier modèle est celui proposé par Driskell et Salas [12], qui ont réalisé des études consé-

1. <https://dictionary.apa.org>

quentes sur les comportements d'équipes en situation extrêmes. Ils proposent ainsi « un nombre limité de mécanismes cognitifs, émotionnels et sociaux à travers duquel le stress impacte les performances » des membres d'une équipe. Un des phénomènes décrit est que le stress augmente la distraction et décroît l'attention, ce qui pourrait entraîner des erreurs de manipulation ou des oublis de la part d'un agent. Ici, on observe que le modèle cognitif impacte directement la perception de l'agent, les conditions d'exécution et indirectement la sélection d'action de par ses connaissances partielles erronées. Un autre mécanisme proposé dans le modèle de Driskell et Salas pourrait impacter d'autres opérations de l'agent : « sous stress, les personnes ont tendance à moins s'entraider. » [13]. L'opérationnalisation de ce modèle au sein d'un modèle d'agent présente donc un double défi, d'une part en raison de la variété des comportements décrits et d'autre part parce qu'il intervient sur les différentes opérations du modèle d'agent.

Ensuite, afin de combiner ce modèle avec d'autres, nous considérons des modèles déjà présents dans les travaux précédents de notre équipe. D'une part, nous avons les comportements de « *followership* » décrits par Demary [14] et résumés dans le tableau 1. En fonction de son profil, l'agent réagit différemment lorsqu'il reçoit un ordre de son supérieur (voir Figure 1).

Enfin, nous avons les activités limites tolérées par l'usage [15], amenant à des comportements non-réglementaires. Un exemple de tels comportements serait qu'en situation de crise, un médecin secouriste s'éloigne légèrement de la réglementation. Il utiliserait ainsi un chiffon pour gagner de précieuses secondes au lieu de mettre des gants pour faire un point de compression.

2.3 Modèles d'agent en simulation

Un modèle d'agent consiste à concevoir une boucle perception-raisonnement-action qui produit un comportement. L'opérationnalisation de modèles cognitifs au sein de ce modèle permet de générer des comportements plus humains et intelligibles.

Par exemple, REPLICANTS [16] est un modèle d'agent raisonnant sur le formalisme ACTIVITY-DL, inspiré des HTNs [17]. Ce formalisme modélise l'activité humaine de manière hiérarchique, proche de la représentation mentale des experts. Le parcours de l'arbre d'activité retourne un ensemble de tâches envisageables,

	Communicant	Non-Communicant
Proactif	Je raisonne, j'informe le leader, je conteste si besoin, j'obéis au final (Exemplaire / Courageux)	Je prend des décisions, je conteste (Aliéné / Individualiste)
Passif	Je demande quoi faire et je fais ce qu'on me dit (Mouton / Faiseur)	Si on ne me dit rien je suis la procédure. Sinon, je fais ce qu'on me demande (Yes people / Ressource)

FIGURE 1 – Tableau tiré de [14] décrivant le comportement en fonction du style de *Followership* et le style de *Communicant*

ainsi que leur priorité, évaluée selon les préconditions. L'intégration de modèles cognitifs dans REPLICANTS se fait en modifiant ce calcul de priorité par l'ajout de bonus/malus en fonction du profil de l'agent et des propriétés de la tâche considérée. Par exemple, l'incorporation du modèle Demary [14] ajoute un bonus lorsque la tâche correspond à un ordre et que l'agent a un profil passif. Le modèle de Fadier [15] applique un malus selon les conditions réglementaires et le profil de l'agent. Intégrer de nouveaux modèles cognitifs nécessite donc de réviser ce calcul de priorité. La difficulté est alors de s'assurer que l'ensemble de modèles restent toujours impactant et que les comportements résultats sont intelligibles.

Un autre modèle qui est souvent utilisé dans les systèmes multi-agents est le modèle BDI [18]. Ainsi, dans la communauté JFSMA, on peut citer les travaux de Bourgeois *et al.* [5] qui proposent l'architecture normative d'agents sociaux émotionnels BEN. Cette architecture intègre plusieurs modèles cognitifs, comme OCC [6], avec le modèle de personnalité OCEAN [7]. Le choix de conception qui est fait consiste à relier les modèles cognitifs aux paramètres du modèle OCEAN, ce qui limite les paramètres à définir. Une force de cette architecture est d'offrir une certaine modularité : le modélisateur peut choisir de n'utiliser que la prise de décision sans tenir compte de l'évaluation d'émotions. Toutefois, l'ensemble des opérations dépend de l'approche BDI : il n'est pas possible d'envisager une autre architecture de prise de décision.

2.4 Synthèse

Bien qu'il existe de nombreux modèles d'agent proposant une implémentation et une combinaison fine de différents modèles cognitifs, les choix de conception qui sont faits contraignent le modélisateur à s'adapter au modèle. À mesure que les besoins évoluent, les utilisateurs peuvent vouloir mettre en œuvre d'autres modèles d'agents pour la prise de décision ou

d'autres modèles cognitifs pour produire de nouveaux comportements. Ces modifications sont difficiles, voire impossibles, dans ces modèles.

C'est pourquoi nous proposons dans cet article un méta-modèle d'agent, c'est-à-dire un modèle pour combiner ensemble plusieurs modèles cognitifs au sein d'un modèle d'agent. L'ensemble d'opérations du modèle d'agent est donc modifié et complété selon la combinaison de modèles cognitifs considérés. Notre méta-modèle doit permettre de programmer n'importe quel modèle d'agent.

Pour illustrer notre travail dans un contexte de formation, nous utilisons le modèle d'agent REPLICANTS [16] et les modèles cognitifs de Demary [14] et Driskell [13]. Nous montrerons comment notre méta-modèle permet de les intégrer simplement, d'une manière qui peut se généraliser à n'importe quel autre modèle d'agent et n'importe quels autres modèles cognitifs.

3 Méta-modèle OPACK

Le méta-modèle OPACK est un modèle de modèle d'agent pour la génération de comportements induite par des modèles cognitifs, dans un contexte de simulation interactive. Nous cherchons à faciliter l'intégration de modèles cognitifs, tout en obtenant des comportements intelligibles. Comme nous le verrons, ce modèle fait le moins d'hypothèses possibles sur le modèle d'agent et sur les modèles cognitifs utilisés.

3.1 Cas d'étude

Pour présenter le fonctionnement de notre méta-modèle, nous nous appuyerons sur le cas d'étude suivant : nos agents sont des médecins secouristes. Ils doivent réaliser une procédure médicale sur un patient. Les observations terrain et les retours d'experts indiquent que cette procédure varie selon les scénarios et l'état des agents.

Notre objectif est rendre compte de cette variabilité grâce à des modèles cognitifs. Nous nous fondons sur un modèle d'agent REPLICANTS, déjà utilisé pour ce cas d'étude.

3.2 État d'un agent

L'état d'un agent est composé des éléments suivants :

P : ensembles des percepts, soit ce qui est

actuellement perceptible par l'agent (mis à jour à chaque cycle d'exécution) ;

A : ensembles des actions actuellement menées par l'agent ;

C : ensembles des caractéristiques de l'agent, pouvant représenter des variables physiologiques ou psychologiques comme le stress, ou bien des paramètres stables dans le temps comme la personnalité ou les qualifications ;

K : ensembles des connaissances de l'agent, qui comprend non seulement les croyances de l'agent mais aussi les paramètres des actions, les plans éventuels, *etc.*

Le modélisateur définit chacun de ces éléments en fonction du domaine, mais il n'a pas l'obligation de tous les utiliser. Par exemple, il est possible de programmer un agent réactif dans lequel **K** sera vide.

3.3 Opérations

La brique de base de notre méta-modèle est la notion d'*opération*. Une opération est une fonction avec des entrées, une seule sortie et un rôle.

Chaque opération a accès à l'ensemble de l'état de l'agent. Les entrées et la sortie sont caractérisées par un type et une valeur — par exemple, l'action (type) « poser un garrot (valeur) » ou le percept (type) « le patient est sain » (valeur). Le cas échéant, il est possible de définir des sous-types — par exemple, l'état du patient est un sous-type de percept.

Le rôle de l'opération définit la signature de l'opération (c'est-à-dire ses entrées et sorties). Les noms des rôles proviennent des fonctions mentales décrites par l'APA² :

— **Manipulation** : aucune entrée et aucune sortie (l'opération travaille uniquement à partir de l'état de l'agent) ;

— **Acquisition** : aucune entrée (sauf l'état de l'agent) mais une sortie ;

— **Storage** : des entrées mais aucune sortie (l'opération agit sur l'état de l'agent) ;

— **Interpretation** : des entrées et une sortie ;

— **Transformation** : cas particulier d'interprétation où une des entrées est de même type que la sortie : c'est notre donnée manipulée. Les autres entrées ainsi que l'état permettent de contrôler la modification apportée à la donnée manipulée par l'opération.

L'ensemble d'opérations d'un agent est noté **O**.

2. <https://dictionary.apa.org>

Exemple Pour implémenter le modèle d'agent REPLICANTS dans le méta-modèle OPACK, une opération importante est la sélection d'action (sortie) à partir d'actions candidates (entrée) qui est une opération d'*interprétation*. Une autre opération importante est l'acquisition d'actions envisageables, notamment grâce au parcours de l'arbre qui permet d'extraire des actions candidates en fonction de l'état de la simulation. Cette opération d'*acquisition* ne reçoit pas d'entrée, mais utilise l'état de l'agent (dont l'arbre ACTIVITY-DL qui est dans **K** et les caractéristiques de l'agent **C**) pour calculer un ensemble d'actions.

Ces deux opérations sont illustrées sur la figure 2. Comme on le voit sur cette figure, la sortie d'une opération est reliée aux entrées d'autres opérations : nous parlons de *flux d'opérations*.

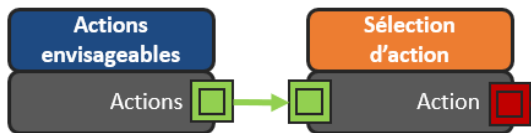


FIGURE 2 – Exemples de deux opérations pour implémenter le modèle d'agent REPLICANTS.

3.4 Flux d'opérations

Le modélisateur définit les opérations et les relie en un flux d'opérations, à la manière de la programmation par flux de Morrison [19].

Concrètement, un flux d'opérations est un graphe orienté acyclique dont les sommets sont des opérations et dont les arcs représentent les flux de données entre la sortie d'une opération et les entrées des suivantes. En complément, le modélisateur peut définir des arcs de précedence entre deux opérations, qui ne correspondent pas à un échange de données mais à un ordre d'exécution.

La figure 3 page suivante donne un exemple de flux. Les couleurs des étiquettes des sommets correspondent aux rôles des opérations, et les couleurs des arcs aux types des données échangées. Naturellement, une sortie ne peut être reliée qu'à des entrées de même type. Les arcs de précedence sont dessinés en gris pointillé.

3.5 Cycle procédural de l'agent

L'exécution d'un agent suit le cycle perception, délibération, action, comme illustré sur la figure 4 :

Perception. L'environnement communique à l'agent l'ensemble **P** de percepts. OPACK n'impose aucune contrainte sur la manière dont cet ensemble est construit et mis à jour par l'environnement. Pour gérer les modifications concurrentes, l'agent travaille sur une copie **P'** de **P**. L'agent ne modifie jamais **P** (seul l'environnement définit les percepts).

Délibération : exécution du flux (voir ci-après), qui modifie l'état de l'agent en agissant possiblement sur les ensembles **P'**, **A**, **C** et **K**.

Action : l'ensemble **A**, éventuellement modifié suite à l'exécution du flux, est utilisé par l'environnement qui est responsable de l'exécution des actions. OPACK n'impose aucune contrainte sur l'exécution des actions, sauf que chaque action terminée est marquée par l'environnement, pour permettre à l'agent d'en tenir compte dans son cycle procédural. Le succès ou l'échec d'une action est constaté par l'agent à travers les percepts (par exemple, j'essaie d'ouvrir la porte et je me rend compte qu'elle ne s'ouvre pas alors que l'action est terminée). L'agent peut choisir de supprimer de **A** les actions terminées mais l'environnement ne modifie jamais **A** de lui-même.

L'exécution d'un flux consiste à parcourir l'ensemble d'opérations dans l'ordre topologique du graphe orienté acyclique, défini par les deux types d'arcs, en commençant par les sommets sans prédecesseurs. Nous utilisons l'algorithme implémenté par Huang [20] dans la bibliothèque TASKFLOW³ qui utilise une approche multi-threadée pour effectuer en parallèle les opérations qui le peuvent. Nous avons ajouté à cette bibliothèque le transfert de données d'une opération à ses successeurs.

3.6 Intégration de modèles cognitifs

Un modèle cognitif peut impacter le modèle d'agent à deux niveaux :

1. sur l'état de l'agent (**P**, **A**, **C** et/ou **K**) en ajoutant des variables.
2. sur son ensemble d'opérations **O** en ajoutant des opérations ou en altérant le fonctionnement d'une opération.

Par exemple, si on ajoute un modèle cognitif du stress, il faut une opération pour gérer la nouvelle variable dans **C** et il faut altérer l'opération de sélection d'action pour tenir compte de l'état de stress lors du choix d'une action.

3. <https://taskflow.github.io/>

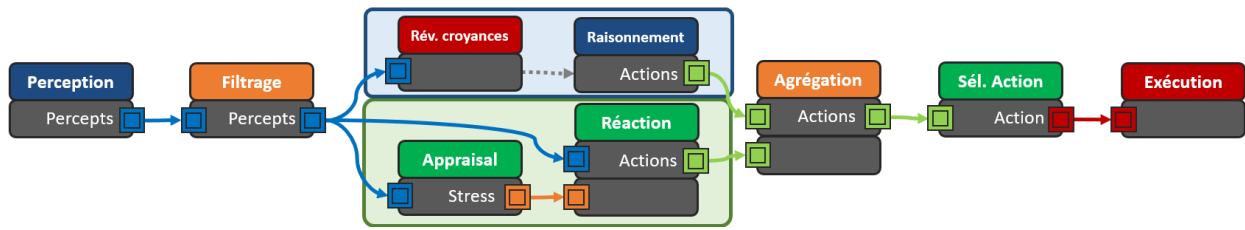


FIGURE 3 – Exemple d'un flux d'opérations.

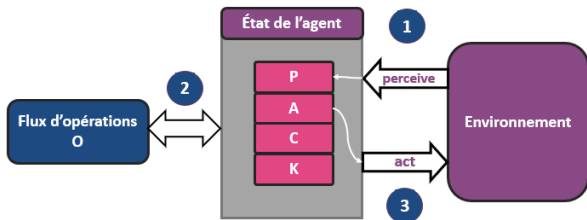


FIGURE 4 – Schématisation du cycle d'un agent.

Une fois que le modélisateur a ajouté les bonnes variables dans l'état de l'agent et, éventuellement, de nouvelles opérations dans O , il doit définir la manière dont le modèle cognitif altère les opérations existantes. Cela se fait au travers d'un ensemble de *fonction-impacts* dans notre modèle OPACK. Une *fonction-impact* est une fonction dont les entrées sont identiques à celles de l'opération qu'elle impacte. Elle peut aussi modifier les éléments de C , K ou A par effet de bord et, sauf dans le cas des opérations de rôle "Storage", elle calcule une sortie qui peut être de nature différente de celle de l'opération. Il faut ensuite combiner les sorties de ces *fonction-impacts* : pour cela, le modélisateur définit une *fonction-stratégie* qui combine les sorties des fonction-impacts pour calculer la sortie de l'opération.

Exemple Considérons le modèle cognitif de Driskell [13] qui indique, parmi d'autre, qu'une personne stressée est plus distraite, communique moins et réalise plutôt des actions maîtrisées. Intégrer ce modèle consiste à définir des fonction-impacts sur deux opérations : « *exécution* » (pour tenir compte d'un risque d'échec en cas de stress) et « *sélection d'action* ». Pour la sélection d'action, la *fonction-impact* pourrait, par exemple, renvoyer la liste des actions avec pour chacune d'elle un score calculé en fonction du stress défini dans l'état de l'agent. Ces scores peuvent être utilisés pour sélectionner l'action de l'agent.

Formalisation Soit une opération o et mc un modèle cognitif. On note I_o^{mc} l'ensemble des fonction-impact de mc sur o (éventuellement

vide). On définit $I_o = \cup_{mc} I_o^{mc}$ l'ensemble des fonction-impacts pour l'opération o . Toutes les fonction-impacts $f \in I_o$ ont les mêmes entrées que l'opération o et le même type de sortie, noté $\tau(I_o)$, qui peut être différent de type de sortie de l'opération o elle-même.

On définit une et une seule *fonction-stratégie* s_o par opération comme une fonction qui prend en entrée un ensemble, éventuellement vide, de valeurs dans $\tau(I_o)$ et qui calcule la sortie de o .

La contrainte que nous imposons dans notre méta-modèle est donc de définir une fonction-stratégie par opération et que celle-ci puisse travailler sur un ensemble quelconque, éventuellement vide, d'éléments de type $\tau(I_o)$. **C'est cette fonction-stratégie qui calcule le résultat de l'opération**, à partir des entrées de l'opération et des résultats des fonction-impacts. C'est le cœur de notre méta-modèle.

Notons qu'il est possible de définir plusieurs fonction-impact pour un mc donné et pour une même opération o . Par exemple, pour intégrer le modèle cognitif de Demary [14], le modélisateur peut définir une fonction-impact pour chacun des quatre quadrants de la figure 1.

Impact conditionnel Une fonction-impact définit *comment* un modèle cognitif altère une opération (et le comportement de l'agent). Mais il faut aussi définir *quand* cet impact a lieu en fonction du profil de l'agent et de son état (qui capture la situation). C'est ce que Faur [21] souligne dans son modèle PERSEED. Par exemple, un agent introverti se comporte de manière extravertie en présence d'amis, mais de manière introvertie en présence de collègues de travail. Il s'agit donc de déclencher des fonction-impacts différentes en fonction de la situation.

Nous définissons une fonction d'activation *act* qui prend en entrée l'état de l'agent et une fonction-impact f et qui retourne *true* ou *false*. Pour chaque opération o , lorsqu'elle est évaluée dans le flux, nous calculons l'ensemble d'im-

pacts $F_o = \{f \in I_o \text{ tq } act(f, PACK) = true\}$ des fonction-impacts activées en fonction de l'état de l'agent et nous en déduisons l'ensemble des résultats de ces fonctions-impacts qui sera passé à la fonction-stratégie. Cet ensemble peut évidemment être vide.

3.7 Exemple d'opérationnalisation

Considérons le flux de la figure 2 composé de deux opérations (actions envisageables et sélection d'action) qui correspond à une partie du modèle d'agent REPLICANTS. Nous allons montrer comment intégrer dans ce modèle les deux modèles cognitifs de Driskell et de Demary.

Nous définissons tout d'abord la fonction-stratégie pour ces deux opérations, ce qui nécessite de définir 1) le type $\tau(I_o)$ des fonction-impacts de l'opération et 2) le résultat obtenu par défaut lorsque l'ensemble d'impacts F_o est vide.

- Pour « actions envisageables », nous choisissons que $\tau(I_o)$ est, comme pour o , un ensemble d'actions. Par défaut, la stratégie renvoie le résultat du parcours de l'arbre défini dans le modèle REPLICANTS, stocké dans \mathbf{K} , c'est-à-dire un ensemble d'actions comprenant a minima l'action « ne rien faire ».
- Pour « sélection d'action », nous choisissons que $\tau(I_o)$ est un ensemble de couples (*action, score*). Par défaut, lorsque cet ensemble est vide, le système choisit une action aléatoirement dans la liste des actions fournies en entrée de l'opération.⁴

Nous souhaitons désormais impacter le comportement de l'agent par le biais de modèles cognitifs.

Ajout d'un impact : modèle cognitif de followership Commençons par un comportement simplifié de Demary [14] : un agent *passif* suit les ordres de son supérieur.

Nous supposons qu'à partir de \mathbf{C} , nous pouvons déterminer si un agent est *passif* ou non. Nous supposons qu'à partir de \mathbf{K} , nous pouvons déterminer si une action est *ordonnée* ou non par un supérieur.

Nous ajoutons une fonction-impact dans l'opération « sélection d'action ». Elle est active uniquement si l'agent est *passif*. Elle donne, pour

4. Il s'agit ici d'une vision simplifiée par rapport au modèle REPLICANTS. Celui-ci calcule en réalité un ordre total sur les actions en fonction des préconditions des actions.

chaque action, un score de 1 aux actions ordonnées par un supérieur et de -1 si elle a été interdite par un supérieur et de 0 sinon.

De même, nous pouvons ajouter des fonction-impacts pour chacun des 4 quadrants du modèle de Demary.

Ajout d'une opération Supposons désormais que nous souhaitons opérationnaliser un modèle de stress, comme celui de Lazarus [22], utilisé dans le modèle cognitif de Driskell. Ce modèle est basé sur un processus d'évaluation cognitive (*appraisal* en anglais) qui travaille sur un ensemble de variables d'évaluation dans l'environnement pour calculer une état émotionnel.

Nous ajoutons donc une variable de stress dans l'ensemble \mathbf{C} de l'état de l'agent et une opération « Manipulation » qui opère sur les perceptions de l'agent (qui correspondent aux variables d'évaluation cognitives) pour calculer la variable de stress. Il faut aussi ajouter un lien de précedence entre cette nouvelle opération et la sélection d'action.

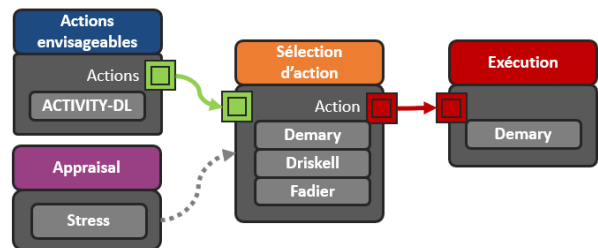


FIGURE 5 – Nouveau flux suite à l'intégration des modèles cognitifs

Ajout de plusieurs impacts : modèle cognitif de Driskell Pour refléter les impacts du stress dans les différentes opérations en nous appuyant sur les travaux de Driskell [13], nous définissons deux fonction-impacts, dépendantes du stress, au sein de l'opération « sélection d'action ». Elles attribuent chacune un score aux actions, telle que : « les actions coopératives ou communicatives ont un score -1 » et « les actions maîtrisées ont un score 1. » La fonction d'activation retourne vrai pour ces deux fonction-impacts uniquement lorsque l'agent est stressé.

Cet exemple simple montre comment il est possible d'intégrer plusieurs modèles cognitifs au sein d'un modèle d'agent dans le méta-modèle OPACK. Nous allons maintenant évaluer l'effet de ces modèles cognitifs sur la génération de comportement et leur intelligibilité.

4 Première évaluation

Pour évaluer l'impact des modèles cognitifs sur le comportement de l'agent, nous nous intéressons à un ensemble de mesures définies sur le comportement. Nous allons vérifier que chaque modèle cognitif modifie bien ces mesures comme il le devrait. Par exemple, si nous ajoutons un modèle cognitif qui réduit la communication d'une personne stressé, alors les mesures en lien avec les actes de communications doivent évoluer dans ce sens. Au contraire, les autres mesures ne devraient pas être impactées. Évidemment, il y aura des impacts relatifs puisqu'il s'agit de choisir des actions, donc de faire évoluer le comportement de l'agent. C'est ce que nous allons mesurer comme première étape de l'étude l'intelligibilité du modèle.

Nous nous limitons dans cette première étude à l'opération de sélection d'action et nous allons évaluer comment les modèles cognitifs impactent cette sélection. Nous utilisons 3 modèles cognitifs : Driskell, Demary et Fadier.

4.1 Protocole

Dans notre scénario, les agents représentant des secouristes doivent choisir parmi 8 actions qui ont les attributs suivants : *ordre* (est-ce que l'action est un ordre transmis par un supérieur), *com* (est-ce que l'action est une action de communication), *coop* (est-ce que l'action se fait à plusieurs ou aide les objectifs de l'équipe), *alu* (est-ce que l'action est tolérée par l'usage, voir section 2.2) et le niveau de qualification requis pour effectuer l'action (SC1, SC2 ou SC3). Nous avons 12,5% d'actions *ordre*, 37,5% d'actions *com*, 50% d'actions *coop*, 12,5% d'actions *ALU*. Nous avons 1 action requérant la qualification la plus haute (SC3) et 3 actions SC2 (les 4 restantes sont SC1).

Chaque agent va donc faire $2^8 - 1 = 255$ fois le cycle pour tirer l'ensemble des combinaisons possibles d'actions. En fonction de son état et des modèles cognitifs actifs ou non, il choisit une action (opération « sélection d'action »). Nous notons à chaque fois l'action choisie et les actions candidates.

Nous avons 24 états d'agent possibles (communicant ou non, pro-actif ou passif, stressé ou non, qualification de l'agent). Pour chaque modèle cognitif, nous pouvons distinguer les états dans lesquels la fonction d'activation renverrait *vrai* pour les fonction-impacts du modèle cog-

nitif considéré. Par exemple, la fonction d'activation renvoie *vrai* pour les fonction-impacts du modèle de Driskell lorsque la variable *stress* est vraie. Nous notons De , Dr et ALU l'ensemble de ces états pour les modèles de Demary, Driskell et Fadier respectivement. Réciproquement, nous notons \overline{De} , \overline{Dr} et \overline{ALU} l'ensemble des états qui n'activent pas les fonctions-impacts des modèles de Demary, Driskell et Fadier respectivement. Concrètement, le groupe d'agents De est le groupe des agents pour lesquels le comportement devrait être impacté quand le modèle de Demary est présent.

4.2 Mesures

Nous considérons les mesures suivantes et nous indiquons entre parenthèse le modèle cognitif qui devrait impacter la mesure :

ORD = pourcentage d'actions « ordre », (De),
COM = pourcentage d'actions « com » (Dr),
ALU = pourcentage d'actions « alu » (ALU),
COOP = pourcentage d'actions « coop » (Dr),
QUAL = pourcentage d'actions choisies dont le niveau de qualification est inférieur ou égal à celui de l'agent (Dr).

Concrètement, pour une combinaison de modèle cognitifs donnée, $ORD(Dr) = 60\%$ signifie que dans l'ensemble des exécutions où une action « ordre » était proposée dans la liste des actions candidates et où la fonction d'activation aurait renvoyé *vrai* suivant les conditions du modèle de Driskell, l'agent a choisi une action « ordre » dans 60% des cas.

4.3 Résultats

Lorsqu'aucun modèle cognitif n'est présent, quelque soit la mesure, il n'y a pas de différences entre les groupes (puisque aucune fonction-impact n'est activée), comme la figure 6 le reporte. C'est le comportement de référence.

Lorsque Demary est le seul modèle cognitif activé et qu'une action ordonnée est proposée, le groupe De choisit toujours cette action ($ORD(De) = 1$) alors que le groupe \overline{De} la choisira une fois sur 4 ($ORD(\overline{De}) = 0.2553$), comme dans le comportement de référence. Puisque, l'unique action ordonnée dans notre jeu de test est une action non-communicative, respectant la réglementation (non-ALU) et coopérative, nous observons une diminution de la

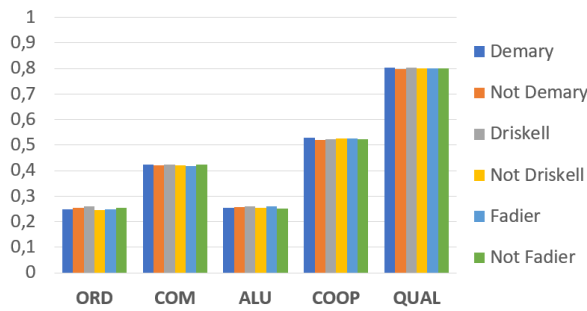


FIGURE 6 – Taux de sélection d'action lorsque aucun modèle n'est activé.

mesure *COM* et *ALU* et une augmentation de la mesure *COOP* pour le groupe actif. Les autres groupes ne sont pas impactés et restent sur le comportement de référence. Ces résultats sont présentés sur la figure 7).

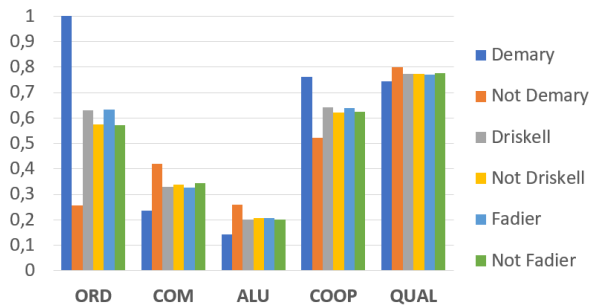


FIGURE 7 – Taux de sélection d'action lorsque seul le modèle de Demary est activé.

Nous obtenons le même résultat pour les autres modèles cognitifs lorsqu'ils sont activés seuls. Lorsque nous activons les trois modèles cognitifs, la tendance de chaque modèle cognitif se conserve (cf. figure 8). Par exemple, $ORD(De) = 0.51 > ORD(\overline{De}) = 0,11$, c'est-à-dire que le groupe actif de Demary continue à privilégier les ordres, contrairement au groupe non-actif. C'est le cas pour tous les modèles cognitifs. Par contre, nous observons des impacts sur les groupes a priori non-concernés par un modèle, en raison du fait que les actions peuvent correspondre à plusieurs critères.

5 Conclusions & Perspectives

Nous avons présenté les fondations d'un méta-modèle pour intégrer dans un modèle d'agent un ensemble de modèles cognitifs. Notre méta-modèle fait un minimum d'hypothèses sur le modèle d'agent et sur les modèles cognitifs considérés. Nous avons détaillé comment, grâce au

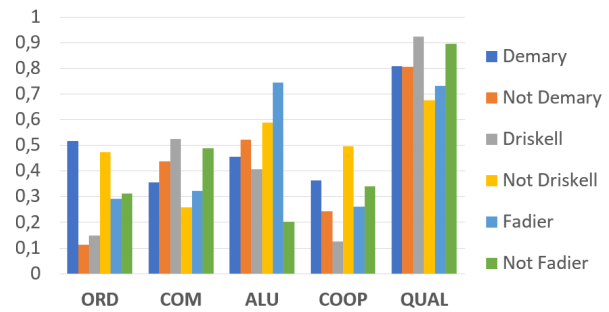


FIGURE 8 – Taux de sélection d'action lorsque tous les modèles sont activés.

flux d'opérations, le modélisateur peut définir et relier les opérations entre-elles pour modéliser le fonctionnement du modèle d'agent. Ensuite, nous avons décrit l'intégration de modèles cognitifs pour impacter cet ensemble d'opérations, grâce à la notion de fonction-impacts et de fonction-stratégies, qui permet de construire des combinaisons de modèles cognitifs non connus au préalable. Enfin, nous avons réalisé une première évaluation de cette approche pour vérifier que les comportements obtenus sont correctement impactés par les modèles cognitifs.

Nous avons montré que les modèles cognitifs ont un impact net sur la sélection d'action, qui est un élément important de la génération de comportements. Pour étendre notre évaluation et qualifier l'intelligibilité du modèle d'agent construit sur notre méta-modèle OPACK, il faudrait simuler le modèle d'agent complet et analyser les séquences d'actions produites. Nous pourrions alors demander à des experts du domaine de qualifier les comportements cognitifs observés (est-ce que l'agent était stressé, est-ce qu'il était communiquant), pour répondre à la question : « le changement de comportement produit par l'intégration ou non d'un modèle cognitif est intelligible » ?

Acknowledgment

Ces travaux sont financés par le projet DGA RAPID ORCHESTRAA.

Références

- [1] C. Barot, "Scénarisation d'environnements virtuels : vers un équilibre entre contrôle, cohérence et adaptabilité," These de doctorat, Compiègne, Feb. 2014.
- [2] L. Huguet, D. Lourdeaux, and N. Sabouret, "Présentation du projet VICTEAMS,"

- in *Workshop Affect Compagnon Artificiel Interaction (WACAI 2016)*, Brest, France, Jun. 2016.
- [3] B. G. Silverman, D. Pietrocola, B. Nye, N. Weyer, O. Osin, D. Johnson, and R. Weaver, "Rich socio-cognitive agents for immersive training environments : case of NonKin Village," *Autonomous Agents and Multi-Agent Systems*, vol. 24, no. 2, pp. 312–343, Mar. 2012.
- [4] P. G. Balaji and D. Srinivasan, "An Introduction to Multi-Agent Systems," in *Innovations in Multi-Agent Systems and Applications - 1*, ser. Studies in Computational Intelligence, D. Srinivasan and L. C. Jain, Eds. Berlin, Heidelberg : Springer, 2010, pp. 1–27.
- [5] M. Bourgais, "Vers des agents cognitifs, affectifs et sociaux dans la simulation," phd-thesis, Normandie Université, Nov. 2018.
- [6] A. Ortony, G. L. Clore, and A. Collins, *The Cognitive Structure of Emotions*. Cambridge : Cambridge University Press, 1988.
- [7] P. T. Costa Jr. and R. R. McCrae, "The Revised NEO Personality Inventory (NEO-PI-R)," in *The SAGE handbook of personality theory and assessment, Vol 2 : Personality measurement and testing*. Thousand Oaks, CA, US : Sage Publications, Inc, 2008, pp. 179–198.
- [8] J. Gratch and S. Marsella, "A domain-independent framework for modeling emotion," *Cognitive Systems Research*, vol. 5, no. 4, pp. 269–306, Dec. 2004.
- [9] C. A. Smith and R. S. Lazarus, "Emotion and adaptation," in *Handbook of personality : Theory and research*. New York, NY, US : The Guilford Press, 1990, pp. 609–637.
- [10] J. E. Laird, A. Newell, and P. S. Rosenbloom, "SOAR : An architecture for general intelligence." Stanford, Tech. Rep., 1986.
- [11] M. Ochs, N. Sabouret, and V. Corruble, "Simulation of the Dynamics of Nonplayer Characters' Emotions and Social Relations in Games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, no. 4, pp. 281–297, Dec. 2009.
- [12] T. Driskell, J. E. Driskell, and E. Salas, "Mitigating Stress Effects on Team Cohesion," in *Team Cohesion : Advances in Psychological Theory, Methods and Practice*, ser. Research on Managing Groups and Teams. Emerald Group Publishing Limited, Jan. 2015, vol. 17, pp. 247–270.
- [13] T. Driskell, E. Salas, and J. E. Driskell, "Teams in extreme environments : Alterations in team development and teamwork," *Human Resource Management Review*, vol. 28, no. 4, pp. 434–449, Dec. 2018.
- [14] G. Demary, "Évaluation cognitive du leader dans une dyade hiérarchique : des comportements non verbaux du suiveur aux comportements de leadership," *Bulletin de psychologie*, vol. Numéro 569, no. 5, pp. 271–275, Nov. 2020.
- [15] E. Fadier, C. De La Garza, and A. Didot, "Safe design and human activity : construction of a theoretical framework from an analysis of a printing sector," *Safety Science*, vol. 41, no. 9, pp. 759–789, Nov. 2003.
- [16] L. Hugué, D. Lourdeaux, and N. Sabouret, "Moteur de sélection de tâches pour des personnages virtuels autonomes non omniscients," in *Workshop affects, compagnons artificiels et interaction (WACAI 2018)*, Porquerolles, France, Jun. 2018.
- [17] K. Erol, J. Hendler, and D. Nau, "HTN Planning : Complexity and Expressivity," *Proceedings of the National Conference on Artificial Intelligence*, vol. 2, May 1994.
- [18] M. Bratman, *Intention, Plans, and Practical Reason*. Center for the Study of Language and Information, 1987.
- [19] J. P. Morrison, "Flow-based programming," in *Proc. 1st International Workshop on Software Engineering for Parallel and Distributed Systems*, 1994, pp. 25–29.
- [20] T.-W. Huang, D.-L. Lin, C.-X. Lin, and Y. Lin, "Taskflow : A Lightweight Parallel and Heterogeneous Task Graph Computing System," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 6, pp. 1303–1320, Jun. 2022.
- [21] C. Faur, "Approche computationnelle du regulatory focus pour des agents interactifs : un pas vers une personnalité artificielle," These de doctorat, Université Paris-Saclay (ComUE), Oct. 2016.
- [22] R. S. L. PhD and S. F. PhD, *Stress, Appraisal, and Coping*. Springer Publishing Company, Mar. 1984.

La modélisation Agent pour sensibiliser aux Solutions Fondées sur la Nature pour gérer les inondations

F. Taillandier^a
franck.taillandier@inrae.fr

A. Moatty^a
annabelle.moatty@inrae.fr

C. Curt^a
corinne.curt@inrae.fr

P. Di Maiolo^a
pascal.di maiolo@inrae.fr

B. Beullac^a
bruno.beullac@inrae.fr

P. Brueder^{a,b}
penelope.brueder@inrae.fr

^A UMR RECOVER (INRAE / Aix-Marseille Université) - 3275 route de Cézanne - CS 40061 – 13182 Aix-en-Provence Cedex 05

^b ADEME - 2 Bd de Gabès - 13008 Marseille

Résumé

De nombreux travaux ont démontré l'intérêt des Solutions fondées sur la Nature (SfN) pour réconcilier la prévention des risques et le développement durable. Cependant, leur mise en œuvre concrète soulève le problème de la réception sociale et de l'appropriation de ces stratégies par les différents acteurs. Le projet MANA vise à concevoir un jeu informatisé (Sim-MANA) permettant de sensibiliser les parties prenantes aux SfN pour gérer le risque inondation. Au cœur du jeu se positionne la simulation d'une inondation. Elle est réalisée à partir du modèle Agent MANA-Flo qui permet de simuler le déroulement d'une inondation en considérant le comportement des habitants (fuite, protection de ses biens, de sa voiture, diffusion d'information, etc.). Il simule ainsi la conjonction entre l'action humaine et la propagation de l'inondation, avec ses conséquences sur le système (par exemple, la fermeture des routes). Ce modèle permet de simuler différents scénarios d'aléas ainsi que différentes stratégies pour faire face à l'inondation. Il constitue en cela un support efficace au jeu Sim-MANA.

Mots-clés : *Modèle Agent, Jeu sérieux, Inondation, Solution fondée sur la Nature*

Abstract

Numerous studies have demonstrated the interest of Nature-based Solutions (NbS) in reconciling risk prevention and sustainable development. However, their concrete implementation raises the problem of the social reception and appropriation of these strategies by the various stakeholders. The MANA project aims to design a computer game (Sim-MANA) to raise stakeholders' awareness regarding NbS for flood risk management. At the center of the game is the simulation of a flood based on the Agent Based Model MANA-Flo, which simulates a flood event by considering the behavior of the inhabitants (escape, protection of property and cars, dissemination of information, etc.). It thus simulates the conjunction between human action and the propagation of the flood, with its consequences on the system (for example, the closure of roads). This model allows the simulation of different hazard scenarios and different strategies to cope with flooding. In this way, it is an effective support to Sim-MANA.

Keywords: *Agent-Based Model, Serious Game, Flood, Nature-based Solution*

1 Introduction

Les sociétés contemporaines sont confrontées à un large éventail de défis liés à la transition environnementale. Parmi ceux-ci se placent les enjeux liés à la gestion des risques naturels dans un contexte de changement climatique, induisant une évolution – et souvent une aggravation – des risques. Différentes stratégies structurelles et non structurelles sont envisageables pour faire face aux aléas naturels (inondations, incendies de forêts, avalanches, etc.). Les stratégies structurelles classiques sont fondées avant tout sur des actions de protection « dure », c'est-à-dire des éléments de protection anthropiques ayant pour objectif de protéger un/des enjeu(x) d'un aléa (digue, paravalanche, etc.). Cette vision « protection dure » (appelée aussi « grise ») pose un double problème. D'une part, ces structures donnent souvent une impression trompeuse de totale protection aux populations, ce qui a tendance à encourager le développement urbain et donc à accroître la quantité d'enjeux exposés. D'autre part, les systèmes de protection dure ont leurs inconvénients : impact environnemental, dégradation paysagère, coût de construction, d'entretien/maintenance, etc. Une solution alternative pour mettre en œuvre une gestion intégrée du risque est le recours aux Solutions fondées sur la Nature (SfN). Les SfN sont définies par l'Union Internationale de Conservation de la Nature comme « *des actions visant à protéger, gérer de manière durable et restaurer des écosystèmes naturels ou modifiés, pour relever les enjeux de société de manière efficace et adaptative tout en assurant le bien-être humain et des avantages pour la biodiversité* » [1].

Toutefois, les SfN sont encore peu implémentées ; il existe de nombreuses barrières à leur mise en œuvre, qu'elles soient techniques, réglementaires ou encore sociales [2]. Des études ont montré que de nombreux acteurs ne connaissent pas les SfN, voire y sont réticents [3]. Les raisons sont nombreuses et le débat est riche, mais ce n'est

pas le sujet de cet article. Ces blocages font émerger un enjeu de sensibilisation des différents acteurs locaux (élus, services techniques, riverains...).

Dans cet objectif, et dans le cadre du projet MANA (Modéliser pour augmenter l'acceptabilité des Solutions fondées sur la Nature), nous proposons une démarche, un modèle et un outil, permettant de nourrir une réflexion collective sur le recours aux SfN pour gérer les inondations, et plus largement aménager durablement un territoire. Pour cela, nous proposons de passer par une approche couplant simulation informatique, démarche participative et jeu sérieux.

2 Etat de l'art

2.1 Jeux sérieux

Les jeux sérieux peuvent avoir plusieurs objectifs tels que former ou sensibiliser. Les jeux sérieux, qui nous intéressent ici, visent à mettre les joueurs dans une situation de conflit entre des intérêts divergents qu'ils doivent gérer tout en étant soumis à des règles qui structurent leurs actions [4]. L'objectif est pédagogique, puisqu'il s'agit de générer des apprentissages auprès des joueurs [5]. À cet égard, les jeux de rôle présentent l'avantage majeur de créer un espace d'échanges et d'interactions entre les participants, sources d'apprentissages sociaux visant à faciliter la mise en œuvre d'actions collectives [6]. Les rôles peuvent être asymétriques (c'est-à-dire que tous les joueurs n'auront pas le même rôle ou le même pouvoir d'action) afin de reproduire des dynamiques sociales réelles. Cela permet aussi de confronter les joueurs à un rôle différent de celui qu'ils ont dans la vie réelle, pour leur faire appréhender le point de vue des autres acteurs [6].

Les jeux de rôle peuvent bénéficier d'un support informatique [6] permettant de simuler avec plus de réalisme certains phénomènes et d'automatiser certaines opérations (par exemple le comptage de

points, la remise à jour entre les tours, etc.). Les joueurs interagissent avec le modèle à travers un avatar (leur rôle) et un système de règles. Les effets d'apprentissage par le jeu reposent sur la succession de situations de réussites et d'échecs virtuels [7], qui incitent le joueur à s'améliorer [8].

2.2 Jeux sérieux pour la gestion des inondations

Plusieurs jeux sérieux et/ou modèles de simulation participative ont été proposés pour sensibiliser les acteurs aux enjeux liés aux inondations. Certains de ceux-ci sont des jeux de plateau (comme par exemple le jeu *Inondaction*) alors que d'autres bénéficient d'un support informatique. Ce sont ces derniers qui nous intéressent dans cet article. En particulier, nous avons fixé notre périmètre aux jeux permettant de : *i*) placer les joueurs dans une situation d'aménagement d'un territoire ; *ii*) simuler le déroulement d'une inondation et ses conséquences sur le territoire sinistré ; *iii*) intégrer des Solutions fondées sur la Nature (ou actions proches telles que des techniques issues de l'ingénierie écologique). Seuls deux jeux dans la littérature répondent à ces critères : LittoSIM [9] et SPRITE [10].

LittoSIM [9] est un modèle de simulation participative développé pour sensibiliser aux inondations côtières sur l'île d'Oléron. Ce dispositif vise à rendre accessibles les connaissances scientifiques pour les élus locaux et les agents communaux, et à leur permettre de faire le lien entre ces connaissances et les stratégies de gestion. SPRITE [10] est un jeu sérieux qui utilise aussi le cas de l'île d'Oléron comme territoire. Il vise à fournir un support pour l'enseignement de la gestion des risques à des étudiants de premier cycle. Ces deux modèles proposent de jouer le rôle du maire, dont l'objectif est d'améliorer la sécurité des habitants au travers de différentes actions, et notamment le recours à des SfN (comme par exemple la restauration du cordon dunaire),

tout en considérant aussi les dimensions économiques et environnementales, sur une période de plusieurs années. Bien que très intéressants ces jeux présentent des différences avec nos objectifs : *i*) ils visent un seul type de public particulier (élus et agents communaux pour LittoSIM et étudiants pour SPRITE) ; *ii*) ils s'intéressent à la submersion marine et non aux inondations par crues de rivières et ruissellement ; *iii*) le territoire étudié est celui de l'île d'Oléron dans son ensemble (SPRITE) ou de plusieurs commune de l'île (LittoSIM), alors que le territoire visé est dans notre cas plus restreint (commune/quartier) ; *iv*) ces jeux proposent de jouer uniquement des rôles de maires, alors que nous souhaitons pouvoir faire jouer différents acteurs locaux. Ainsi, il nous a fallu développer un jeu original, reposant sur son propre modèle pour simuler les inondations et en évaluer les conséquences sur le territoire en fonction des choix des joueurs. Comme pour LittoSIM et SPRITE, nous avons opté pour un système multi-agents (SMA).

2.3 Modèles multi-agents de simulation des inondations

De nombreux SMA ont été développés pour simuler des inondations. Ils peuvent être classés en 5 catégories [11] : gestion du risque d'inondation (GRI), simulation participative pour la gestion du risque d'inondation (SP), simulation d'évacuations (SEV), simulation d'événements d'inondations (SEI) et enjeux spécifiques sur le risque d'inondation (MES).

Une revue de littérature a révélé, que bien que nombreux, la plupart de ces SMA ont un objectif différent du nôtre. Nous avons évoqué les modèles de simulation participative (SP) dans la partie précédente. Les modèles spécifiques (MES) sont les plus éloignés de nos objectifs ; ils sont spécifiques à une problématique particulière : entreprises, personnes âgées, ou transmissions de maladies par exemple. Les modèles de gestion du risque d'inondation (GRI) proposent une perspective d'analyse moins fine que celle

souhaitée dans cette étude. Ils s'intéressent des territoires vastes (régions, pays) et ne simulent pas de façon fine l'inondation et le comportement des personnes. Cependant, ils offrent une réflexion intéressante sur les stratégies pour faire face aux inondations. Contrairement aux modèles d'évacuation (SEV), notre objectif n'est pas de se focaliser sur la gestion de crise et l'organisation d'une évacuation, mais plus généralement de simuler l'événement d'inondation. Il reste néanmoins important de considérer l'évacuation, qui a des conséquences majeures sur les bilans (humains principalement) des inondations. Les modèles de simulation d'inondation (SEI) permettent de rendre compte du déroulement d'une inondation et de son impact. Mais le cadre du jeu Sim-MANA impose plusieurs exigences: 1) simuler l'événement d'inondation, 2) intégrer les informations géographiques, 3) prendre en compte les comportements complexes des habitants, 4) pouvoir simuler différents scénarios d'aléas, 5) pouvoir intégrer différentes stratégies pour faire face aux inondations et notamment les SfN, 6) assurer la généralité et la flexibilité du modèle pour pouvoir le transposer d'un territoire à l'autre et 7) permettre une simulation sur un territoire en un temps court (égal ou inférieur à 5 minutes) pour pouvoir être utilisé dans le cadre d'un jeu sérieux. Aucun des modèles issus de la littérature ne permet de répondre à l'ensemble de ces exigences. Le modèle qui s'en rapproche le plus est SiFlo [11], même s'il n'est lui-même pas utilisable tel quel dans le cadre d'un jeu sérieux. Nous détaillons ce point ainsi que le modèle agent développé pour le jeu Sim-MANA dans la partie suivante.

3 Modèle

3.1 Objectif du modèle

Afin de sensibiliser et d'apporter des connaissances sur la gestion du risque d'inondation et sur les actions de prévention

et d'atténuation, nous avons conçu un SMA dédié à la gestion des inondations appelé MANA-Flo. Le modèle intègre, comme SPRITE [10] et LittoSIM [9], une double dynamique en simulant l'événement de crue (plusieurs heures) et la gestion de la crue (plusieurs années). Mais l'échelle simulée et le type d'inondation sont différents ; en effet, ces deux modèles ont simulé une submersion marine sur l'île d'Oléron, alors que MANA-Flo est dédié aux inondations (crues et ruissellement) à l'échelle d'une ville. La simulation de l'événement de crue dans MANA-Flo est donc plus fine que dans SPRITE ou LittoSIM. Le modèle s'inspire largement de SiFlo [11], qui est un SMA dédié à la simulation des inondations dans les zones urbaines. SiFlo prend en compte la propagation de l'eau et la réaction des habitants qui peuvent effectuer différentes actions : protéger leur maison, évacuer, partager des informations, etc. Cependant, SiFlo n'est pas un modèle de simulation participatif ; il est capable de simuler différents scénarios, mais chaque scénario doit être instancié dans le modèle via les paramètres du modèle. De plus, le temps de calcul de la simulation empêche de l'utiliser dans un contexte d'ateliers de jeu sérieux dans lequel le participant ne doit pas attendre une heure, ou plus, pour pouvoir accéder aux résultats de la simulation. MANA-Flo est une version simplifiée de SiFlo, destinée à faire partie d'un processus collaboratif incluant un jeu sérieux. En outre, SiFlo est un modèle stochastique dont les résultats ont un sens lorsque l'on considère un grand nombre de répétitions, ce qui n'est pas possible dans une approche interactive dans laquelle une seule simulation – par tour de jeu – sera effectuée. MANA-Flo réduit largement la stochasticité du modèle afin de pouvoir rendre compréhensible l'évolution de la situation en considérant une action sans le bruit des incertitudes.

Les sections suivantes exposent les principaux éléments du modèle.

3.2 Agents du modèle

MANA-Flo dispose de 10 types d'agents (figure 1) : *Grid*, *Building*, *Road*, *River*, *Dyke*, *Car*, *People*, *Stormwater network*, *Green area* et *World* (global agent).

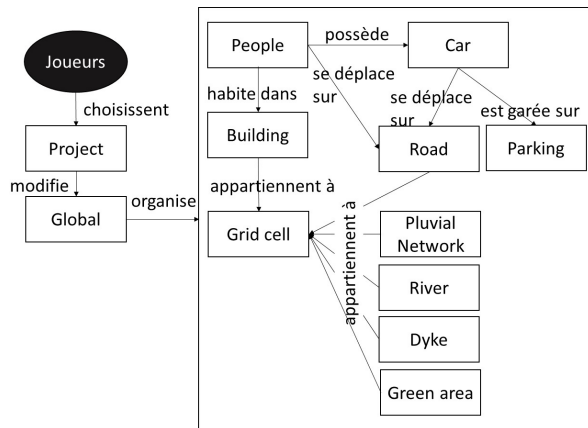


FIG. 1 – Agents de MANA-Flo

Grid. L'agent *Grid* est le support du monde virtuel et peut être vue comme le plateau de jeu. Il est composé d'un ensemble de parcelles qui correspondent chacune à un carré de 10m x 10m du territoire simulé. Chaque cellule a une altitude. La simulation de l'inondation utilise l'agent *Grid* comme support pour gérer les écoulements. Chaque parcelle est aussi caractérisée par une imperméabilité et une capacité d'évacuation qui dépend de la nature des sols et des infrastructures présentes.

River. L'agent *River* correspond au lit de la rivière. Il constitue une voie prioritaire pour l'écoulement de l'eau. Chaque segment de rivière est caractérisé par une largeur et une profondeur.

Building. L'agent *Building* couvre tous les types de bâtiments. Le modèle distingue les bâtiments résidentiels, les bâtiments commerciaux (magasins ou bureaux) et les bâtiments recevant du public (ERP). Chaque agent *People* habite dans un bâtiment et peut se déplacer d'un bâtiment à l'autre en utilisant les routes. L'eau peut pénétrer dans les bâtiments et causer des dommages. Les

bâtiments sont caractérisés par leur hauteur et nombre d'étages, et peuvent servir de refuge pendant l'inondation.

Dyke. L'agent *Dyke* regroupe tous les types d'éléments dont le but est de faire obstacle à l'eau. Il peut s'agir d'un barrage en remblai, d'un mur, etc. Les digues sont caractérisées par leur hauteur et leur résistance ; celles-ci peuvent en effet se briser lors d'une inondation, en cas de défaut d'entretien.

Road. L'agent *Road* correspond à tout élément routier (route, autoroute, chemin, etc.). Les routes constituent le réseau sur lequel les agents *People* se déplacent (à pied ou en voiture). Elles peuvent être inondées, ce qui limite la vitesse de déplacement, voire rend la route impraticable.

Pluvial network. L'agent *Pluvial network* correspond à tous les éléments du réseau permettant l'évacuation des eaux pluviales. Il peut être de type classique (génie civil) ou de type fossé/noue. Il est, comme les routes, géo-localisé et est en plus caractérisé par une capacité d'évacuation.

Green area. L'agent *Green area* correspond aux espaces naturels et/ou végétalisés. Ces espaces constituent des enjeux en termes de biodiversité et d'attractivité du territoire, mais peuvent aussi jouer positivement en cas d'inondation de par leur capacité d'absorption de l'eau (sols perméables).

Parking. L'agent *Parking* correspond à l'ensemble des zones dont la priorité est d'assurer le stationnement des voitures. Il s'agit donc d'espaces reliés aux routes où les agents *People* peuvent (ou pourraient) garer leur voiture. Ils sont caractérisés par un niveau de perméabilité qui dépend de leur nature (enrobé classique, parking enherbé, etc.).

Car. La voiture est le moyen de déplacement privilégié des agents *People*. Ceux-ci pourront l'utiliser pour se déplacer rapidement, mais elle constitue également un risque. Les agents *People* peuvent vouloir les récupérer et les mettre à l'abri en cas

d'inondation. L'eau peut immobiliser et endommager les agent *Car* ; les agents *People* voyageant en voiture dans des zones inondées sont particulièrement vulnérables.

People. L'agent *People* modélise un ménage. Leur comportement est inspiré de celui dans SiFlo, mais afin de simplifier le modèle et de réduire sa stochasticité, MANA-Flo n'utilise pas l'architecture BEN [12] de SiFlo. Au lieu de cela, MANA-Flo utilise un modèle réactif plus classique. Toutefois, le modèle prend en compte une large variété d'actions pouvant être réalisées par les habitants : évacuer (en voiture ou à pied) ; mettre à l'abri leur voiture ; accomplir leur tâche classique (hors inondation) dans ou hors de leur bâtiment ; ou exécuter un ensemble d'actions de protection au sein de leur maison telles que calfeutrer les ouvertures, protéger leurs biens, éteindre le gaz et l'électricité, se mettre à l'abri à l'étage et prévenir d'autres habitants.

Project. L'agent *Project* correspond aux actions possibles qui peuvent être mises en œuvre sur le territoire par les joueurs. La mise en œuvre d'un projet peut conduire à plusieurs sous-actions comme par exemple, créer de nouveaux bâtiments, ou bien des routes. MANA-Flo propose 19 projets différents, déclinés en trois niveaux (qui peuvent être par exemple différents emplacements pour le projet). Les projets sont classés du plus « gris » au plus « vert » et intègrent des actions telles que la construction d'un barrage écrêteur de crue, la construction d'un nouveau quartier, la mise en place de protections individuelles, la végétalisation des toitures, l'extension de la zone naturelle du plan local d'urbanisme (PLU), ou la création de bassin arboré de rétention. En plus de ces projets, le modèle intègre 6 actions de maintenance en lien avec les infrastructures, les bâtiments, la rivière et les parcs.

3.3 Dynamique du modèle

Le modèle MANA-Flo utilise une double dynamique : la première pour la phase de gestion, et la seconde pour le déroulement de

l'inondation (figure 2). La dynamique d'inondation utilise un pas de temps de 30s alors que celle de gestion utilise un pas de temps de 1 an. C'est dans cette deuxième dynamique que la composante interactive du modèle est intégrée, en proposant aux joueurs de choisir des projets à mettre en œuvre.

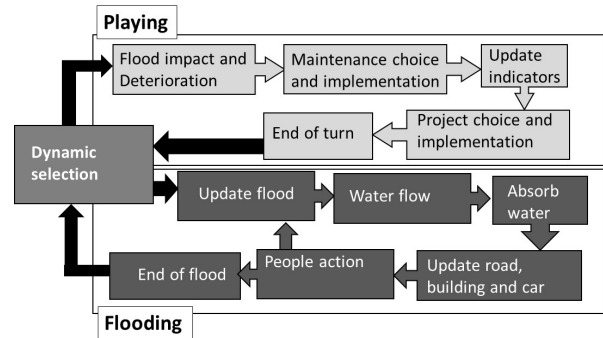


FIG. 2 – Dynamique de MANA-Flo

Processus de simulation d'inondation. La simulation d'une inondation dans MANA-Flo est proche de celle du modèle SiFlo. Tout d'abord, l'agent global met à jour les données environnementales (par exemple, l'eau en amont du bassin versant et la pluie) à partir des caractéristiques de l'aléa simulé. Le modèle intègre plusieurs scénarios d'aléas préconstruits (chacun étant composé de l'eau arrivant de l'amont, l'intensité et la durée de pluie et la durée de l'événement simulé) ; il est bien sûr possible d'en ajouter. De cette façon, on définit l'entrée d'eau dans le système et sa localisation. Ensuite, un sous-modèle d'écoulement est utilisé pour simuler le flux d'eau. C'est un modèle simple de déversement. Globalement, une parcelle va pouvoir déverser son eau sur les parcelles voisines si l'altitude d'eau sur cette parcelle est supérieure à l'altitude des parcelles voisines (en tenant compte des digues et rivières). On considère aussi la cinétique d'écoulement ; seule une certaine quantité d'eau peut s'écouler en un pas de temps d'une parcelle à l'autre. Le modèle prend ainsi en compte la capacité des digues à arrêter l'eau et leur état ; une digue en mauvais état peut se rompre sous l'action de l'eau. Le déversement d'eau se traduit par une quantité d'eau actualisée sur chaque parcelle. Une partie de

cette eau est absorbée par les sols (selon leur perméabilité) ou évacuée au travers du réseau d'eau pluviale. La quantité d'eau restante sur chaque parcelle est alors traduite en hauteur d'eau (en prenant en compte la présence de rivières).

Comme nous l'avons dit plus haut, à partir de ces hauteurs d'eau, MANA-Flo calcule l'état des routes (inondées, dangereuses, inutilisables). L'état de la route se dégrade avec l'eau induisant des dommages matériels, qui devront être réparés après l'inondation. Après la route, MANA-Flo calcule la quantité d'eau entrante dans les bâtiments, qui dépend de la hauteur d'eau et de la perméabilité du bâtiment. L'eau dans les bâtiments représente un danger pour les habitants et endommage les biens présents. Les habitants peuvent toutefois anticiper ce problème et s'en protéger en réalisant des actions de calfeutrement (diminution de la perméabilité), de coupure du gaz et de l'électricité et accès à l'étage (diminution du danger) ou en protégeant leur bien par le biais de batardeaux (diminution des dommages matériels). L'eau peut également s'écouler du bâtiment si le niveau d'eau à l'extérieur est inférieur au niveau à l'intérieur.

Ensuite, les agents *People* (habitants) vont agir. Pour cela, ils vont tout d'abord évaluer la situation à partir de la présence ou l'absence d'eau dans leur bâtiment, sur leur parcelle et les parcelles voisines. Ils peuvent aussi avoir ou non connaissance de l'inondation (information au préalable ou acquise durant la simulation). A partir de ces informations, les agents vont agir différemment en privilégiant certaines actions à d'autres. Afin de donner de la variabilité aux comportements des agents (par exemple, face à un danger important, un habitant peut décider de fuir alors qu'un autre va au contraire se réfugier chez lui), et cela sans induire de stochasticité, nous avons retenu un système basé sur un numéro attribué à chaque agent *People*. Par exemple, les agents qui ont un numéro multiple de 4 ont tendance à fuir en cas de fort danger. Ce système permet une distribution variée des

typologies de *People* sur le territoire. Bien sûr c'est un paramètre qui peut être redéfini selon l'objectif de la simulation.

Le processus de simulation d'inondation est reproduit de pas en pas jusqu'à ce que le temps de simulation correspondant au scénario d'aléa soit écoulé. Le processus s'arrête alors, et on bascule dans le processus de gestion, ou sur la fin de jeu si tous les tours ont été joués.

Processus de gestion. Ce processus commence par un bilan de l'inondation précédente (cela n'a bien sûr pas lieu au premier tour), qui présente notamment le décompte des éléments endommagés et les coûts de réparation. Ces informations sont exposées aux joueurs qui doivent alors faire des choix quant à la maintenance/réparation. Il est proposé pour chaque domaine (bâtiments, routes, etc.) quatre solutions possibles : ne rien faire, réparer/maintenir au minimum (l'état ne revient pas à son niveau initial), revenir à l'état d'avant inondation, ou aller au-delà (améliorer les éléments). Bien sûr le coût n'est pas le même selon le choix fait. A partir de ces choix, des conséquences de l'inondation, et des projets réalisés, 9 indicateurs sont évalués. Ils sont regroupés en 3 domaines : la sécurité, l'attractivité du territoire et la soutenabilité (tels que les enjeux de développement durable). Chaque indicateur est évalué sur une échelle à 6 niveaux (note de 0 à 5 ; 0 étant la pire, 5 la meilleure, et 2 correspondant au *statu quo*).

L'étape suivante est le choix des projets à réaliser à ce tour. C'est là que les joueurs interviennent et choisissent d'implémenter des projets. Leur choix sont contraints par le budget disponible. Ils vont devoir négocier pour retenir les projets à implémenter. Cette étape de négociation est hors du modèle et se déroulera dans l'espace de jeu. Ce qui est intégré au modèle est simplement le choix du/des projet(s) à réaliser associé au niveau. L'implémentation d'un projet est réalisée instantanément dans la simulation ; on fait l'hypothèse que le pas de temps d'une année permet de mettre en œuvre la mesure prise.

Cette implémentation peut avoir plusieurs conséquences selon le type de projet : cela peut conduire à modifier un attribut d'un ou plusieurs agents (par exemple, les toitures végétalisées apportent une certaine capacité d'absorption de l'eau aux bâtiments) ou à créer de nouveaux agents (comme de nouveaux bâtiments et des routes pour l'aménagement d'un nouveau quartier). Cela peut aussi conduire à supprimer des agents. Par exemple un bassin de rétention peut prendre la place de bâtiments existants obligeant à les supprimer et à reloger les occupants.

Une fois que les joueurs sont satisfaits de leur choix de projet ou qu'ils n'ont plus de budget pour effectuer de nouvelles actions, on peut mettre fin au tour de jeu. On repart alors sur une phase d'inondation.

3.4 Entrées et sorties du modèle

Le modèle utilise en entrées des données géographiques ; celles sont directement lues depuis des fichiers SIG (*shapefile* et données topographique sous format *asc* ou *geotiff*). Les bâtiments, les routes, les digues, les cours d'eau et la topographie sont directement importés depuis les fichiers dédiés. Si, théoriquement, le modèle peut être appliqué à n'importe quel territoire, dans le cadre de la simulation interactive, il a été conçu pour être appliqué à un territoire virtuel spécifique. Ce territoire appelé La Vita rassemble de nombreuses caractéristiques des territoires urbains du Sud-Est de la France et en particulier de La Ciotat et Vitrolles. Les agents *People* sont créés au début de la simulation en fonction des bâtiments ; en début de jeu, on a près de 9000 agents *People*, soient environ 19000 habitants (les agents *People* représentant des foyers). En plus de ces données, le modèle nécessite des informations sur les projets, dont des données géographiques pour les projets qui en nécessitent (e.g. bassin de rétention, barrage...).

En sortie, MANA-Flo fournit deux principaux

éléments (figure 3) : une carte de la ville et des indicateurs. La carte de la ville permet de rendre compte de la réalité de la ville (topographie, positionnement des enjeux, etc.) et permet aux joueurs de suivre le processus d'inondation (montée du niveau de l'eau sur les parcelles, détérioration des bâtiments et des infrastructures, activités des agents *People*, etc.). Les indicateurs permettent de réaliser un bilan de l'action des joueurs et d'évaluer la situation du territoire. Comme nous l'avons indiqué, nous utilisons 9 indicateurs réparti dans 3 catégories. Ces indicateurs sont : logement, infrastructure, économie, satisfaction de la population, environnement, aménagement des sols, bilan humain, bilan matériel parc privé et bilan matériel parc public. Les trois derniers sont bien sûr relatif aux conséquences de l'inondation. Ces indicateurs et leur évaluation sont fondés sur la littérature, mais nous ne les détaillerons pas plus ici.

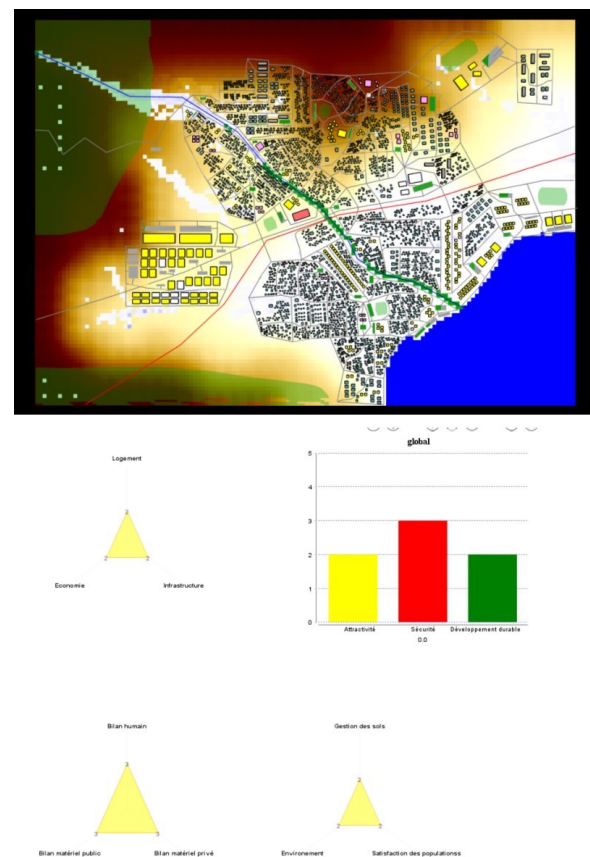


FIG. 3 – Interface de MANA-Flo

3.5 Implémentation

MANA-Flo a été implémenté sur la plateforme GAMA [13]. GAMA est une plateforme de SMA open source qui offre de nombreuses fonctionnalités particulièrement intéressantes pour MANA-Flo. GAMA permet une intégration native des données SIG et fournit des « *skill* » pour gérer les déplacements des agents. De plus, GAMA propose différentes fonctionnalités pour intégrer l'interactivité dans la simulation.

4 Du Modèle au jeu

4.1 Principes

Dans Sim-MANA, les joueurs incarnent l'un des cinq rôles suivants : Equipe municipale, Services techniques, Syndicat de l'environnement, Représentant des habitants et Représentant des commerçants et chefs d'entreprises. Chaque rôle dispose d'objectifs, de capacités d'actions et d'un budget qui lui sont propres. Chaque tour de jeu débute par la distribution du budget à chaque joueur / équipe. Ce budget peut être intégralement dépensé pendant le tour de jeu, mais aussi économisé d'un tour sur l'autre. Ce budget permet à chaque joueur / équipe de mener à bien des projets d'aménagement qui peuvent être portés par un seul rôle, soit être financés à plusieurs. A chaque tour (qui représente chacun une année), les joueurs prennent connaissance de la situation au travers de la simulation et notamment des indicateurs). Ils choisissent des projets qu'ils souhaitent porter parmi le panel de projets qui leur est proposé et en fonction de leur rôle. Ces projets peuvent être des actions de construction (logements, commerces et entreprises, établissements recevant du public, etc.), d'aménagement d'espaces verts, d'entretien et de maintenance mais aussi de mise en œuvre de mesures de prévention et de protection vis-à-vis des crues du Vitello et du ruissellement (SfN, réponses hybrides – telles que les revêtements de sols perméables par exemple,

et « solutions grises » comme un barrage écrêteur de crues). Les joueurs / équipes peuvent aussi investir dans l'amélioration du confort des bâtiments. Chaque projet choisi par les équipes est alors présenté et défendu devant les autres joueurs au sein du « Conseil Municipal ». Les projets financés sont mis en œuvre et implémentés dans la simulation. On simule ensuite un événement d'inondation dont l'intensité varie de la crue décennale à centennale. Les indicateurs sont recalculés après l'événement afin de donner des renseignements aux joueurs vis-à-vis de l'atteinte de leurs objectifs.

4.2 Séance de jeu

Le jeu est prévu pour être joué par au minimum cinq personnes et au maximum 15 personnes, soit trois personnes par équipe / rôle. Le jeu doit s'intégrer dans une séance d'environ deux heures et demi, composée de trois temps : la présentation du jeu et l'énoncé des règles, le jeu lui-même et le débriefing. Le débriefing est un moment réflexif qui permet de confronter l'expérience de jeu à la réalité [7]. Le débriefing peut prendre plusieurs formes, qu'il soit oral et animé par un facilitateur (le plus souvent un scientifique ayant participé au développement du dispositif et qui joue le rôle d'animateur d'atelier, comme c'est le cas dans LittoSIM), ou écrit et réalisé individuellement par les participants, ce qui peut constituer un temps de réflexion individuelle avant la mise en commun des idées [6]. Nous proposons de coupler les deux en associant un temps individuel de réponse à un questionnaire, suivi par un temps collectif d'échange sur les événements marquants [14] et les émotions suscitées par le jeu.

5 Conclusion

Nous avons présenté dans cet article le modèle MANA-Flo qui s'intègre dans le jeu Sim-MANA qui vise à sensibiliser les acteurs d'un territoire aux solutions fondées sur la nature pour gérer le risque d'inondation. Le

modèle et le dispositif sont en place, du moins dans une version préliminaire, mais les phases de tests n'ont pas encore commencé. Un ensemble de tests et expérimentation est ainsi prévu dans le but d'améliorer le modèle et le jeu ainsi que pour valider l'approche retenue. Pour cela, nous concevons présentement un dispositif d'évaluation au travers de différents questionnaires et un plan d'expérience. Nous espérons que les résultats confirmeront l'intérêt des jeux sérieux et du couplage entre jeu et SMA pour la sensibilisation aux SfN.

Remerciements

Ce travail a été réalisé dans le cadre du projet MANA, financé par la Fondation MAIF.

Références

- [1] UICN, *Définition des solutions fondées sur la nature*. 2016. [En ligne]. Disponible sur: <https://portals.iucn.org/congress/fr/motion/077>
- [2] G. Lupp *et al.*, « Stakeholder Perceptions of Nature-Based Solutions and Their Collaborative Co-Design and Implementation Processes in Rural Mountain Areas—A Case Study From PHUSICOS », *Frontiers in Environmental Science*, vol. 9, 2021, Consulté le: 9 mars 2022. [En ligne]. Disponible sur: <https://www.frontiersin.org/article/10.3389/fen vs.2021.678446>
- [3] C. C. Anderson et F. G. Renaud, « A review of public acceptance of nature-based solutions: The 'why', 'when', and 'how' of success for disaster risk reduction measures », *Ambio*, vol. 50, n° 8, p. 1552-1573, août 2021, doi: 10.1007/s13280-021-01502-4.
- [4] L. Sauvé et D. Kaufman, Éd., *Jeux et simulations en éducatifs: études de cas et leçons apprises*. Québec, Québec: Presses de l'Université du Québec, 2010.
- [5] A. Monfort, M. Amalric, et N. Becu, « Approche et méthode d'évaluation de la participation dans un dispositif de simulation participative : le cas de LittoSIM », présenté à Sixièmes journées doctorales du GIS Démocratie & Participation sur les expérimentations démocratiques, la participation du public et la démocratie participative, 2020. Consulté le: 9 mars 2022. [En ligne]. Disponible sur: <https://www.participation-et-democratie.fr/approche-et-methode-d-evaluation-de-la-participation-dans-un-dispositif-de-simulation-participative>
- [6] N. Becu, « Les courants d'influence et la pratique de la simulation participative : contours, design et contributions aux changements sociétaux et organisationnels dans les territoires », Habilitation à diriger des recherches, La Rochelle Université, 2020. Consulté le: 19 janvier 2022. [En ligne]. Disponible sur: <https://hal.archives-ouvertes.fr/tel-02515352>
- [7] M. Amalric *et al.*, « Sensibiliser au risque de submersion marine par le jeu ou faut-il qu'un jeu soit spatialement réaliste pour être efficace ? », *sdj*, n° 8, déc. 2017, doi: 10.4000/sdj.859.
- [8] D. A. Kolb, *Experiential learning: experience as the source of learning and development*. Englewood Cliffs, N.J: Prentice-Hall, 1984.
- [9] N. Becu *et al.*, « Participatory simulation to foster social learning on coastal flooding prevention », *Environmental Modelling and Software*, vol. 98, n° Supplement C, p. 1-11, 2017, doi: 10.1016/j.envsoft.2017.09.003.
- [10] F. Taillandier et C. Adam, « Games Ready to Use: A Serious Game for Teaching Natural Risk Management », *Simulation & Gaming*, vol. 49, n° 4, p. 441-470, août 2018, doi: 10.1177/1046878118770217.
- [11] F. Taillandier, P. Di Maiolo, P. Taillandier, C. Jacquenod, L. Rauscher-Lauranceau, et R. Mehdizadeh, « An agent-based model to simulate inhabitants' behavior during a flood event », *International Journal of Disaster Risk Reduction*, vol. 64, p. 102503, oct. 2021, doi: 10.1016/j.ijdr.2021.102503.
- [12] M. Bourgeois, P. Taillandier, et L. Vercouter, « BEN: An Architecture for the Behavior of Social Agents », *Journal of Artificial Societies and Social Simulation*, vol. 23, n° 4, p. 12, 2020, doi: 10.18564/jasss.4437.
- [13] P. Taillandier *et al.*, « Building, composing and experimenting complex spatial models with the GAMA platform », *GeoInformatica*, vol. 23, n° 2, p. 299-322, avr. 2019, doi: 10.1007/s10707-018-00339-6.
- [14] R. M. Fanning et D. M. Gaba, « The Role of Debriefing in Simulation-Based Learning », *Simulation in Healthcare: The Journal of the Society for Simulation in Healthcare*, vol. 2, n° 2, p. 115-125, 2007, doi: 10.1097/SIH.0b013e3180315539.

Session 6 : Démonstrations et posters

Modélisation et simulation multi-agents pour la gestion des tâches entre véhicules et infrastructures dans le cadre de la ville intelligente

Emmanuelle Grislin-Le Strugeon^{a,b}
emmanuelle.grislin@uphf.fr

Hamza Ouarnoughi^{a,b}
hamza.ouarnoughi@uphf.fr

^aUniversité Polytechnique Hauts-de-France, LAMIH, CNRS, UMR 8201, Valenciennes F-59313, France

^bINSA Hauts-de-France, Valenciennes F-59313, France

Résumé

Les données collectées et les traitements à réaliser ne font qu'augmenter au sein des véhicules visant à une certaine autonomie, alors que les capacités des systèmes embarqués sont limitées. Les infrastructures des villes intelligentes telles que les Road Side Units (RSU) peuvent fournir des ressources supplémentaires aux véhicules. Se posent alors des questions de répartition des tâches et de gestion des interactions entre véhicules et RSUs, ainsi qu'entre RSUs. Ces problématiques concernent des éléments distribués et connectés, dont certains présentent une forme d'autonomie, avec une composante dynamique et des informations partielles. Ces caractéristiques en font un cas d'étude et d'application naturel pour les systèmes multi-agents. Afin d'évaluer différents modes d'interaction entre véhicules et RSUs, nous proposons de modéliser et simuler ceux-ci dans le cadre d'une architecture multi-agent.

Mots-clés : Modélisation et simulation agent, Interaction, Véhicule autonome, Ville intelligente

Abstract

The volume of collected data and the amount of related processes are constantly rising in the context of autonomous vehicles, but the embedded system capacities are still limited. The intelligent city infrastructures such as the Road Side Units (RSU) can provide additional resources to the vehicles. Questions arise then, about task allocation and interaction management, between vehicles and RSU or among the RSU. These questions concern distributed and connected entities, with aspects related to autonomy, dynamics and partial information. Because of these characteristics, we propose to use a multiagent architecture to model, simulate and evaluate different interaction modes between vehicles and RSU.

Keywords: Agent-Based Modeling and Simu-

lation, Interaction, Autonomous vehicle, Smart city

1 Introduction

La mobilité dans les villes du futur reposera en partie sur des véhicules autonomes. La tendance actuelle montre que ces véhicules nécessitent de plus en plus de capacités de stockage et de traitement de données. Même si la miniaturisation continue de progresser, les capacités des systèmes embarqués restent limitées par les contraintes matérielles.

Par ailleurs, les infrastructures des villes dites intelligentes peuvent fournir des ressources supplémentaires aux véhicules. En effet, une caractéristique importante de ces villes, est l'omniprésence de services accessibles via des réseaux sans fil. Ces services sont conçus de façon à accompagner les utilisateurs dans leur mobilité, relayés et supportés par des infrastructures de proximité. De telles infrastructures sont constituées d'unités en réseau, situées à proximité des sources de données et des utilisateurs, et en limite du "nuage", participant ainsi à ce qui est généralement dénommé le *Mobile Edge Computing* (MEC) [6]. Un exemple d'unités appartenant à des infrastructures de ce type est formé par les Road Side Units (RSU). Les RSU ont la particularité d'être à même de fournir un ensemble de services à la demande dans des délais raisonnables du fait de leur proximité.

L'idée est d'utiliser ces services afin permettre aux véhicules de se décharger de certaines tâches de stockage et de traitement. La problématique est alors de permettre aux véhicules d'interagir avec les RSU dans ce but. Cette problématique met en jeu des éléments distribués et connectés, dont certains montrant une forme d'autonomie, une composante dynamique, et un accès

partiel à l'information. Ces caractéristiques en font un cas d'étude et d'application naturel pour les systèmes multi-agents. Nous proposons ainsi de faire supporter les interactions des véhicules et RSU par un SMA.

Afin de montrer l'intérêt d'une telle approche, ainsi que de permettre de tester différentes architectures et politiques d'échanges possibles, une étape préalable est de fournir une simulation du futur système multi-agents. Cette étape est décrite dans ce papier, dont l'objectif est ainsi de présenter la modélisation et la simulation à base d'agents réalisées. Le focus est ainsi posé ici sur la partie modèles et outil de simulation, plutôt que sur la proposition d'architectures ou de politiques adaptées.

La section 2 présente un ensemble d'études en réponse à la problématique. La section 3 présente la conception multi-agent des interactions entre véhicules et infrastructure intelligente. La section 4 décrit la modélisation multi-agent qui a été réalisée. La section finale conclut et propose un ensemble de perspectives à cette étude.

2 Modélisation et simulation des décharges de traitement entre véhicules et nuage

Depuis une quinzaine d'années, des simulateurs tels que SUMO[5] et NS-3[8], permettent d'évaluer des approches de communication entre véhicules ou entre véhicules et infrastructures, selon des critères liés aux problématiques réseau comme le routage et la sécurité. Concernant le déchargement de traitements entre edge, fog et cloud, les simulateurs existants ont généralement pour objectif d'évaluer des politiques d'optimisation de placement de tâches, selon des critères tels que la consommation d'énergie et la latence. Les plus utilisés dans ce cadre sont iFogSim [3] pour les interactions IoT-Edge ou Fog, et CloudSim [1] pour les traitements dans le cloud.

Des architectures basées agent ont été proposées en réponse à la problématique du computing offloading. Par exemple dans [10], l'architecture repose sur un agent logiciel attaché à chaque véhicule, qui prend les décisions relatives aux décharges de tâches et réalise les communications avec des serveurs distants à l'aide d'un drone supervisant une zone du trafic. L'architecture proposée par Zhang et al. [12] est basée sur des agents en essaim qui gèrent la communication de données entre les niveaux véhicules, "sub-cloud" et cloud. Dans d'autres études [9],[4], les agents

gèrent l'équilibre de charge dans le contexte du cloud computing. Nous envisageons à terme une simulation assez détaillée pour mettre en évidence les avantages et inconvénients de différentes politiques de déchargement de tâches.

3 Conception multi-agent des interactions entre véhicules et infrastructure intelligente

3.1 Intérêt du système multi-agent

Comme indiqué précédemment, l'intérêt d'une architecture multi-agent apparaît en raison des caractéristiques de distribution, interaction et autonomie attachées à la problématique du système intégrant véhicules et RSUs.

Côté véhicules, des décisions sont à réaliser concernant la pertinence de se décharger de données et de tâches. Côté RSUs, des décisions sont à réaliser concernant l'allocation des tâches et les échanges d'information. Les décisions prises dépendent de différents paramètres, des paramètres aux valeurs fixées tels que les capacités de stockage et de traitement de chaque élément du système, et des paramètres dont la valeur change en fonction du contexte. Le contexte est en effet dynamique, avec des tâches variables générées par la situation de l'environnement, et des possibilités variables en fonction de l'état des autres éléments du système.

Par ailleurs, chaque véhicule et chaque RSU n'a accès qu'à une vue partielle de l'environnement et du système. Chaque véhicule perçoit par un ensemble de capteurs (caméras, radars) une partie limitée de son environnement constitué du réseau routier et de certaines signalisations, d'autres véhicules, de certains piétons. Chaque RSU perçoit les véhicules qui sont dans sa zone de perception du trafic, les autres RSUs accessibles par le réseau et éventuellement d'autres éléments distants appartenant au nuage.

3.2 Objectifs de la simulation

L'objectif de la simulation est de montrer les possibilités offertes par une architecture de système multi-agent en termes de prise de décision et d'allocation de tâches en dynamique dans le cadre d'un réseau comprenant des véhicules et des RSUs.

La simulation a ainsi pour but premier de montrer comment un système multi-agents peut

prendre en charge 1) les prises de décision et l'adaptation en dynamique de la gestion de certaines tâches de traitement de données au sein d'un véhicule autonome et au sein d'un RSU ; 2) la gestion des interactions entre RSUs d'une part, entre véhicules et RSUs d'autre part. De plus, la simulation doit permettre de tester différentes solutions de décharge des tâches et d'obtenir des mesures pertinentes pour la comparaison de ces solutions.

Il a été décidé de simuler les aspects suivants : (i) aspects liés à la génération des données et tâches (déplacement des piétons et véhicules, volumes de données collectées et tâches à réaliser par les véhicules autonomes), (ii) aspects liés aux communications (délais de communication, échanges de messages entre véhicules et RSUs, échanges de messages entre RSUs), (iii) aspects liés au traitement des tâches (délais de réalisation, retours de résultats).

Le comportement détaillé des véhicules ne sera pas simulé, en particulier nous ne cherchons pas à représenter les tâches "réelles" des véhicules mais plutôt certaines de leurs caractéristiques telles que leurs durées, les capacités de traitement nécessaires, les volumes de données nécessaires et les délais limites acceptables.

4 Une modélisation dirigée par les objectifs de simulation

La simulation réalisée de l'architecture visée est elle-même à base d'agents, à l'aide du logiciel NetLogo [11] choisi pour ses facilités de modélisation rapide pour une version de démonstration. Le modèle créé est ici décrit selon ses différents aspects, structurels, dynamiques et organisationnels, communs aux approches de modélisation et simulation basés agent [2].

4.1 Aspects structurels

L'environnement statique de la simulation est formé par le réseau routier, les feux de circulation, les passages piétons et les vitesses limites des axes routiers. Le modèle créé reprend et étend le modèle de simulation de trafic proposé par Jiri Lukas ¹.

Dans l'outil Netlogo, hormis l'environnement en arrière-plan, quasiment tout est défini comme étant un agent. Nous pouvons cependant distinguer deux catégories d'agents dans le système

1. Modèle "Town-Traffic & Crowd simulation", nov. 2014, visible à : <http://ccl.northwestern.edu/netlogo/models/community>

simulé, selon qu'ils communiquent directement ou pas avec les autres agents. Ainsi les agents qui simulent le comportement des piétons et des véhicules non autonomes interagissent sans communication directe avec les autres agents. Nous pouvons les considérer comme des parties de l'environnement dans notre contexte. Au contraire des précédents, les agents qui simulent les véhicules autonomes et ceux qui simulent les RSUs communiquent directement les uns avec les autres. Ce sont des agents logiciels dans la simulation et ils représentent des agents qui feront également partie de l'architecture réelle visée.

4.2 Aspects dynamiques

Les véhicules et les RSUs peuvent échanger des informations lorsqu'ils se trouvent dans les zones d'accessibilité réseau les uns des autres. Chaque agent-véhicule autonome doit prendre des décisions sur la pertinence de se décharger de certaines tâches. Chaque agent RSU doit prendre des décisions sur l'acceptation et l'ordonnement des tâches demandées en fonction de ses ressources. Des diagrammes de comportement agents issus d'une version précédente du simulateur sont détaillés dans [7], dans le cadre d'une politique spécifique de gestion des requêtes côté RSU qui privilégie les agents les moins servis.

4.3 Aspects organisationnels

Nous avons choisi de modéliser une première organisation très proche du réseau limité aux seuls véhicules et RSUs, sans considérer d'éléments de plus haut niveau tels que des composants plus distants appartenant au nuage qui pourraient interagir avec les RSUs. Dans cette organisation, chaque véhicule ou RSU ne dispose pas d'information sur la distribution de charge parmi les RSUs. Chaque RSU gère un groupe formé des véhicules appartenant à sa zone d'accessibilité par le réseau. L'appartenance au groupe varie au fil des déplacements des véhicules, les groupes peuvent être disjoints ou non, selon les positions spatiales des RSUs et leur zone d'accessibilité.

4.4 Paramétrage de la simulation

Le simulateur offre un ensemble de paramétrages du trafic routier, des véhicules et des RSUs (voir figure 1). Aux paramètres liés au modèle de base du trafic, nous avons ajouté le paramétrage de la proportion de véhicules autonomes parmi l'ensemble des véhicules et de leur capacité de traitement (identique pour tous). Les paramètres des

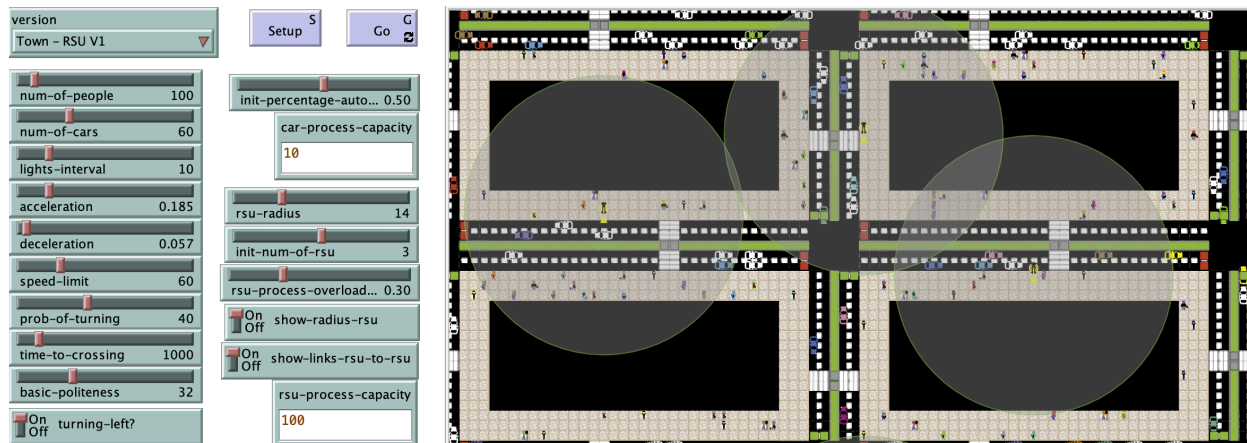


FIGURE 1 – Interface de la simulation

RSUs concernent leur nombre, leur positionnement spatial, leur rayon d'action et leur capacité de traitement. Concernant la politique d'allocation de tâches utilisée par les RSUs (identique pour tous), le choix se fait parmi : 1) pas de décharge de la part des véhicules, 2) le RSU traite les demandes selon l'ordre chronologique de réception et sa décision dépend uniquement de sa capacité de traitement à l'instant du traitement de la demande, 3) le RSU trie les demandes reçues afin de traiter en priorités les véhicules les moins bien servis.

5 Conclusion

Nous avons proposé une modélisation multi-agent et une simulation à base d'agents dans un objectif de démonstration des possibilités offertes par des architectures multi-agents de décharge de tâches entre véhicules et infrastructure dans le cadre des villes intelligentes. Une première version de la simulation permet de mettre en évidence certaines conséquences des modèles implantés. Des premiers résultats, obtenus avec une politique simple de partage d'historiques entre RSUs, sont décrits dans l'article [7].

L'organisation des agents est peu développée dans ce premier modèle, nous envisageons de permettre la construction d'autres types d'organisations, et d'offrir des possibilités de reconfiguration suite à la panne d'un des RSUs. Enfin, une réflexion reste à mener sur les limites précises du simulateur et donc des résultats obtenus.

Références

- [1] Rodrigo N. Calheiros, Rajiv Ranjan, César A. F. De Rose, and Rajkumar Buyya. Cloudsim : A novel framework for modeling and simulation of cloud computing infrastructures and services. *CoRR*, abs/0903.2525, 2009.
- [2] Massimo Cossentino, Carmelo Lodato, Patrizia Ribino, and Valeria Seidita. A heuristic for problem formalization in agent based simulation studies. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *FedCSIS 2015*, volume 5 of *Annals of Computer Science and Information Systems*, pages 1733–1743, Łódź, Poland, Sept. 2015. IEEE.
- [3] H. Gupta, A.V. Dastjerdi, S.K. Ghosh, and R. Buyya. ifogsim : A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments. *Software : Practice and Experience*, 47(9) :1275–1296, 2017.
- [4] L. Ismail and H. Materwala. Iot-edge-cloud computing framework for qos-aware computation offloading in autonomous mobile agents : Modeling and simulation. In S. Bouzeffrane, M. Laurent, S. Boumerdassi, and E. Renault, editors, *MSPN 2020. Lecture Notes in Computer Science*, volume 12605. Springer, 2021.
- [5] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wiessner. Microscopic traffic simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582, 2018.
- [6] C. Mouradian, D. Naboulsi, S. Yangui, R.H. Glitho, M.J. Morrow, and P.A. Polakos. A comprehensive survey on fog computing : State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials*, 20(1) :416–464, 2018.
- [7] H. Ouarnoughi, E. Grislin-Le Strugeon, and S. Niar. Simulating multi-agent-based computation offloading for autonomous cars. *Cluster Computing Journal*, 2021.
- [8] G.F. Riley and T.R. Henderson. The NS-3 network simulator. In K. Wehrle, M. Güneş, and J. Gross, editors, *Modeling and Tools for Network Simulation*. Springer, Berlin, Heidelberg, 2010.
- [9] Aarti Singh, Dimple Juneja, and Manisha Malhotra. Autonomous agent based load balancing algorithm in cloud computing. In *Int. Conf. on Advanced Computing Technologies and Applications (ICACTA)*, volume 45, pages 832 – 841, 2015.
- [10] Rui Wang, Yong Cao, Adeeb Noor, Thamer A Alamoudi, and Redhwan Nour. Agent-enabled task offloading in uav-aided mobile edge computing. *Computer Communications*, 149 :324–331, 2020.
- [11] U. Wilensky. Netlogo. Technical report, Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL., <http://ccl.northwestern.edu/netlogo/>, 1999.
- [12] Y. Zhang, F. Tian, B. Song, and X. Du. Social vehicle swarms : A novel perspective on socially aware vehicular communication architecture. *IEEE Wireless Communications*, 23(4) :82–89, 2016.

Maisons collaboratives

Julien Cumin^a Fano Ramparany^a Iago Felipe Trentin^{a,b}
julien1.cumin@orange.com fano.ramparany@orange.com iagofelipe.trentin@orange.com

Olivier Boissier^b
olivier.boissier@emse.fr

^aOrange Labs, 28 chemin du Vieux Chêne,
F - 38240, Meylan, France

^bMines-Saint-Étienne, Univ. Clermont Auvergne, CNRS, UMR 6158 LIMOS,
Institut Henri Fayol, F - 42023 Saint-Étienne, France

Résumé

Cet article présente une démonstration de l'utilisation des technologies multi-agents et de données liées pour l'implémentation de services prenant en compte le contexte ambiant, au bénéfice de personnes vivant dans des environnements dits intelligents. Dans cette démonstration, chaque maison d'un voisinage collaboratif est gérée par un agent dédié, qui exploite le contenu d'un graphe de connaissances de contexte de la maison. Nous mettons en œuvre un cas d'usage de sécurité, autour de la détection d'intrusions dans les maisons. Pour implémenter notre démonstrateur, nous avons utilisé la plateforme Home'in [5], et la plateforme multi-agents JaCaMo [1].

Mots-clés : systèmes multi-agents, données liées, contexte, internet des objets

Abstract

This paper presents a demonstration of the use of multi-agent technologies and linked data for the implementation of context aware services, which benefit people living in smart environments. In this demonstration, each home of a collaborative neighbourhood is monitored by a dedicated agent, which exploits the contents of a context knowledge graph of the home. We investigate a home security use case, based on situations of intrusions. To implement our demonstrator, we have used the Home'in platform [5], and the multi-agent platform JaCaMo [1].

Keywords: multi-agent systems, linked data, context, internet of things

1 Introduction

Une maison dite « intelligente » est typiquement instrumentée par des capteurs et des actuateurs, offrant des données en temps réel captées dans la maison. En analysant ces flux de données, nous pouvons inférer de l'information contextuelle comme l'identité, la localisation, et l'ac-

tivité des personnes présentes. Les technologies d'agents peuvent permettre de valoriser ces informations à travers des services de maison intelligente mieux adaptés au contexte ambiant. En particulier, ces technologies apportent des solutions d'autonomie, de proactivité, de réactivité, et peuvent agir comme médiateurs entre la maison et ses habitants.

Des cas d'usages concrets que nous envisageons sont relatifs au partage de ressources (par ex. se prêter des outils), à favoriser les interactions sociales (par ex. surveillance collaborative), et à l'opportunisme (par ex. le covoiturage).

Une vidéo de présentation de la démonstration est disponible à la lecture à l'adresse suivante : <http://ramparany.free.fr/atac/collaborativeHomesFinal.mp4>. Le code source de la démonstration est disponible à l'adresse suivante : <http://ramparany.free.fr/atac/code.tar>.

Cet article est la version française d'un article précédemment publié [3].

2 Implémentation

L'architecture de notre agent de maison est présentée sur la Figure 1. La modélisation du contexte, que nous détaillons plus en détails en Section 4, requiert un langage de représentation expressif pouvant gérer l'hétérogénéité intrinsèque des données de contexte. Le web sémantique et les technologies de données liées sont des éléments clés pour gérer cette hétérogénéité. La plateforme Home'in [5] intègre un manager de graphe de connaissances contextuelles (que nous appelons manager de contexte dans le reste de ce papier).

Ce manager de contexte implémente le modèle de contexte de chaque maison sous la forme d'un graphe RDF, via la librairie Jena [7]. Ce graphe

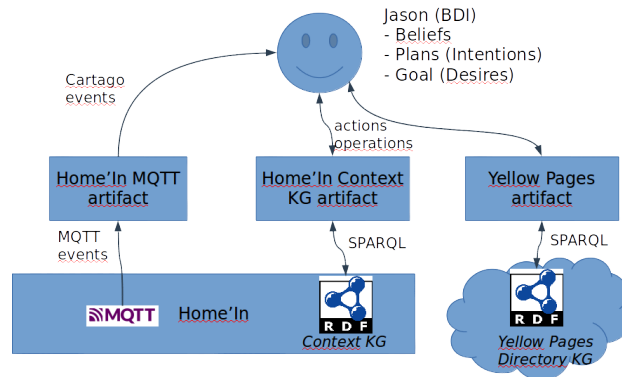


FIGURE 1 – Architecture de l’agent d’une maison.

RDF repose principalement sur les ontologies FoaF [2] et SOSA [4], classiquement utilisées dans le domaine de l’internet des objets. Une description détaillée de ce manager de contexte pour la maison est proposée par Ramparany et Cao [6]. Ce graphe de connaissances est alimenté à la volée, en traitant en temps réel les mesures de capteurs circulant sur le bus MQTT de la plateforme Home’in.

Un agent Jason surveille l’activité de la maison via un artéfact Cartago connecté à ce même bus MQTT, et émet des requêtes SPARQL, via un autre artéfact, au manager de contexte pour obtenir le contexte ambiant de la maison. Cet agent peut alors détecter et reconnaître certaines situations spécifiques, qu’il peut gérer grâce à une bibliothèque de plans.

Un autre agent, appelé agent pages jaunes, apporte des informations de contact (numéros de téléphone, adresses, etc.) de services et d’entreprises de la ville. Ces informations sont également stockées dans le graphe de connaissances.

3 Cas d’usage

Nous présentons le prototype que nous avons développé dans la Figure 2. Trois agents délibératifs analysent en continue le graphe de connaissances de leurs maisons respectives. Lorsqu’un agent détecte une intrusion dans sa maison, il exécute un plan demandant assistance aux autres agents. Si ce plan requiert une intervention de la police, l’agent de la maison récupère les informations de contact de la police via l’agent pages jaunes.

Nous présentons le comportement d’un agent de maison dans la Figure 3, avec le formalisme des automates finis. Les 3 agents de maisons exhibent le même comportement et partagent la

même implémentation de ce comportement, ce qui assure une grande cohérence dans les interactions entre ces agents. Dans ce diagramme d’états, les disques verts représentent les états des agents, et les losanges orange représentent les actions et décisions des agents. L’état d’un agent peut changer, selon des événements externes que l’agent prend en compte, ou bien selon les décisions propres de l’agent. Les transitions entre états sont représentées par des flèches qui lient l’état initial et l’état cible. Une transition barrée d’un segment perpendiculaire indique que la transition a été partiellement causée par un événement externe.

Nous détaillons ci-dessous chaque état qu’un agent de maison peut atteindre durant son exécution, et les actions et décisions qu’il peut prendre dans chaque état.

waiting : après le démarrage de l’agent (état start), celui-ci est en attente de 2 types événements externes différents : un événement de localisation, ou une requête d’aide d’un autre agent. Dans le cas d’un événement de localisation qui correspond à une présence dans la maison, l’agent requête le manager de contexte pour savoir s’il y a au moins un occupant connu dans la maison. Dans le cas où il n’y a aucune personne connue, l’agent conclut qu’il y a une intrusion et déclenche le plan correspondant. Dans le cas contraire, l’agent reste dans l’état waiting. La requête SPARQL émise par l’agent au manager de contexte est la suivante :

```
PREFIX sosa: <http://www.w3.org/ns/sosa/>
PREFIX homein: <http://orange.smart.home/HomeIn#>
ASK
WHERE {
    ?o1 sosa:resultTime ?maxt .
    ?o1 sosa:hasFeatureOfInterest ?x .
    ?o1 sosa:hasResult ?place .
    FILTER(?x != homein:unknown)
    FILTER(?place != homein:outside)
```

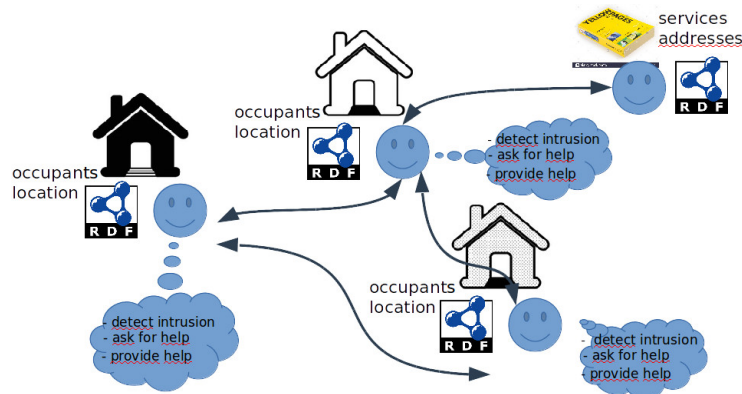


FIGURE 2 – Architecture de déploiement du prototype.

```

{
  SELECT (MAX(?t) AS ?maxt) ?x
  WHERE {
    ?o a homein:LocationObservation .
    ?o sosa:hasFeatureOfInterest ?x .
    ?o sosa:resultTime ?t .
  } GROUP BY ?x
}

```

Cette requête SPARQL contient une requête SELECT dont l'objectif est de récupérer la dernière localisation de chaque personne. On vérifie ensuite si chaque personne est connue (c'est-à-dire différente de `unknown`), et si chaque personne est dans la maison (c'est-à-dire dans un lieu différent de `outside`). Si une intrusion est détectée, l'agent lance le plan ci-dessous, exprimé en Jason :

```

+onlyUnknownOccupantDetected : true <-
  ?name(Me);
  .broadcast(achieve,pleaseGetHelpRequestFrom(Me));
  -+state(waitingFeedback);
  !!timer30000;

```

L'agent diffuse une requête d'aide, et se met dans l'état `waitingFeedback`. L'agent met également en place un timer de 30 secondes, au bout duquel il engagera un plan de secours en cas de non-réponse des autres agents; nous ne détaillons pas ce plan de secours ici.

Dans l'état `waiting`, un agent peut recevoir une requête d'aide de la part d'un autre agent. L'agent va alors vérifier si des personnes sont présentes dans sa maison respective, par une requête au manager de contexte similaire à celle utilisée pour détecter une intrusion. Si au moins une personne est présente, l'agent va informer les occupants présents et va leur demander s'ils sont prêts à aider à la résolution de cette situation. Après avoir attendu la réponse des occupants présents,

et si celle-ci est positive, l'agent informe l'agent de la maison où se déroule l'intrusion que cette maison est prête à apporter de l'aide.

waitingFeedback : dans cet état, l'agent est attendu de propositions d'aide de la part des agents du voisinage. Si une proposition d'aide est reçue dans les 30 secondes, l'agent informe les occupants absents de la situation. Puisque les occupants de la maison où se déroule l'intrusion, et ceux prêts à apporter de l'aide sont bien identifiés par leurs agents respectifs, ils peuvent communiquer par un moyen standard comme un appel téléphonique, pour organiser l'opération de sécurisation de la maison. L'agent se met alors dans l'état `processing`.

processing : dans cet état, les occupants et les aidants s'occupent de la situation. L'agent est en attente d'une confirmation que la situation a été résolue, pour retourner dans l'état `waiting`.

4 Discussion

La vidéo de la démonstration (lien dans l'introduction) présente les principaux résultats de notre projet à travers un scénario où l'agent d'une maison détecte une intrusion et demande de l'aide aux agents des autres maisons voisines. Dans cette section, nous mettons en avant certains aspects de notre prototype.

Caractéristiques des agents : comme mentionné dans la Section 3, nos agents sont délibératifs et informationnels.

Données liées utilisées : notre prototype implique 4 sources de données liées. Les 3 premières fournissent des informations sur chacune des 3 maisons. La dernière source est un agent de pages jaunes qui propose un service d'annuaire.

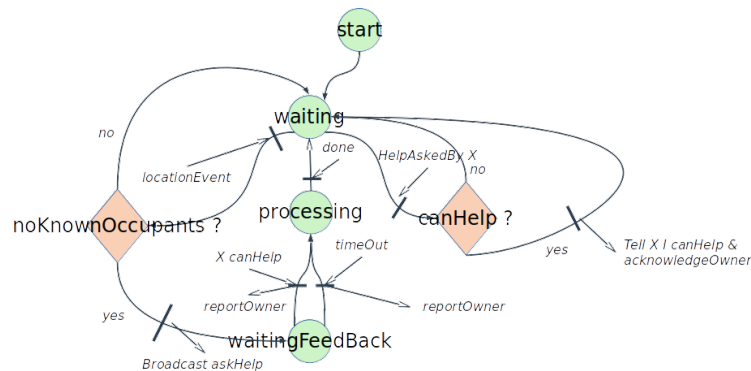


FIGURE 3 – Comportement d'un agent de maison.

Gestion de l'hétérogénéité : comme mentionné dans la section 2, les informations contextuelles de la maison sont par nature hétérogènes : identités, équipements, environnement, etc. Nous utilisons le formalisme offert par le graphe de connaissances pour gérer cette hétérogénéité. Par ailleurs, le graphe de connaissances de la maison, et le graphe de connaissances d'annuaire de l'agent pages jaunes, utilisent deux ontologies différentes. Nos agents ont également des comportements hétérogènes : les agents de maisons sont délibératifs, alors que l'agent pages jaunes est informationnel. Par leurs interactions, ces agents constituent une couche de médiation pour gérer l'hétérogénéité, en gérant et partageant leurs sources de données liées.

Gestion de la dynamique : chaque maison est équipée de capteurs (de température, de pression, et autres capteurs ambiants), dont les mesures sont utilisées pour mettre à jour en temps réel le graphe de connaissances. Dès lors, l'agent d'une maison peut évaluer à temps le contexte d'une maison et prendre des décisions rapides, ce qui s'avère essentiel dans notre cas d'usage de sécurité collaborative.

Interactions sociales entre agents : dans notre prototype, les 3 agents de maisons et l'agent pages jaunes forment une communauté qui collaborent pour assurer la sécurité collective dans le voisinage. La maison détectant une intrusion émet une requête d'aide aux autres maisons, qui répondront favorablement ou non en fonction de leurs occupations respectives. Les agents de maisons soumettent chaque décision critique à la validation des occupants, avant de réaliser les actions correspondantes.

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. Special section : The Programming Languages track at the 26th ACM Symposium on Applied Computing (SAC 2011) & Special section on Agent-oriented Design Methods and Programming Techniques for Distributed Computing in Dynamic and Complex Environments.
- [2] Dan Brickley and Libby Miller. FOAF Vocabulary Specification 0.97. Namespace document, January 2010.
- [3] Julien Cumin, Fano Ramparany, Iago Felipe Trentin, and Olivier Boissier. Collaborative home. In *All the Agents Challenge (ATAC 2021)*, 2021.
- [4] Krzysztof Janowicz, Armin Haller, Simon J. D. Cox, Danh Le Phuoc, and Maxime Lefrançois. Sosa : A lightweight ontology for sensors, observations, samples, and actuators. *CoRR*, abs/1805.09979, 2018.
- [5] Hamed Rahimi, Iago Felipe Trentin, Fano Ramparany, and Olivier Boissier. SMASH : a Semantic-enabled Multi-agent Approach for Self-adaptation of Human-centered IoT. Submitted to PAAMS 2021, June 2021.
- [6] Fano Ramparany and Quyet H. Cao. A semantic approach to iot data aggregation and interpretation applied to home automation. In *2016 International Conference on Internet of Things and Applications (IOTA)*, pages 23–28, 2016.
- [7] Andy Seaborne. Jena, a Semantic Web Framework, November 2010.

Modélisation Multi-Agent pour les Réseaux Énergétiques Insulaires

I. TALEB^a
ihab.taleb@devinci.fr

G. GUERARD^a
guillaume.guerard@devinci.fr

F. FAUBERTEAU^a
frederic.fauberteau@devinci.fr

N. NGUYEN^a
nga.nguyen@devinci.fr

^aLéonard de Vinci Pôle Universitaire, Research Center, 92 916 Paris La Défense, France

Résumé

Les Smart Grids sont devenus, depuis deux décennies, un enjeu majeur et un grand défi scientifique et technique, car ils peuvent aboutir à minimiser le réchauffement climatique grâce à une gestion intelligente des différents acteurs du réseau électrique. Les Smart Grids sont encore problématiques pour les milieux isolés et/ou insulaires, car leurs capacités de production sont limitées et avec une structure peu flexible. Nous proposons un modèle multi-agent pour les Smart Grids qui est une agrégation d'un élément de base appelé cellule, inspiré du concept de poupées russes. Les cellules sont composées d'un même groupe d'agents compétitifs qui tiennent compte des spécificités de chaque acteur du système électrique. Ces cellules sont supervisées par des cellules supérieures, suivant l'aspect structurel du réseau, qui leur attribuent des récompenses ou des sanctions afin d'apporter la flexibilité manquante.

Mots-clés : Smart Grid, Réseau énergétique insulaire, Modèle Multi-Agent

Abstract

Smart grids have become, for two decades, an urgent technical and scientific challenge as they can minimize global warming thanks to smart management techniques for the different grid actors. However, smart grids are still problematic for islands as they have limited generation resources and do not provide enough flexibility. Indeed, we propose a multi-agent model for smart grids that is merely an aggregation of a basic element called Cell, like the concept of the Matryoshka Dolls. These cells are composed of the same group of competitive agents that take into consideration the characteristics of each actor in the system. These cells are supervised by superior cells, which attribute rewards or penalties to ensure the required flexibility.

Keywords: Smart Grid, Island electric grid, Multi-agent modelling

1 Introduction

Le réseau électrique se transforme depuis une vingtaine d'année. Il incorpore de plus en plus d'intelligence à tout niveau de son système. Le réseau a commencé à muter dans les années 1980, mais c'est au milieu des années 2000 qu'un nouveau concept est né : le Smart Grid (SG).

Depuis l'émergence du SG, des nouvelles technologies de digitalisation et de communication, déjà annoncé par Massoud Amin en 2005 [1], modifie en permanence le réseau électrique monolithique et unidirectionnel des années 1960. Que ce soit la digitalisation du pilotage des lignes électrique en bidirectionnel [7, 13], l'intégration de technologies blockchains sur le marché de l'énergie et sur les échanges pairs à pairs d'auto-producteur [2, 11] ou même les jumeaux numériques de villes pour la gestion de la consommation [5], la transformation du réseau électrique s'opère à tout niveau.

Les nouvelles technologies du SG possèdent toujours des verrous tels que les îlots de consommation. Ce sont des auto-producteurs reliés à une unique ligne électrique, ou des zones peu peuplées ou dont les besoins énergétiques dépendent presque uniquement du facteur humain. C'est pourquoi nous proposons d'étudier, dans le contexte du projet H2020 MAESHA, ces réseaux isolés ou insulaires afin de pouvoir mieux les modéliser. Le projet MAESHA est un projet qui a pour but de décarboner la production électrique dans l'île de Mayotte principalement, en transformant les réseaux électriques traditionnels basés sur les centrales thermiques en y intégrant les énergies renouvelables et des capacités de stockage.

Cette modélisation sera une plate-forme permettant de tester des technologies et de simuler le réseau afin d'en voir son évolution et son acceptation.

Dans un premier temps, nous verrons les technologies du SG dans la Section 2. Puis nous ana-

lyserons ces dernières pour en comprendre les limites en milieux insulaires en Section 3. La Section 4 conclura sur notre proposition de modèle.

2 Le Smart Grid

L'ensemble des informations de cette section sont une synthèse de revues de littérature de références sur les SGs datant de 2011, 2016 et 2020 [3, 4, 9], montrant les évolutions scientifiques et technologiques au cours de la dernière décennie.

Le SG est une évolution du réseau électrique et énergétique rompant avec l'ancien réseau sur les points suivants : digitalisation, collecte et traitement des informations, distribution bidirectionnelle, génération distribuée (batteries, énergie renouvelable), pilotage automatique, outils de stabilisation du réseau et de flexibilité automatique, gestion de la demande et marché dynamique. Toutes ces transformations ont pour objectif d'améliorer le réseau électrique que ce soit en flexibilité, diminuer l'impact carbone de tous les acteurs en optimisant les divers flux et marchés côté consommateurs et producteurs. Le SG est souvent décomposé en trois systèmes : le système de protection intelligente (à la fois le *self-healing* et la cybersécurité), l'infrastructure intelligente (digitalisation de la structure, nouveaux paradigmes comme les microgrids ou le *Vehicles-to-Grid*, *Power-to-Gas*, etc.) et le système de gestion intelligente qui sert à contrôler et gérer les différents acteurs de l'énergie (marché dynamique, échange pairs à pairs, gestion de la demande, demande-réponse).

Les modèles de gestion intelligente peuvent être construits sur deux paradigmes : optimisation (modélisation mathématique par optimisation) ou *data-driven*. Ces derniers construisent les modèles mathématiques à l'aide d'apprentissage machine et d'apprentissage profond [12].

Les modèles *Data-Driven* permettent une analyse systémique du réseau électrique. Ils sont construits pour répondre à des cas d'usages multiples. Ces modèles représentent le système dans son entièreté ou uniquement certains de ses sous-systèmes tels que : contrôle préventif, estimation ou prédiction de l'état des modules (comme la consommation), classification et détection d'évènements, calibration et simulation, vérification de l'intégrité, classification et détection de profils (comme des consommateurs ayant des courbes / comportements similaires).

3 Les limites

Cependant, les modèles sont construits pour des réseaux continentaux et ne prennent pas en compte les particularités de réseaux plus petits ou insulaires. Voici une liste non exhaustive des limites des modèles actuelles.

Outils de prédiction à toute granularité – Les réseaux insulaires étant peu interconnectés et avec de nombreuses ramifications en moyenne et basse tension, une baisse de tension peut rapidement mettre à mal l'ensemble du réseau. Afin d'éviter toute erreur dans le réseau, il est important d'avoir des outils de prédiction de la consommation et de la production à toute granularité d'espace et de temps. Il faut connaître les tendances au niveau local ainsi qu'à grandes échelles, sur un temps proche comme une heure autant que sur une journée [8].

Amélioration des outils de pilotage et détection des erreurs – La prédiction préventive est essentielle pour éviter qu'un problème ponctuel n'entraîne un effet de cascade dans l'ensemble du réseau

Profilage des consommateurs – Dans le but de pouvoir faire correspondre l'offre et la demande, il est important de pouvoir classifier les consommateurs et auto-producteurs. Le profilage comprend un aspect social et économique pour chaque cas d'usage des bâtiments. Cela est réalisable en utilisant une classification non supervisée mixte [10].

Amélioration des outils de demande-réponse – Les outils existants de demande-réponse sont calibrés sur un prix de l'énergie dynamique. Cependant, compte tenu du manque de flexibilité et de résilience d'un réseau insulaire, ces outils doivent aussi prendre en compte un facteur de risque non basé sur la stabilité du réseau et l'impact socio-économique que cela engendrera [6]. En effet, le coût d'une action sur le consommateur et le producteur aura un impact sur le système plus important que dans un réseau continental.

Marché de l'énergie et prix dynamique – Le prix de l'énergie a tendance à devenir dynamique, donc relatif au marché. Tels les outils de demande-réponse, les aspects de résilience et socio-économique sont là aussi à prendre en compte. Ainsi, le prix dynamique ne devra pas dépendre uniquement du coût de production

mais aussi de l'impact du flux énergétique sur le réseau.

Modélisation multi-agent comme banc de test – Le SG possède des acteurs sociaux, économiques, structurels, de communication et d'échange, qui ont chacun leurs objectifs et contraintes et peuvent soit collaborer soit entrer en compétition avec les autres acteurs. Il est donc primordial de posséder une modélisation viable d'un SG dans toute sa complexité, sous-entendu en tant que système complexe, avec les technologies sélectionnées existantes ou à tester.

4 Le système proposé

4.1 Architecture

Le système est proposé principalement dans le but d'avoir le maximum de flexibilité possible dans les domaines spatiaux et temporels. En effet, pour bien garantir la flexibilité spatial, le SG dans tous ces niveaux possibles (régions, quartiers, appartements, etc.) est composé d'une seule structure que nous appelons *cellule*. Chacune de ces cellules est composée des cinq agents suivants (Figure 1) :

- **Agent de mesures** : cet agent est responsable de capter, traiter et restituer les valeurs réelles de la consommation et/ou la production de l'électricité, et de les envoyer à l'agent de données.
- **Agent de prédiction** : cet agent doit être capable de donner des prédictions dans des différentes marges temporelles (flexibilité temporelle), pour la consommation et la production, selon le type de la cellule et ces besoins.
- **Agent de contrôle** : cet agent surveille les cellules sous sa responsabilité, donc d'après une hiérarchie basée sur la structure du réseau (consommateur, agrégateur, etc.) pour leur attribuer des récompenses ou des sanctions dans leur stratégie de demande-réponse. Le niveau le plus bas doit gérer les appareils énergétiques afin de proposer des stratégies de demande-réponse.
- **Agent de données** : cet agent est responsable d'enregistrer les données et les connaissances liées aux cellules et leurs niveaux relatifs par rapport à cette cellule.
- **Agent de communication** : cet agent est responsable de la communication avec les autres cellules.

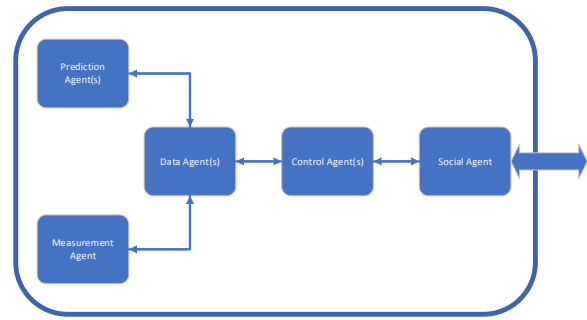


FIGURE 1 – La structure générale des cellules.

Le système de *cellules* est compatible avec tous les types de générateurs, de consommateurs et d'auto-producteurs. Une *cellule* peut aussi représenter une entité abstraite tel qu'un agrégateur.

Les *cellules* ont également différents rôles. En effet, elles surveillent les cellules inférieures et leurs attribuent des récompenses et des sanctions. Elles sont surveillées par les cellules les plus supérieures d'une manière similaire.

Étant donné que le réseau sera complètement construit par le même type de cellule, les cellules dirigeantes peuvent facilement détecter et fournir (et si c'est impossible, isoler) les cellules qui ont un comportement inhabituel pouvant mettre en péril la stabilité. Par conséquent, nous évitons les black-outs en isolant les cellules qui sont les causes du problème. Le deuxième avantage est que tous les utilisateurs peuvent jouer les rôles de consommateur et de producteur, ce qui facilite l'intégration des sources d'énergies distribuées en intégrant les auto-producteurs, les énergies renouvelables et les systèmes de stockage énergétique.

Pour finir, une cellule est entièrement paramétrable. Il est donc possible de pré-fabriquer des profils de consommateurs et de producteurs pour les intégrer dans la simulation. La modélisation se voulant sans-contexte, tout utilisateur peut simuler le réseau de son choix.

4.2 Simulation et scénarios

La simulation multi-agent, en considérant l'adaptation nécessaire précisée dans la section précédente, permettra de comprendre et de voir l'émergence, l'auto-organisation et les comportements convergents d'un réseau. Compte tenu de la particularité des milieux insulaires, la simulation intégrera les scénarios suivants :

- Intégration de nouvelles technologies :

les cellules doivent pouvoir incorporer toutes technologies nouvelles.

- Intégration de nouveaux paradigmes : comme préciser précédemment, de nouveaux paradigmes et algorithmes devront être créés puis éprouvés par le modèle.
- Intégration de particularité propre au milieu insulaire : les conditions climatiques et catastrophes naturelles peuvent mettre le réseau à rude épreuve. La simulation doit prendre en compte ces éléments externes.
- Résilience et efficacité : le modèle doit être capable de s'adapter à tous imprévus afin de garder le système le plus flexible et résilient possible.

5 Conclusion

Dans cet article, nous avons réalisé un état des lieux des besoins et les développements dans le domaine des systèmes multi-agent en général et notamment dans le cas des zones insulaires. Cet article propose une structure d'un système multi-agent qui se concentre principalement sur la flexibilité et l'adaptabilité du système. En effet, le système est composé d'un seul composant élémentaire qui peut représenter et diriger tous les types d'utilisateurs (consommateurs, générateurs et auto-producteurs). Ce composant élémentaire est appelé *cellule* et est composé de cinq types des agents représentant les aspects énergétique, social et économique. Chaque *cellule* est organisée dans une hiérarchie représentant l'aspect structurel du réseau qui est nécessaire pour garantir sa flexibilité et son intégrité.

Remerciements

Ce projet a reçu un financement du programme de recherche et d'innovation Horizon 2020 de l'Union européenne dans le cadre de la convention de subvention n° 957843.

Références

- [1] S. M. Amin and B. F. Wollenberg. Toward a smart grid : power delivery for the 21st century. *IEEE power and energy magazine*, 3(5) :34–41, 2005.
- [2] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock. Blockchain technology in the energy sector : A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 100 :143–174, 2019.
- [3] G. Dileep. A survey on smart grid technologies and applications. *Renewable Energy*, 146 :2589–2625, 2020.
- [4] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid—the new and improved power grid : A survey. *IEEE communications surveys & tutorials*, 14(4) :944–980, 2011.
- [5] A. Francisco, N. Mohammadi, and J. E. Taylor. Smart city digital twin-enabled energy management : Toward real-time urban building energy benchmarking. *Journal of Management in Engineering*, 36(2) :04019045, 2020.
- [6] N. Good, K. A. Ellis, and P. Mancarella. Review and classification of barriers and enablers of demand response in the smart grid. *Renewable and Sustainable Energy Reviews*, 72 :57–72, 2017.
- [7] G. Guérard, S. B. Amor, and A. Bui. A context-free smart grid model using pretopologic structure. In *2015 International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, pages 1–7. IEEE, 2015.
- [8] G. Guerard, H. Pousseur, and I. Taleb. Isolated areas consumption short-term forecasting method. *Energies*, 14(23) :7914, 2021.
- [9] Y. Kabalci. A survey on smart metering and smart grid communication. *Renewable and Sustainable Energy Reviews*, 57 :302–318, 2016.
- [10] L.-N. Levy, J. Bosom, G. Guérard, S. B. Amor, M. Bui, and H. Tran. Application of pretopological hierarchical clustering for buildings portfolio. In *SMARTGREENS*, pages 228–235, 2021.
- [11] Z. Nehai and G. Guérard. Integration of the blockchain in a smart grid model. In *The 14th International Conference of Young Scientists on Energy Issues (CYSENI) 2017*, pages 127–134, 2017.
- [12] C. Tu, X. He, Z. Shuai, and F. Jiang. Big data issues in smart grid—a review. *Renewable and Sustainable Energy Reviews*, 79 :1099–1107, 2017.
- [13] X.-P. Zhang, C. Rehtanz, and B. Pal. *Flexible AC transmission systems : modeling and control*. Springer Science & Business Media, 2012.

Une approche collaborative pour la recherche de places de stationnement

J-D. Collard^a
jdc@angelsbaytech.com

E. Stattner^b
erick.stattner@univ-antilles.fr

P. Gergos^a
jdc@angelsbaytech.com

^a AngelsBayTech SAS, France

^bLaboratoire de Mathématiques, Informatique et Applications (LAMIA), France

Résumé

La recherche de places de parking en voirie est devenue un problème majeur. Au-delà des solutions qui consistent à restreindre le nombre de véhicules ou augmenter le nombre de places, il existe des solutions intelligentes basées le plus souvent sur l'exploitation de capteurs ou de caméras. De telles solutions sont peu flexibles, lourdes à mettre en œuvre et à maintenir et de fait coûteuses. Cet article présente une solution collaborative. Une modélisation basée sur les agents est proposée et des simulations computationnelles sont menées afin d'étudier la pertinence d'une solution parcimonieuse qui ne nécessite pas la mise en œuvre d'infrastructures matérielles, ouverte à chacun en tout lieu et qui promeut l'entraide entre les citoyens.

Mots-clés :

Mobilité urbaine intelligente ; approche collaborative

Abstract

The search for on-street parking spaces has become a major problem. In addition to solutions that involve limiting the number of vehicles or increasing the number of spaces, there are intelligent solutions, most often based on the use of sensors or cameras. Such solutions are not really flexible, are cumbersome to implement and maintain and are indeed costly. This article presents a collaborative solution. Agent-based modelling is proposed and computational simulations are conducted to study the relevance of a parsimonious solution open to everyone everywhere, that promotes mutual aid between citizens and does not require the implementation of physical infrastructures.

Keywords:

Intelligent urban mobility ; collaborative approach

1 Introduction

Cet article s'intéresse au problème de la gestion des *places de stationnement* qui est un défi pour réduire le trafic, limiter la consommation d'énergie et diminuer les émissions de gaz à effet de serre [1]. Des travaux menés sur la ville de Los Angeles (USA) ont montré que la recherche de parking génère environ 730 tonnes de dioxyde de carbone (CO₂), 95000 heures perdues et 200000 litres de carburant. De tels impacts avaient déjà été observés dans une étude de 2006 [2].

Les travaux précurseurs pour une gestion intelligente des parkings ont exploité des capteurs matériels capables de collecter des données brutes. Les travaux plus récents ont abordé cette question sous l'angle des *systèmes complexes* [3] dans lequel des agents interagissent pour faciliter la gestion des espaces de stationnement. Le potentiel des approches collaboratives pour traiter ce problème avait déjà été mis en évidence dans [4], où les auteurs montrent comment "*la détection collaborative peut apporter des avantages clés dans le transport urbain, en contribuant à une plus grande efficacité des déplacements, à la sécurité et à la réduction de la pollution, grâce à des applications innovantes qui bénéficient de données produites collaborativement*". D'autres études ont par exemple adopté des modèles multi-agents pour simuler différents types de comportements collaboratifs entre conducteur cherchant une place et conducteur offrant une place [5].

Cet article présente une modélisation à base d'agents pour évaluer l'efficacité d'une gestion collaborative des places de stationnement. Nous qualifierons de *donneur* un conducteur dont le véhicule est garé et qui quittera éventuellement son stationnement afin de le "donner" à un autre conducteur à la recherche d'une place ; de même, un *preneur* sera un conducteur qui circule dans les rues et qui sera éventuellement prêt à "prendre" une place libérée par un *donneur*.

Si l'on voit les places de parking comme des *ressources* alors les donneurs sont des *producteurs* et les preneurs des *consommateurs*. Nous sommes alors dans un schéma *producteur-consommateur* où chacun joue alternativement les deux rôles. Le problème du stationnement peut être formulé en disant que, à tout instant, on doit trouver (ou maintenir) un équilibre entre l'offre et la demande. L'approche collaborative consiste à créer, selon les besoins, des liens directs entre producteurs et consommateurs.

2 Mobilité collaborative

Cet article considère le cas limite où la voirie est totalement saturée, c'est-à-dire qu'il n'existe aucun parking vacant. La seule façon de se garer est de "prendre" la place d'une voiture en stationnement prête à quitter son emplacement. Ainsi un *preneur* deviendra un *donneur* et réciproquement. Le nombre de *preneurs* $|P|$ et *donneurs* $|D|$ sont donc constants.

Pour équilibrer l'offre et la demande, et donc favoriser les échanges, il faut que les nombres d'occurrences, (i) recherche d'une place par un *preneur* et (ii) cession de sa place par un *donneur*, soient proches, c'est-à-dire que le *rapport Offre/Demande* (ci-après noté O/D) soit proche de 1.

Par conséquent, $O/D = \frac{|D|}{|P|} \times \frac{\text{dureeDeplacement}}{\text{dureeStationnement}}$ où *dureeStationnement* est le temps pendant lequel un véhicule reste stationné et *dureeDeplacement* est le temps entre deux stationnements consécutifs d'un même véhicule. On suppose ici que la demande est inférieure à l'offre ($O/D < 1$); le temps pour trouver une place est donc significatif et celui pour libérer une place est négligeable.

2.1 Stratégie de recherche collaborative

Pour qu'il y ait un échange de places entre un *preneur* et un *donneur*, les deux conducteurs doivent se trouver à proximité au même moment. Afin de relâcher cette contrainte spatio-temporelle, nous proposons une solution qui s'appuie sur une stratégie collaborative. Une telle stratégie a déjà été implémentée dans cadre du projet de service urbain d'aide au stationnement *ReadyPark* qui, *via* une application mobile facilite en temps réel la recherche de places de parking sur la voirie [6].

Un échange de place est basé sur le jumelage entre un *preneur* en recherche de place et un *donneur* prêt à quitter sa place; le jumelage

prend en compte des critères tels que la proximité, les attributs des voitures, les caractéristiques des conducteurs (PMR, médecin, aide-à-la-personne, ...), etc. Finalement le *preneur* est guidé vers le *donneur* *via* un itinéraire visible sur l'écran de son portable.

L'objectif ici est de formaliser la démarche suivie lors de la conception de ce service afin de valider *via* des simulations la pertinence des choix conceptuelles retenus. Précisons ici que l'objectif n'est pas de promouvoir la création d'une communauté fermée car il s'agit d'une solution ouverte à tous les usagers qui, de plus, priorise les personnes en difficulté ou qui œuvrent pour l'intérêt commun.

2.2 Modèle de Manhattan

La "ville" est modélisée selon le modèle de *Manhattan* par une grille 2-D qui représente une structure de rues horizontales et verticales délimitant des blocs d'immeubles de taille similaire. Les places de stationnement sont positionnées sur chaque coté des rues. Un conducteur est un agent qui soit se déplace dans les rues soit demeure immobile sur un parking. La vitesse de déplacement est constante et identique pour tous.

2.3 Dynamique des agent-conducteurs

Un conducteur a un statut qui indique s'il coopère (*statut* = *C*) ou il s'agit d'un conducteur *standard* (*statut* = *S*). On notera C_ratio la proportion de coopérateurs dans la population. Un conducteur a un rôle (donneur ou bien preneur) qui varie au cours du temps.

Un conducteur a un état qui peut prendre successivement sept valeurs différentes (Figure 1). Au pas de temps $t = 0$, un conducteur est soit dans l'état [1] soit dans l'état [-1] et son rôle est respectivement preneur ou donneur.

Transition d'état.

[1 → 2] déclenchée dès que le temps *dureeDeplacement* est écoulé.

[2 → 3] déclenchée dès que le preneur est jumelé à un donneur dans l'état [-2].

[3 → 0] déclenchée dès que le preneur est proche du donneur jumelé.

[0 → -1] et [0 → 1] correspond à l'échange physique des positions entre les jumelés; les rôles sont alors permutés.

$[-1 \rightarrow -2]$ déclenchée dès que le temps $dureeStationnement$ est écoulé.

$[-2 \rightarrow -3]$ déclenchée dès que le donneur est jumelé à un preneur dans l'état [2].

$[-3 \rightarrow 0]$ déclenchée dès que le preneur jumelé est proche du donneur.

A chaque carrefour, dans l'état [1] la nouvelle direction est aléatoire et dans l'état [3] le preneur choisit le chemin le plus court pour atteindre le donneur.

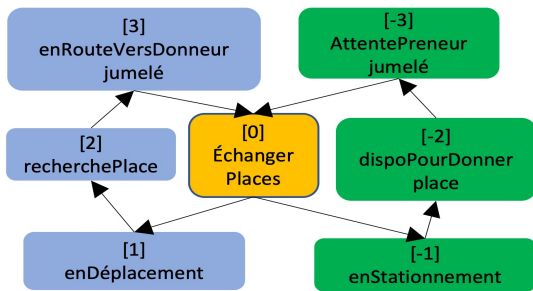


FIGURE 1 – Diagramme d'états des agents

Stratégies de jumelage. Un jumelage intervient entre un preneur dans l'état [2] et un donneur dans l'état [-2]. Cependant, la stratégie de jumelage varie selon le statut du preneur. Au sein des coopérateurs, chaque donneur priorise le jumelage avec un preneur coopérateur; cependant, comme cela s'opère au détriment des non-coopérateurs, il existe une limite temporelle à cela fixée par le paramètre $dureePrioriteC$.

- Si le preneur est un coopérateur, le donneur est soit lui-même un coopérateur *éloigné* qui priorise sa communauté soit un conducteur en stationnement à *proximité*, coopérateur ou non.

- Si le preneur est un non-coopérateur, le donneur disponible pour donner sa place doit se trouver à *proximité*; de plus, si le donneur est un coopérateur, le temps pendant lequel il a priorisé sa communauté pour un jumelage doit avoir expiré.

3 Simulations et Résultats

Les simulations ont été réalisées avec l'environnement de simulation *Netlogo* [7]. Les agents conducteurs sont représentés par des *turtles* *Netlogo*. Le "monde" est une grille de 99×99

1. Une "turtle *Netlogo*" représente un agent mobile capable de se déplacer de patch en patch.

patches *Netlogo*. Un *immeuble* est constitué de 10×10 patches.

3.1 Calibration

Un *patch* correspond à un espace carré de 5×5 mètres. La ville est donc constituée d'immeubles de 50×50 mètres, la longueur d'une place de parking est de 5 mètres et entre deux carrefours consécutifs, il y a 10 places de stationnement, soit 50 mètres.

Le temps est discrétisé. Un véhicule se déplace à une vitesse de 15 km/h, il parcourt donc 50 mètres en 12 secondes. Comme pendant un *tick*, un véhicule avance d'un patch (soit 5 mètres), un tick correspond à 1,2 seconde; c'est la fréquence de mise à jour de l'état et de la position des agents-conducteur. Les durées de déplacement et de stationnement sont fixées respectivement à 20 minutes et 90 minutes. La durée de priorité $dureePrioriteC$ est fixée à 2 minutes.

3.2 Résultats

L'offre étant insuffisante pour satisfaire la demande, on s'intéressera uniquement au temps moyen passé par les preneurs dans les états [2] et [3], soit d_2 et d_3 ces durées.

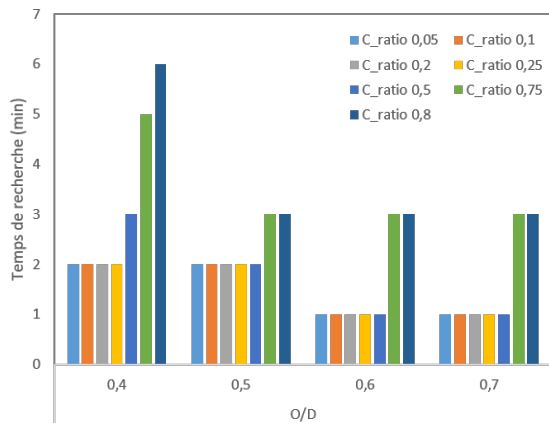
Temps passé dans chaque état. Le premier résultat est que dans toutes les configurations testées avec O/D dans l'intervalle $[0, 4..0, 7]$ et C_ratio dans $[0, 05..0, 8]$ le temps moyen d_3 pour rejoindre une place est toujours inférieur à la minute ce qui signifie que la distance qu'un preneur doit parcourir pour atteindre le donneur est inférieure à 250 mètres.

Nous pouvons donc nous concentrer sur l'étude de la durée moyenne d_2 comme indicateur de performance. En déterminant *via* des simulations la valeur d_2 pour les coopérateurs et les non-coopérateurs nous pourrions mesurer le gain apporté par la solution collaborative et l'impact de l'entre-aide sur les non-coopérateurs. Nous noterons $d_2^C(C_ratio)$ (resp. $d_2^S(C_ratio)$) la moyenne du temps passé dans l'état [2] calculée sur l'ensemble des coopérateurs (resp. conducteurs standards) pour une valeur donnée de C_ratio . On peut visualiser ces valeurs moyennes sur la figure 2 pour quatre valeurs de O/D . Comme attendu, on observe que le temps de recherche décroît lorsque le

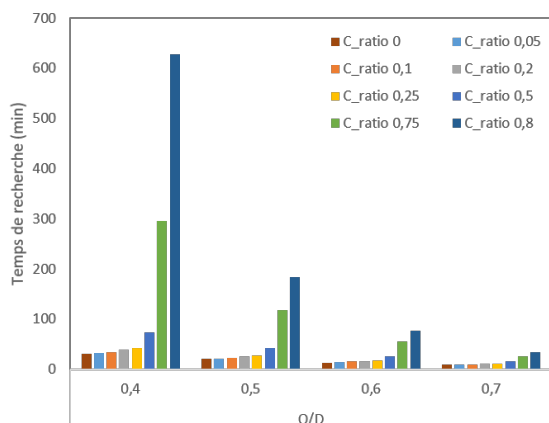
2. Un "patch *Netlogo*" représente un lieu fixe.

3. Un "tick *Netlogo*" est un pas de temps élémentaire du simulateur

ratio offre/demande croit. On observe également que pour un ratio offre/demande donné, le temps de recherche croit significativement avec le C_ratio pour les non-coopérateurs, alors qu'il est relativement stable pour les coopérateurs (du moins si $C_ratio \leq 0,5$)).



(a) Coopérateurs



(b) Non-coopérateurs

FIGURE 2 – Temps moyen de recherche d’une place de stationnement en fonction de C_ratio pour quatre valeurs de O/D .

Impact pour les non-coopérateurs. Les résultats précédents en faveur de l’approche collaborative doivent être mis en regard de l’impact de l’entre-aide sur les conducteurs standard.

Pour cela nous étudions l’augmentation du temps d’attente résultant de la présence des coopérateurs puis nous en déduisons le pourcentage d’augmentation par rapport à la situation sans coopérateur. Les résultats montrent que si la proportion de coopérateurs est inférieure à 0,20 alors l’augmentation du temps d’attente pour les non-coopérateurs est de l’ordre de 10% (Table 1).

TABLE 1 – $(d_2^S(C_ratio) - d_2^S(0))/d_2^S(0)$

O/D	0,05	0,1	0,2	0,5
0,4	6%	13%	30%	144%
0,5	4%	12%	26%	112%
0,6	5%	11%	24%	97%
0,7	3%	8%	23%	93%

4 Conclusion

Cet article a présenté une approche collaborative pour la recherche de places de stationnement en milieu urbain. L’idée principale est de desserrer la contrainte de coïncidence spatio-temporelle en jumelant un "preneur" et un "donneur" de place. Un coopérateur à la recherche d’une place de stationnement est informé *via* une application mobile qu’une place est sur le point de se libérer, puis, après jumelage, il est guidé vers le point de rencontre.

Les simulations computationnelles ont permis de montrer l’efficacité de l’approche tout en fixant les limites de son applicabilité afin de respecter pour les non-coopérateurs un seuil d’acceptabilité.

Références

- [1] Amin Kianpisheh, Norlia Mustaffa, Pakapan Limtrairut, and Pantea Keikhosrokiani. Smart parking system (sps) architecture using ultrasonic detector. *International Journal of Software Engineering and Its Applications*, 6(3) :55–58, 2012.
- [2] Donald C Shoup. Cruising for parking. *Transport policy*, 13(6) :479–486, 2006.
- [3] Nino Boccara. *Modeling complex systems*. Springer Science & Business Media, 2010.
- [4] S. Ilarri, O. Wolfson, and T. Delot. Collaborative sensing for urban transportation. *IEEE Data Eng. Bull.*, 37(4) :3–14, 2014.
- [5] Michal Chalamish, David Sarne, and Raz Lin. Enhancing parking simulations using peer-designed agents. *IEEE Transactions on Intelligent Transportation Systems*, 14(1) :492–498, 2012.
- [6] J-D. Collard, E. Stattner, and P. Gergos. The “readypark” collaborative parking search strategy. volume 4, pages 1130–1145.
- [7] Uri Wilensky. Netlogo. Technical report, Center for Connected Learning and Computer-Based Modeling, 1999.

Comment les échecs de preuve peuvent aider à la correction de spécifications erronées de Systèmes Multi-Agents

B. Mermet^a G. Simon^a
Bruno.Mermet@unicaen.fr Gaële.Simon@unicaen.fr

^aGreyc - UMR 6062 & Université Le Havre - Normandie,
Caen, France

Depuis plusieurs années, nous avons défini le modèle GDT4MAS (dont la sémantique repose sur la LTL) afin de spécifier formellement les systèmes multi-agents dans l'objectif de permettre leur validation par la preuve. A cet effet, nous avons effectué des travaux sur l'utilisation des techniques de preuve de théorèmes dans ce contexte. Nous avons pour cela défini un système de génération d'obligations de preuve à partir de la spécification GDT4MAS, obligations de preuve qui peuvent ensuite être confiées à un prouveur de théorèmes.

Depuis, nous avons amorcé une nouvelle phase qui consiste à étudier comment les outils de preuve peuvent être exploités pour aider à mettre au point des SMA. L'idée est d'exploiter la sémantique des échecs de preuve pour essayer de mettre en évidence une ou plusieurs erreurs dans la spécification GDT4MAS. Dans un précédent article [1], nous avons montré sur une étude de cas comment la structuration des obligations de preuve générées par notre système pouvait aider à la localisation de l'erreur dans la spécification. Pour aller plus loin, il faut pouvoir proposer des outils capables dans un premier temps d'aider à caractériser la cause d'un échec de preuve et dans un deuxième temps de proposer quand c'est possible des pistes de solutions. Dans cet article, nous présentons une étude de cas préliminaire qui rentre dans ce cadre.

Plus précisément, ce travail montre comment on peut extraire des graphes de dépendances de variables à partir des obligations de preuve ayant donné lieu à des échecs de preuve. Un graphe de dépendances de variables permet de représenter les liens qui existent entre les différentes variables intervenant dans une obligation de preuve à tous les instants considérés. L'absence de chemin reliant deux variables figurant dans le but d'une obligation de preuve est ainsi souvent révélateur d'une hypothèse manquante, liée à une erreur dans la spécification.

Nous détaillons ainsi différents types d'erreurs classiques dans les spécifications de systèmes multi-agents et comment ces graphes peuvent être exploités pour aide à corriger la spécification. Ainsi, l'absence d'arc entre 2 nœuds particuliers peut être révélateur d'un manque de synchronisation, menant à un bug qui ne surviendra que lors de certaines exécutions, et donc difficile à mettre en évidence par des techniques de débogage classiques. D'autres absences d'arcs peuvent être révélatrices de comportements sous-spécifiés, autorisant par exemple des modifications de variables qui ne devraient pas être possibles. Nous montrons cependant que la même caractéristique sur un graphe de variables peut être révélateur d'un autre défaut de conception, et nous proposons une méthode permettant de différencier les 2 cas. Enfin, nous identifions 3 autres types d'erreurs courants qui ne sont par contre pas révélés par les graphes de variables. Cette information aide malgré tout à s'orienter vers des hypothèses d'erreurs, et nous envisageons, par la suite, de travailler à discriminer ces différentes erreurs.

Cette approche est complémentaire des techniques classiques de débogage non seulement car elle permet de mettre en évidence des erreurs qui ne se traduiraient que par des bugs intermittants, mais aussi parce que, dans certains cas, elle permet d'obtenir directement une solution pour corriger le problème.

Références

- [1] Bruno Mermet and Gaële Simon. Using proof failures to help debugging MAS. In Ana Paula Rocha, Luc Steels, and H. Jaap van den Herik, editors, *Proceedings of the 11th International Conference on Agents and Artificial Intelligence, ICAART 2019, Volume 2, Prague, Czech Republic, February 19-21, 2019*, pages 523–530. SciTePress, 2019.

Apprentissage profond & systèmes multi-agents: Application à la segmentation d'images médicales

M.T. Bennai^{a,b} L. Gally^b Z. Guessoum^{b,c}
m.bennai@univ-boumerdes.dz lyse.gally@hotmail.fr zahia.guessoum@univ-reims.fr

S. Cormier^b S. Mazouzi^d
stephane.cormier@univ-reims.fr s.mazouzi@univ-skikda.dz

^aLaboratoire LIMOSE, Faculty des Sciences, University M'hamed Bougara de Boumerdes,
Avenue de l'indépendance, 35000, Boumerdes, Algérie

^bUniversité de Reims Champagne Ardenne, CReSTIC EA 3804
51097 Reims, France

^cSorbonne Université, LIP6 UMR7606,
75005 Paris, France

^dDépartement informatique, University 20 Août 1955,
Skikda, Algérie

La complexité de l'anatomie humaine et les artefacts d'acquisition des images médicales rendent l'analyse et l'interprétation de ces dernières très complexe. Ce processus se concentre principalement sur la détection et la localisation précise de tissus anormaux (tumeurs cérébrales par exemple) et des structures saines environnantes (matière blanche ou matière grise). L'objectif est de poser un diagnostic fiable et poser un pronostic précis. Pour aboutir à cela, l'une des premières étapes de l'analyse d'images médicales, et aussi l'une des plus importantes, est la segmentation. La segmentation sémantique est plus précisément le processus qui permet, grâce à un algorithme d'Intelligence Artificielle, de classer chaque pixel d'une image avec un label particulier.

Différentes approches ont été proposées pour automatiser cette segmentation. Cependant, la plupart des solutions existantes utilisent des connaissances préalables et/ou nécessitent une forte interaction avec l'utilisateur.

Depuis 2014, le Deep Learning en général, et les réseaux de neurones convolutifs (CNNs) en particulier, ont clairement révolutionné le domaine de la segmentation d'images médicales. Ainsi, différents types de modèles de Deep Learning obtiennent les meilleures performances dans la plupart des benchmarks pour l'analyse d'images médicales. Cependant, pour une exploitation optimale, les modèles de Deep Learning nécessitent de la puissance de calcul ainsi que la disponibilité d'une grande quantité de données d'entraînement manuellement étiquetées [1]. Ces deux conditions rendent difficile l'exploitation de ces modèles en conditions réelles de traite-

ment d'images médicales. Il est donc nécessaire de poursuivre les recherches s'intéressant à l'utilisation d'autres outils de l'Intelligence Artificielle, comme les systèmes multi-agents.

Dans cet article, nous proposons de comparer deux approches de segmentation : 1) une approche multi-agent que nous avons proposée pour la segmentation d'images médicales 3D, basée sur un ensemble d'agents autonomes et interactifs qui utilisent un algorithme modifié de croissance de régions et coopèrent pour segmenter une image 3D, et 2) une approche de type apprentissage profond basé sur U-Net. U-Net est l'une des architectures de type CNN les plus connues pour la segmentation d'images médicales, car elle permet tout d'abord d'obtenir des performances de segmentation supérieures en comparaison aux autres modèles d'apprentissage automatique, mais également car elle parvient à être efficace, même entraînée sur un ensemble limité d'images.

Ce travail représente une première étape d'un projet s'attachant à déterminer comment nous pouvons, a priori, en fonction des images et de la taille du data set, déterminer un seuil pertinent pour l'utilisation d'une méthode ou d'une autre.

Références

- [1] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

Architecture SMA/RàPC enrichie d'outils d'aide à la décision pour prédire les risques en chirurgie

Bruno Perez^a Christophe Lang^a Julien Henriët^a Laurent Philippe^a
 bruno.perez@univ-fcomte.fr christophe.lang@univ-fcomte.fr julien.henriet@univ-fcomte.fr laurent.philippe@univ-fcomte.fr

Frédéric Auber^b
 frederic.auber@univ-fcomte.fr

^aInstitut FEMTO-ST/CNRS Université de Bourgogne-Franche-Comté, France

^bCHRU Becaçon Centre Hospitalier Régional Universitaire Besançon, France

Une intervention chirurgicale ne peut être dissociée de la sécurité du patient, et donc des risques induits par un environnement technique de plus en plus sophistiqué et pluridisciplinaire.

Ainsi, l'Organisation Mondiale pour la Santé (OMS) préconise la mise en œuvre de protocoles pour assurer la sécurité du patient. Cependant, même dans cet environnement sécurisé, les événements indésirables associés aux soins (EIAS) persistent [1] (erreurs de dosage, maladroresses gestuelles, fatigue humaine...).

Dans ce contexte, nous avons modélisé l'environnement et les ressources de la salle d'opération afin de générer des alertes mais aussi afin de prédire et de quantifier l'exposition du patient au risque. Ce projet a soulevé en particulier trois difficultés majeures liées à l'univers du bloc opératoire assimilable à un système complexe non déterministe. La première concerne la capacité à produire ce type d'événement. En effet, comment générer une situation sans la caractériser au préalable ? La deuxième difficulté est liée à la première et porte sur l'identification de ces événements indésirables graves. Enfin, la troisième difficulté concerne la prévisibilité de ces situations afin de les éviter. Nos contributions ont donc consisté à faire émerger des situations à risques a priori imprévisibles, à capitaliser l'ensemble des simulations (base d'expériences), à analyser les EIAS et à les prédire. Pour atteindre ces objectifs, nous avons choisi d'une part de coupler un système multi-agents (SMA) avec un raisonnement à partir de cas (RàPC), d'autre part d'intégrer des outils d'aide à la décision. Dans cette architecture logicielle présentée par la Figure 1, des agents produisent des phénomènes émergents en interagissant entre eux et le RèPC permet de résoudre des problèmes par analogie avec des expériences passées. Les différentes espèces qui composent le SMA ont une organisation holonique (dépendance entre les groupes).

Les outils d'aide à la décision sont quant à eux destinés à prédire l'évolution de notre système et à déterminer des causes possibles de survenue d'EIAS.

Cette figure met en avant la coopération qui existe entre les deux paradigmes. Ainsi, la détermination de l'état du système dépend du RèPC qui lui même s'enrichit des différents événements produits par notre SMA.

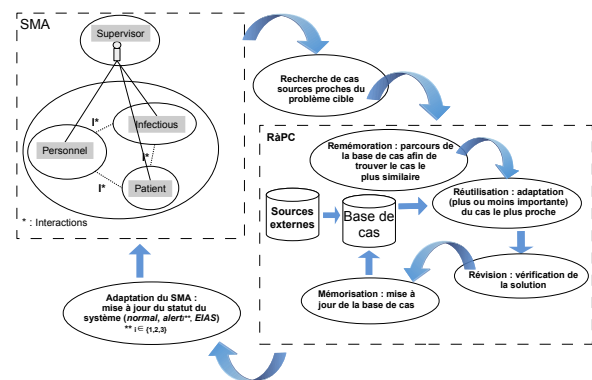


FIGURE 1 – Architecture globale de MASSAI

Acknowledgments

Les auteurs remercient l'Europe et la Région Bourgogne-Franche-Comté pour le soutien financier de ces travaux dans le cadre des Fonds Européens pour le Développement Régional (FEDER).

Références

[1] Suliburk, James W and Buck, Quentin M and Pirko, et Al. Analysis of human performance deficiencies associated with surgical adverse events. *JAMA network open*, 2(7) :e198067–e198067, 2019.

