



AfIA

Association française
pour l'Intelligence Artificielle

JFSMA

Journées Francophones sur les Systèmes Multi-Agents

PFIA 2023



Table des matières

Maxime Morge Éditorial	5
Comité de programme	6
Conférences invitées	7
S. Kraus AI for Explaining Decisions in Multi-Agent Environments	8
O. Boissier Programmation orientée multi-agent et développement de collectifs d'agents autonomes	9
Allocation multi-agents de tâches	10
G. Picard Allocation décentralisée de lots par consensus pour des tâches composites multi-modes	11
E. Beauprez, A.-C. Caron, M. Morge, J.-C. Routier Consommation adaptative par négociation continue	21
Modèles formels multi-agents	31
J. Guéron, G. Bonnet Un concept de solutions avec un biais d'exploration pour les jeux de coalitions stochastiques répétés	32
M. Déhais, B. Mermet, G. Bonnet Vérification formelle de propriétés de vivacité pour des SMA stochastiques à l'aide de GDT ..	42
Jeunes-chercheurs	52
P. Quentel Architecture distribuée de collaboration pour systèmes critiques : application aux systèmes navals	53
J. Soulé, J.-P. Jamont, M. Ocelllo, P. Théron De l'organisation d'un système multi-agent de cyberdéfense	54
J.-B. Ly, Q. Reynaud, C. Le Bail, M. Schumann, V. Boccara, N. Sabouret Modélisation des déterminants de l'activité humaine au sein d'une simulation multi-agent	55
Modélisation de comportements humains	56
J. Vanderlynden, P. Mathieu, R. Warlop Comprendre l'impact des stratégies de prix sur le comportement des consommateurs	57
S. Delarre, F. Eloire, M. Morge, A. Nongaillard Modèle explicatif de la sécession des experts dans les communautés de pratiques	67
A. Kalaitzakis, J. Euzenat À quoi sert la spécialisation en évolution culturelle de la connaissance ?	76
Modèles et outils multi-agents	86
M. Beuret, I. Foucherot, C. Gentil, J. Savelli COBAI : un modèle générique à base d'agents centré sur les contextes et les interactions pour la simulation de comportements	87
T. Balaguer, O. Simonin, I. Guérin Lassous, I. Fantoni	

Etat de l'art sur la co-simulation robotique et réseau des systèmes multi-robots	97
M. Limame, J. Henriet, C. Lang, N. Marilleau	
Gestion des connaissances partagées par des agents à ressources et connectivité limitées : étude, analyse et expérimentation	107
Outils et démonstrations logiciels	117
O. Gillet, E. Daudé, A. Saval, C. Caron, S. Rey-Coyrehourcq, P. Taillandier, P. Tranouez	
ESCAPE : simulation à base d'agents pour l'évacuation de populations lors des situations d'urgence	118
A. Saval, M. Bourgais, E. Daudé, P. Tranouez	
Escape-SG : un jeu sérieux pour mieux préparer les évacuations de masse	128
C. Raïevsky, F. Suro	
Construction incrémentale d'un comportement collectif autour d'une proto-émotion de peur	132
Optimisation et sécurité multi-agents	136
C. Fèvre, H. Zgaya-Biau, P. Mathieu, S. Hammadi	
L'optimisation du covoiturage dynamique multi-saut	137
H. Roussille, Ö. Gürcan, F. Michel	
Taxonomie des vulnérabilités liées aux incitations dans les blockchains	147
R. N'goran, Y. Kermarrec, J.-L. Tetchueng, O. Asseu	
Community-OrBAC : un modèle de contrôle d'accès établi à partir des agents pour les systèmes de collaboration centrés sur la communauté	157

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 les domaines de l'intelligence artificielle distribuée et des Systèmes Multi-Agents (SMA). Ces journées sont un moment privilégié d'échanges scientifiques multidisciplinaires. Elles réunissent des académiques et des industriels qui étudient, manipulent et font évoluer le paradigme multi-agents pour examiner des problématiques issues de domaines liés à l'informatique (intelligence et vie artificielle, génie logiciel, robotique collective, etc.), à l'automatique, aux sciences naturelles (épidémiologie, éthologie, écologie, etc.) et aux sciences humaines et sociales (économie, sociologie, linguistique, etc.).

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), Bordeaux (PFIA virtualisée, 2021) et enfin Saint-Étienne (PFIA, 2022).

Par tradition, chaque édition des JFSMA met en avant une thématique spécifique que les auteurs sont invités à prendre en compte dans leurs contributions, s'ils le souhaitent. Cette année, le thème des journées est « l'explicabilité des système multi-agents ». L'explicabilité d'un dispositif numérique est une propriété essentielle pour que l'humain s'approprie le résultat fourni. Loin des mythes mais pour répondre à des enjeux sociaux (juridique, éthique ou économique), un système de type intelligence artificielle doit être en capacité de soutenir un processus d'explication en fournissant les moyens aux utilisateurs de comprendre les données exploitées, le modèle employé et le résultat produit dans le contexte spécifique de son déploiement pratique. Ce défi est d'autant plus difficile pour un SMA que son audience est variée (concepteurs, simple utilisateurs, thématiciens) et que le résultat produit découle d'entrelacements entre des phénomènes microscopiques et des dynamiques globales.

Cet ouvrage contient les contributions sélectionnées et présentées lors de la trente-et-unième édition des JFSMA qui se déroule à Strasbourg, du 5 au 7 juillet 2023, dans le cadre de la Plate-Forme Intelligence Artificielle (PFIA 2023). Parmi les 22 contributions soumises cette année, le comité de programme des JFSMA'23 a sélectionné 9 présentations longues, 5 présentations courtes et 2 démonstrations. Toutes ces contributions sont regroupées dans le présent ouvrage en 6 parties, correspondant aux 6 sessions de la conférence.

Je tiens à remercier les 72 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'23. Les implications de chacun ont permis l'édition de cet ouvrage et ont contribué à la réussite de cette édition.

Maxime Morge

Comité de programme

Présidence

- Maxime Morge (CRISAL, Université de Lille, France).

Comité consultatif

- Valérie Camps (IRIT, Université de Toulouse, France) ;
- Jean-Paul Jamont (LCIS, Université Grenoble Alpes, France) ;
- Gauthier Picard (DTIS - SYD, ONERA, France) ;
- Nicolas Sabouret (LISN, Université Paris-Saclay, France) ;
- Olivier Simonin (CITI lab, INSA Lyon, France).

Membres

- Emmanuel Adam (LAMIH, Université Polytechnique Hauts-De-France, 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, Toulouse) ;
- Florent Carlier (CREN, Le Mans Université, France) ;
- Anne-Cécile Caron (CRISAL, Université de Lille, France) ;
- François Charpillat (LORIA, INRIA, Nancy, France) ;
- Vincent Chevrier (LORIA, Université de Lorraine, France) ;
- Rémy Courdier (LIM, Université de la Réunion, France) ;
- Stéphane Galland (CIAD, Université de Technologie Belfort-Montbéliard, France) ;
- Sylvain Giroux (DOMUS, Université de Sherbrooke, Canada) ;
- Zahia Guessoum (CRESTIC, Université de Reims, France) ;
- Cédric Herpson (LIP6, Sorbonne Université, France) ;
- Guillaume Hutzler (IBISC, Université Evry-Paris Saclay, France) ;
- Elsy Kaddoum (IRIT, Université de Toulouse, 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) ;
- Philippe Mathieu (CRISAL, Université de Lille, France) ;
- Bruno Mermet (GREYC, Université Caen Normandie, France) ;
- Fabien Michel (LIRMM, Université de Montpellier, France) ;
- Frédéric Migeon (IRIT, Université de Toulouse, France) ;
- Gildas Morvan (LGI2A, Université d'Artois, France) ;
- Guillaume Muller (Institut Henri Fayol, École des Mines de Saint-Etienne, France) ;
- Jean-Pierre Müller (SENS, CIRAD, Université de Montpellier, France) ;
- Antoine Nongaillard (CRISAL, Université de Lille, France) ;
- Michel Occello (LCIS, Université Grenoble Alpes, France) ;
- Alexandre Pauchet (LITIS, INSA Rouen Normandie, France) ;
- Sébastien Picault (INRAE, UMR BIOEPAR, France) ;
- Suzanne Pinson (LAMSADE, Université Paris Dauphine, France) ;
- Manon Prédhumeau (School of Geography, University of Leeds, UK) ;
- Clément Raïevsky (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) ;
- Patrick Taillandier (COGIT, IGN, France) ;
- Laurent Vercouter (LITIS, INSA Rouen Normandie, France).

Conférences invitées

AI for Explaining Decisions in Multi-Agent Environments

Sarit Kraus
sarit@cs.biu.ac.il

Université Bar-Ilan, Israël

Résumé

Explanations are necessary for humans to understand and accept decisions made by an AI system when the system's goal is known. It is even more important when the AI system makes decisions in multi-agent environments where humans do not know the systems' goals since they may depend on other agents' preferences. In such situations, explanations should aim to increase user satisfaction, considering the system's decision, the user's and the other agents' preferences, the environment settings, and properties such as fairness, envy, and privacy. We will discuss three cases of Explainable decisions in Multi-Agent Environments (xMASE) : explanations for multi-agent Reinforcement Learning, advice explanations in complex repeated decision-making environments and explaining preferences and constraints-driven optimization problems. For each case, we will present an algorithm for generating explanations and report on human experiments that demonstrate the benefits of providing the resulting explanations for increasing human satisfaction from the AI system.



Biographie

Sarit Kraus (Ph.D. Computer Science, Hebrew University, 1989) is a Professor of Computer Science at Bar-Ilan University. Her research is focused on intelligent agents and multi-agent systems integrating machine-learning techniques with optimization and game theory methods. For her work, she received many prestigious awards. She was awarded the IJCAI Computers and Thought Award, the ACM SIGART Agents Research award, the ACM Athena Lecturer, the EMET prize and was twice the winner of the IFAAMAS influential paper award. She is an ACM, AAI and EurAI fellow and a recipient of the advanced ERC grant. She also received a special commendation from the city of Los Angeles. She is an elected member of the Israel Academy of Sciences and Humanities.

Sarit Kraus is the the winner of the IJCAI-23 Award for Research Excellence. She is recognized for her pioneering work of the study of interactions among self-interested agents, creating the field of automated negotiation, and developing methods for coalition formation and teamwork, both as formal models and real-world implementations.

Programmation orientée multi-agent et développement de collectifs d'agents autonomes

Olivier Boissier
olivier.boissier@emse.fr

LIMOS, École des Mines de Saint-Etienne, France

Résumé

Le domaine des systèmes multi-agents (SMA) propose des modèles, langages et technologies pour l'ingénierie et la programmation de collectifs d'agents autonomes situés dans un environnement dans lequel ils interagissent entre eux ou par le biais de ressources partagés. Parmi ces approches, nous proposons la programmation orientée multi-agent (MAOP) de SMA, intégrant de manière cohérente et transparente les concepts et primitives nécessaires à la programmation d'environnements, d'agents autonomes, de leurs interactions et organisations. Nous présenterons le modèle MAOP à la base de cette approche de programmation et illustrerons au travers de différentes applications comment elle est utilisée pour l'intégration et l'ingénierie de systèmes intelligents décentralisés au travers de sa mise en oeuvre dans la plateforme JaCaMo. Nous présenterons ensuite comment ce modèle MAOP sert de base actuellement au développement de communautés hybrides sur le Web, i.e. de communautés d'agents autonomes et d'humains interagissant entre eux et avec les ressources déployées dans cet environnement large échelle et ouvert, porteur de nombreux défis et d'opportunités.



Biographie

Olivier Boissier est professeur titulaire, directeur de l'Institut Henri Fayol, l'un des cinq centres de formation et de recherche de Mines Saint-Etienne. Co-animateur de la communauté scientifique Data&AI de l'Institut Mines Télécom, il est également membre du LIMOS UMR CNRS 6158. Il a soutenu son doctorat à l'Institut national polytechnique de Grenoble en 1993 et son Habilitation à Diriger des Recherches en 2003 à l'Université Jean Monnet de Saint-Etienne. Olivier Boissier participe au comité de programme de nombreuses conférences (IJCAI, AAAI, IAT/WI, AAMAS) et ateliers dans le domaine de l'intelligence artificielle et des systèmes multi-agents (COINS, EUMAS, EMAS). Il a été l'un des initiateurs de la série d'ateliers COIN (Coordination, Organisations, Institutions et Normes dans les Systèmes Multi-Agents). Il a présidé le programme de la conférence IAT/WI en 2011. Ses principales contributions de recherche concernent la définition de modèles multi-agents, d'outils et de méthodologies pour l'ingénierie orientée multi-agent d'applications logicielles sur le Web ou l'IoT, dans les domaines de l'industrie et de la ville intelligentes. Ses contributions incluent également le développement d'architectures d'agents intégrant des considérations éthiques, de modèles de coordination basés sur des organisations normatives. Il a notamment participé au projet national français ForTrust (Formalisation de la confiance dans les systèmes multi-agents), ETHICAA (Éthique des agents autonomes) et est coordinateur du projet bilatéral franco-suisse HYPERAGENT (systèmes multi-agents hypermédias pour les communautés hybrides sur le Web).

Allocation multi-agents de tâches

Allocation décentralisée de lots par consensus pour des tâches composites multi-modes

Gauthier Picard
gauthier.picard@onera.fr

ONERA/DTIS, Université de Toulouse, France

Résumé

Nous considérons des agents planifiant des tâches à exécuter dans des créneaux temporels qu'ils possèdent sur des ressources disjonctives. Des utilisateurs non propriétaires souhaitent programmer des tâches composites multi-modes nécessitant l'accès à un ou plusieurs créneaux privés, dont les propriétaires doivent se coordonner pour répondre collectivement aux requêtes, sans divulguer leurs propres plans. Nous adoptons une approche d'allocation de tâches basée consensus, MM-CBGA, adaptée à ce cadre. La contribution est évaluée et comparée à des allocations centralisées gloutonnes et à des enchères séquentielles classiques, en utilisant des constellations d'observation de la Terre simulées et des carnets de commande réalistes pour des observations sur l'Europe.

Mots-clés : Allocation de tâches, consensus, observation de la Terre

Abstract

We consider agents scheduling tasks to be executed in time slots they own on disjunctive resources. Non-owner users wish to schedule multi-mode composite tasks requiring access to one or more private slots, whose owners must coordinate to collectively respond to requests, without disclosing their own plans. We adopt a consensus-based task allocation approach, MM-CBGA, adapted to this framework. The contribution is evaluated and compared to greedy centralized allocations and classical sequential auctions, using simulated Earth observation constellations and realistic order books for observations over Europe.

Keywords: task allocation, consensus, Earth observation

1 Introduction

Le déploiement de constellations de plus grandes tailles pour les tâches d'observation de la Terre nécessite la résolution de problèmes de planification et d'ordonnancement de plus en plus complexes. En effet, si un plus grand nombre de satellites augmente linéairement le nombre d'opportunités de satisfaire les demandes des utilisateurs à une fréquence plus élevée, cela augmente aussi exponentiellement la

taille des problèmes d'ordonnancement et d'allocation de tâches qui en résultent [21]. De plus, la résolution de tels problèmes doit être effectuée plusieurs fois par jour, en moins de quelques minutes, afin d'adapter les plans aux conditions météorologiques, ou aux demandes de dernière minute, et afin que les plans soient envoyés aux satellites dès que possible. Pour rendre la situation encore plus complexe, certaines constellations envisagent désormais des concepts d'exploitation où certains utilisateurs possèdent certains créneaux orbitaux pour une longue période [19, 15, 16]. Les propriétaires gèrent entièrement les plans dans leurs créneaux orbitaux et les gardent privés. Tout utilisateur non propriétaire souhaitant effectuer certaines observations nécessitant l'accès à des créneaux orbitaux privés doit alors faire une demande aux propriétaires, qui doivent se coordonner puis décider d'accepter ou non ces tâches. Ce problème revient à trouver une allocation de tâches aux créneaux privés, afin que les demandes soient satisfaites de la meilleure manière, tout en minimisant leur impact sur les plans privés et sans les divulguer.

Ce problème entre dans le cadre de l'allocation multi-agents de tâches [7, 17]. Parmi les méthodes efficaces, les techniques basées enchères ont fait l'objet d'une attention particulière, et ont été utilisées dans les domaines de la robotique collective [11] et de l'Espace [13, 18]. Ici, les agents mettent en œuvre des enchères pour répartir les tâches de manière coopérative et distribuée. Les agents font des offres pour des tâches uniques ou des lots de tâches qu'ils veulent insérer dans leurs plans, et les gagnants sont déterminés en fonction de la valeur des offres, de manière centralisée (e.g., enchères combinatoires, PSI, SSI [12]) ou décentralisée (e.g., CBBA [2]). Dans le domaine de l'allocation de tâches d'observation, Picard a fourni des modèles et des algorithmes pour des scénarios spécifiques à l'observation de la Terre avec des créneaux privés [19], où chaque mode ne contenait qu'une seule tâche. Dans une direction légèrement différente, Phillips et Parra étudient l'utilisation de l'allocation de lots basée sur le consensus entre les satellites eux-mêmes, mais sans aucune confidentialité [18]. Enfin, Lee et al. considèrent plusieurs constellations appartenant à différents agents et proposent d'utiliser des protocoles de consensus

pour allouer les tâches au sein de ces constellations [13]. Dans tous ces modèles et algorithmes, les demandes composites ne sont pas prises en compte. Les tâches ne sont pas interdépendantes ou ne font pas partie d'une tâche composite : les agents ne se soucient que de la cohérence et de la valeur de leur lot, et non de certaines contraintes sur l'ensemble des tâches atomiques de la même tâche composite. Pour répondre à de telles interdépendances d'éléments, on peut s'appuyer sur une détermination centralisée du gagnant (ou *Winner Determination Problem*, WDP), garantissant que les contraintes ou les critères sur les tâches composites sont optimisés [14]. Dans des contextes plus distribués, tels que l'allocation de tâches par consensus (e.g., CBAA ou CBBA), où le WDP est distribuée, la capture des interdépendances entre les tâches n'est pas souvent prise en compte. Cependant, Hunt et al. ont proposé une extension de CBBA, appelée CBGA, pour traiter les tâches multi-agents, c'est-à-dire les tâches nécessitant la participation de plusieurs agents ; mais là encore, un seul mode a été considéré pour chacune de ces tâches.

Nous proposons ici une extension de CBGA pour faire face aux tâches composites multi-modes, et l'appliquons à un scénario d'observation de la Terre avec de multiples propriétaires de créneaux orbitaux. Nos contributions sont les suivantes : (i) nous modélisons le problème d'allocation de tâches composites multi-mode multi-agent (MACTA) ; (ii) nous modélisons MACTA comme un programme mathématique, et proposons un algorithme glouton pour le résoudre ; (iii) nous concevons un solveur distribué pour MACTA (MM-CBGA) ; (iv) nous faisons correspondre une version étendue du problème d'ordonnancement de constellations de satellites d'observation de la Terre [19] à MACTA, en considérant des tâches composites au lieu de tâches monoscopiques ; (v) nous évaluons les performances de MM-CBGA sur des carnets de commande réalistes et un simulateur de constellation.

Cet article est une traduction d'un article accepté à AAMAS'23 [20].

2 Modèle et notations

Cette section présente le problème d'allocation de tâches composites multi-agents abordé ici.

Définition 1. Un *créneau* $w = [s_w, e_w]$ est une fenêtre temporelle, où $s_w \in [t_{\min}, t_{\max}]$, $e_w \in [t_{\min}, t_{\max}]$, $s_w < e_w$, et $t_{\min} < t_{\max} \in \mathbb{R}_{\geq 0}$.

Soit \mathcal{T} un ensemble de tâches à exécuter et \mathcal{R} un ensemble de ressources *disjonctives*, i.e. qui ne peuvent être utilisées que pour une seule tâche à la fois. Dans notre article, les tâches seront des tâches d'observation à effectuer sur des satellites, qui ne peuvent effectuer qu'une seule observation à la fois.

Définition 2. Une *tâche* $\tau \in \mathcal{T}$ est un tuple $\langle w_\tau, d_\tau, r_\tau, \omega_\tau \rangle$ où $w_\tau = [s_\tau, e_\tau]$ est le créneau dans lequel τ doit être planifiée, $d_\tau \in \mathbb{R}_{\geq 0}$ est la durée de τ , $r_\tau \in \mathcal{R}$ est une ressource sur laquelle τ doit être exécutée, et $\omega_\tau \in \mathbb{R}_{\geq 0}$ est la récompense reçue pour avoir planifié τ .

Définition 3. Un *plan* est un ensemble $\pi = \{(\tau, t) \mid \tau \in \mathcal{T}, t \in [t_{\min}, t_{\max}]\}$ définissant le temps de début t de chaque tâche τ , tel qu'il n'y a aucun chevauchement entre tâches sur la même ressource : $\forall (\tau, t), (\tau', t'), [t, t+d_\tau] \cap [t', t'+d_{\tau'}] = \emptyset$.

Soit $\mathring{\mathcal{T}}$ l'ensemble des requêtes à réaliser. Dans notre étude, certaines requêtes sont émises par des clients non propriétaires et nécessitent l'accès à des créneaux privés.

Définition 4. Une *requête* (ou *tâche composite*) $\hat{\tau} \in \mathring{\mathcal{T}}$ est un tuple $\langle M_{\hat{\tau}}, \oplus_{\hat{\tau}} \rangle$ où $M_{\hat{\tau}} \subset 2^{\mathcal{T}}$ est l'ensemble d'ensembles de tâches pouvant compléter la requête (i.e. l'ensemble de *modes*), et $\oplus_{\hat{\tau}} : M_{\hat{\tau}} \rightarrow \mathbb{R}_{\geq 0}$ est une fonction d'agrégation qui calcule la récompense associée à chaque mode.

La notion de *mode* est courante dans la gestion de projet à contraintes de ressources (RCPSP) [3], et est même utilisée dans le domaine de l'observation de la Terre [21]. Il est pratique de modéliser les interdépendances entre les tâches atomiques pour répondre aux requêtes. La fonction d'agrégation $\oplus_{\hat{\tau}}$ peut être n'importe quelle fonction monotone croissante telle que la somme (+), ou une fonction linéaire avec un certain facteur d'actualisation pour toute tâche après la première¹. Une requête est satisfaite si l'un de ses modes est entièrement planifié, i.e. toutes les tâches d'un mode sont planifiées.

Définition 5. Pour un mode m d'une requête $\hat{\tau}$, la *récompense de mode* est l'agrégation des récompenses des tâches composant le mode : $\omega_m \stackrel{\text{def}}{=} \oplus_{\hat{\tau}}(m)$.

Pour une tâche donnée τ , on note $\text{request}(\tau)$ (resp. $\text{mode}(\tau)$) la requête (resp. le mode) à laquelle τ contribue. On note également $\text{modes}(\hat{\tau})$ l'ensemble de tous les modes pour la requête $\hat{\tau}$, et par extension nous notons $\text{modes}(T) = \bigcup_{\hat{\tau}_i \in T} \text{modes}(\hat{\tau}_i)$.

Définition 6. Pour un plan π , la *récompense de plan*, notée ω_π , est la somme des récompenses des modes planifiés :

$$\omega_\pi \stackrel{\text{def}}{=} \sum_{m \in \{\text{mode}(\tau) \mid \tau \in \pi\}} \omega_m$$

1. Pour des raisons de lisibilité et de simplicité, nous utiliserons la somme simple (+ et \sum), au lieu de l'opérateur spécifique à la requête ($\oplus_{\hat{\tau}}$ et $\oplus_{\hat{\tau}}$) dans le reste de l'article.

Soit \mathcal{A} un ensemble d'agents capables d'effectuer des tâches. Dans nos expériences, les agents seront des propriétaires de créneaux privés, qui peuvent recevoir des demandes d'insertion de certaines observations dans leurs créneaux privés.

Définition 7. Un agent $i \in \mathcal{A}$ est un tuple $\langle O_i, \mathring{\mathcal{T}}_i^-, \mathring{\mathcal{T}}_i^+, \pi_i \rangle$ où $O_i = \{(r, w) \mid r \in \mathcal{R}, w = [s_w, e_w]\}$ est un ensemble de créneaux privés sur des ressources, $\mathring{\mathcal{T}}_i^- \subset \mathring{\mathcal{T}}$ est un ensemble de requêtes privées à positionner dans ses créneaux privés, $\mathring{\mathcal{T}}_i^+ \subset \mathring{\mathcal{T}}$ est un ensemble de requêtes externes pour lesquelles i est sollicité, et π_i est le plan courant de i qui affecte un temps de début à certaines tâches privées ou externes.

Notez qu'il n'y a pas de copropriété des créneaux, c-à-d. que pour la même ressource r , il n'y a pas de chevauchement entre les créneaux privés. De plus, nous supposons que les fenêtres des tâches privées sont entièrement incluses dans les créneaux privés. Nous notons également que π_i est divisé en deux sous-parties : π_i^- pour les tâches privées, et $\pi_i^+ = \pi_i \setminus \pi_i^-$ pour les tâches externes. Nous identifions les tâches qu'un agent peut exécuter, c-à-d. dont les fenêtres temporelles sont incluses dans certains de ses créneaux privés, en utilisant $\text{can} : \mathcal{A} \rightarrow 2^{\mathring{\mathcal{T}}}$. Par extension, nous utilisons également can pour les tâches composites, si au moins une tâche peut être programmée dans l'un des créneaux privés de l'agent.

Définition 8. Un problème d'allocation de tâches composites multi-modes multi-agents (ou MACTA) est un tuple $P = \langle \mathcal{A}, \mathcal{R}, \mathring{\mathcal{T}} \rangle$, défini par un ensemble d'agents \mathcal{A} , un ensemble de ressources disjonctives \mathcal{R} , et un ensemble de requêtes $\mathring{\mathcal{T}}$ (privées ou non), et vise à trouver l'allocation de tâches atomiques aux agents, qui maximise la somme des récompenses des requêtes satisfaites tout en respectant les contraintes de disjonction entre les ressources : $\max \sum_{\tau \in \mathring{\mathcal{T}}} \omega_{\tau}$.

Pour un MACTA P donné, on note $\mathring{\mathcal{T}}^+ \stackrel{\text{def}}{=} \bigcup_{i \in \mathcal{A}} \{\mathring{\mathcal{T}}_i^+\}$ l'ensemble des tâches qui sont privées à aucun propriétaire. Pour un agent donné i , nous notons P_i^- le problème restreint à $\mathring{\mathcal{T}}_i^-$, afin de calculer son plan initial π_i ou de le réviser, compte tenu de certaines tâches supplémentaires. Nous notons également $P_i^- \cup M$ le problème consistant à résoudre le problème privé P_i^- avec des modes supplémentaires contenus dans l'ensemble M , où chaque mode est filtré de façon à ne considérer que les tâches situées dans les créneaux de i . Il sera utilisé pour évaluer le coût de l'insertion des modes dans les plans.

Notez que, bien que les agents puissent avoir des plans privés initiaux, nous considérons que ces

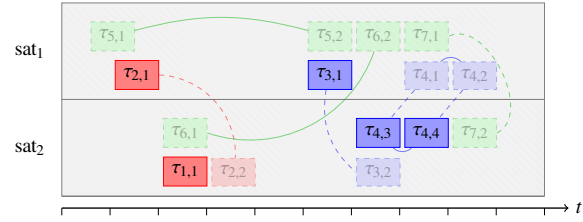


FIGURE 1 – Un exemple de MACTA. Les créneaux privés sont hachurés en rouge (u_1) ou en bleu (u_2). Les tâches sont représentées par des rectangles colorés pleins lorsque planifiées, en pointillés lorsque non planifiées. Les tâches d'une même requête sont liées par des lignes pleines pour le même mode, ou par des pointillées pour des modes distincts.

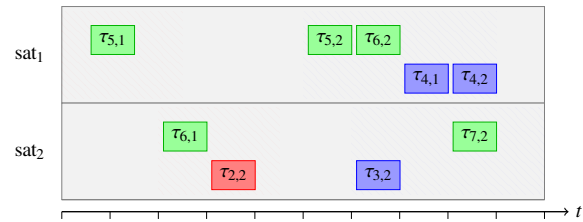


FIGURE 2 – Une solution pour l'exemple 1.

plans peuvent être révisés afin d'augmenter la récompense collective. Cependant, si certains agents sont réticents à réviser leur plan, ils peuvent soit toujours répondre "non" à toute demande chevauchant leurs créneaux, soit augmenter la récompense de leurs propres tâches et demandes pour garantir leur présence dans le plan final.

Exemple 1. La figure 1 illustre un MACTA simple. Nous considérons l'horizon $[t_{\min}, t_{\max}] = [0, 10]$, et une constellation de deux satellites, $\mathcal{R} = \{\text{sat}_1, \text{sat}_2\}$. Deux agents, $\mathcal{A} = \{u_1, u_2\}$, possèdent deux créneaux chacun, émettent deux requêtes privées, et ont déjà calculé leurs plans initiaux. Ainsi, u_1 a émis τ_1 avec un unique mode $\{\tau_{1,1}\}$, et τ_2 avec deux modes $\{\tau_{2,1}\}$ et $\{\tau_{2,2}\}$. u_2 a émis deux requêtes, τ_3 avec deux modes $\{\tau_{3,1}\}$ et $\{\tau_{3,2}\}$, et τ_4 avec deux modes $\{\tau_{4,1}, \tau_{4,2}\}$ et $\{\tau_{4,3}, \tau_{4,4}\}$. En particulier, la requête "tout-ou-rien" τ_4 nécessite de planifier deux tâches, ou aucune.

Considérons les récompenses suivantes $\omega_{\tau_{1,1}} = \omega_{\tau_{2,1}} = \omega_{\tau_{4,3}} = \omega_{\tau_{4,4}} = 10$, $\omega_{\tau_{2,2}} = \omega_{\tau_{3,1}} = \omega_{\tau_{3,2}} = \omega_{\tau_{4,1}} = 8$, $\omega_{\tau_{4,2}} = 6$. En utilisant la fonction d'agrégation +, les meilleurs plans privés sont représentés dans la figure 1, sélectionnant les meilleurs modes pour chaque requête, et avec une récompense globale de 48.

Maintenant, lorsque des requêtes externes sont émises, les plans privés peuvent changer radicalement afin d'augmenter la récompense globale. Considérons trois nouvelles requêtes (en vert sur les figures) émises par un utilisateur externe, avec

toutes la même récompense de tâche atomique de 6 : τ_5 avec un mode $\{\tau_{5,1}, \tau_{5,2}\}$, τ_6 avec un mode $\{\tau_{6,1}, \tau_{6,2}\}$, et τ_7 avec deux modes $\{\tau_{7,1}\}$ et $\{\tau_{7,2}\}$. Sans réviser les plans initiaux de u_1 et u_2 , la seule demande externe satisfaite est τ_7 , en plaçant $\tau_{7,2}$, ce qui entraîne une récompense globale de $48+6=54$.

u_1 gagne à supprimer la tâche $\tau_{1,1}$ au profit de $\tau_{6,1}$ et $\tau_{6,2}$, pour un gain de $6+6-10=2$, mais nécessite de se coordonner avec u_2 qui doit insérer $\tau_{6,2}$ dans son plan. u_1 remplace $\tau_{2,1}$ par $\tau_{2,2}$, laissant de la place pour $\tau_{5,1}$, et nécessitant de planifier $\tau_{5,2}$, ce qui oblige u_2 à remplacer $\tau_{3,1}$ par $\tau_{3,2}$, ce qui génère finalement un gain de $6+6-10+8-8=2$. La récompense du plan solution illustré dans la figure 2, qui est optimale, est 60.

3 Approches centralisées

Une approche classique pour résoudre de manière centralisée un problème d'allocation consiste à utiliser un programme mathématique – idéalement linéaire – afin de bénéficier de solveurs efficaces disponibles, tels qu'IBM CPLEX [10] ou Gurobi [8]. Il est également possible de s'appuyer sur des algorithmes gloutons, afin de passer à l'échelle. Notez que ces solutions ne respectent évidemment pas la confidentialité : les agents doivent envoyer toutes leurs données à une autorité centrale chargée de calculer les horaires. Cependant, elles représentent de bonnes bases de comparaison pour évaluer la qualité des solutions obtenues de manière décentralisée, et elles peuvent être utilisées par les agents pour calculer ou évaluer des plans privés.

Approche optimale. Nous formulons MACTA comme un programme linéaire mixte en nombres entiers (MILP). Considérons les variables de décision suivantes : $x_\tau \in \{0,1\}$ une variable binaire indiquant si la tâche τ est planifiée ; $y_m \in \{0,1\}$ une variable binaire indiquant si le mode m est sélectionné pour répondre à une requête τ ; $t_\tau \in [s_{w_\tau}, e_{w_\tau} - d_\tau]$ une variable continue indiquant la date de début de τ ; et $\beta_{\tau,\tau'} \in \{0,1\}$ une variable binaire indiquant si τ précède τ' . Nous notons $(\tau, \tau') \in \mathcal{T}_\cap^2$ les paires de tâches distinctes sur la même ressource avec des fenêtres de temps qui se chevauchent.

$$\max_{y_m} \sum_{\tau \in \mathcal{T}} \sum_{m \in M_\tau} \omega_m y_m \quad (1)$$

$$\text{t.q.} \quad \sum_{\tau \in m} x_\tau \geq |m| y_m, \quad \forall \tau \in \mathcal{T}, \forall m \in M_\tau \quad (2)$$

$$\sum_{m \in M_\tau} y_m \leq 1, \quad \forall \tau \in \mathcal{T} \quad (3)$$

$$2 - \beta_{\tau,\tau'} - \beta_{\tau',\tau} \leq x_\tau, \quad \forall (\tau, \tau') \in \mathcal{T}_\cap^2 \quad (4)$$

$$2 - \beta_{\tau,\tau'} - \beta_{\tau',\tau} \leq x_{\tau'}, \quad \forall (\tau, \tau') \in \mathcal{T}_\cap^2 \quad (5)$$

$$\beta_{\tau,\tau'} + \beta_{\tau',\tau} \leq 3 - x_\tau - x_{\tau'}, \quad \forall (\tau, \tau') \in \mathcal{T}_\cap^2 \quad (6)$$

$$t_\tau - t_{\tau'} \geq d_\tau - \Delta_{\tau,\tau'}^{\max} \beta_{\tau,\tau'}, \quad \forall (\tau, \tau') \in \mathcal{T}_\cap^2 \quad (7)$$

$$t_{\tau'} - t_\tau \geq d_{\tau'} - \Delta_{\tau',\tau}^{\max} \beta_{\tau',\tau}, \quad \forall (\tau, \tau') \in \mathcal{T}_\cap^2 \quad (8)$$

avec $\Delta_{\tau,\tau'}^{\max} = e_\tau - s_{\tau'} + d_\tau$ une valeur servant de constante *big-M* pour déclencher des contraintes conditionnées par la précédence entre deux tâches. Ce MILP vise à maximiser la récompense des modes actifs, comme indiqué dans l'équation (1). La contrainte (2) relie les x et les y en indiquant que si un mode est sélectionné, toutes ses tâches doivent être programmées. La contrainte (3) oblige à sélectionner au maximum un mode par requête. Les contraintes (4) à (6) assurent la cohérence des variables de précédence, tandis que les contraintes (7) et (8) garantissent que des tâches distinctes sur la même ressource ne se chevauchent pas. Malheureusement, un tel programme, compte tenu de la présence de variables entières et binaires potentiellement nombreuses, ne passe pas à l'échelle. Il faut donc envisager une approche non optimale, telle que l'allocation gloutonne, qui a montré de bonnes performances sur des problèmes similaires d'allocation de tâches [1, 22].

Approche gloutonne. L'algorithme glouton, présenté dans l'algorithme 1, trie d'abord les modes par ordre décroissant de récompense (ligne 4). Pour chaque mode, il tente de trouver des créneaux pour toutes les tâches du mode (lignes 5-11). Si toutes les tâches peuvent être programmées, le mode est sélectionné et toutes les tâches sont programmées (lignes 12-14). Sinon, les créneaux réussis sont supprimés du plan des ressources. Pour trouver un créneau, la fonction `first` recherche simplement le premier créneau disponible sur le plan des ressources, compatible avec la fenêtre temporelle de la tâche, sans chevauchement avec une tâche déjà planifiée. Cet algorithme n'est pas optimal, mais fournit des solutions très rapidement, puisqu'il est polynomial en nombre de modes et de tâches. Mais, comme pour le MILP, il nécessite de partager toutes les contraintes et informations avec une autorité centrale de planification. Cet algorithme sera notre référence pour les expérimentations, puisque les algorithmes gloutons sont largement utilisés dans la planification de missions spatiales, par les opérateurs de constellations, et parce que le solveur MILP n'est pas adapté même pour les plus petites instances que nous considérons.

4 Approche décentralisée

Cette section expose la principale contribution de l'article : l'algorithme MM-CBGA basé consensus pour l'allocation de tâches composites à modes multiples. Auparavant, nous rappelons brièvement le contexte de l'allocation de tâches par enchères.

Algorithme 1 : Solveur glouton**Données :** Un MACTA $P = \langle \mathcal{A}, \mathcal{R}, \hat{\mathcal{T}} \rangle$ **Résultat :** Un plan π

```

1  $\pi \leftarrow \{ \}$  // le plan à construire
2  $R \leftarrow \{ (r, []) \mid r \in \mathcal{R} \}$  // les plans des ressources
3  $S \leftarrow \{ \}$  // les requêtes remplies
4  $M \leftarrow \text{sort}(\text{modes}(\hat{\mathcal{T}}))$  // les modes triés
5 pour chaque  $m \in M$  faire
6   si  $\text{request}(m) \notin S$  alors // requête non remplie
7      $T \leftarrow \{ \}$  // les créneaux trouvés jusqu'à présent
8     pour chaque  $\tau \in m$  faire
9        $t \leftarrow \text{first}(\tau, P, R)$ 
10      si  $t \neq \emptyset$  alors // créneau trouvé pour la tâche
11         $T \leftarrow T \cup \{ t \}$ 
12      si  $|T| = |m|$ 
13        alors // tous les créneaux trouvés pour le mode
14           $\pi \leftarrow \pi \cup \{ (\tau, t) \mid t \in T \}$ 
15           $S \leftarrow S \cup \{ \text{request}(m) \}$ 
16        sinon // pas de place pour toutes les tâches
17           $\text{remove}(T, R)$ 
18          // supprimer les créneaux explorés
19 retourner  $\pi$ 

```

4.1 Allocation de tâches basée enchères

Le cadre classique d'allocation de tâches consiste en un ensemble de ressources et un ensemble de tâches à exécuter sur ces ressources. L'objectif est d'assigner des tâches aux ressources de manière à optimiser certains critères (e.g., le nombre de tâches assignées, la somme des récompenses des tâches créneaux, la durée totale, etc.). Il s'agit d'un problème d'allocation classique qui peut être modélisé comme un MILP. L'idée est que les tâches à planifier sont ouvertes aux enchères par un *commissaire-priseur*. Les agents *enchérisseurs* évaluent les tâches en fonction de leur plan actuel, et envoient leurs offres pour certaines de ces tâches. Ensuite, le commissaire-priseur détermine les gagnants en fonction de leurs offres et des contraintes sur les ressources. Ici, les opérations les plus coûteuses sont l'étape d'enchérissement effectuée par chaque enchérisseur, qui peut avoir un nombre exponentiel de lots à évaluer, et le problème de détermination du gagnant (WDP) effectué par le commissaire-priseur, qui revient à résoudre un programme linéaire en nombres entiers de taille potentiellement exponentielle, et qui entre dans le cadre des enchères combinatoires (CA) [4].

Si l'on se réfère à la littérature sur l'allocation de tâches multi-robots [5, 11] et sur l'allocation d'observations multi-satellites [18, 13, 19], ces limites de calcul peuvent être surmontées en utilisant la relaxation classique consistant à n'autoriser les enchères que sur des articles individuels (et non sur des lots). Lorsque les enchérisseurs enchérisent sur l'ensemble des tâches en parallèle, nous nous situons dans le cadre du système PSI (*Parallel*

Single Item) [12]. Lorsque le commissaire-priseur annonce les tâches une par une, et que les enchérisseurs construisent leur offre en connaissant les allocations de tâches précédentes, nous entrons dans le cadre du *Sequential Single Item* (SSI) [12]. En général, SSI a de très bonnes performances avec un temps de calcul très limité, alors que la qualité des solutions PSI est souvent limitée, puisque les enchérisseurs ne peuvent pas facilement raisonner sur les lots. Plus récemment, l'algorithme d'allocation basé sur le consensus (CBBA) combine les idées des enchères et du consensus pour converger plus rapidement (en tours) que SSI tout en produisant des solutions similaires et en ayant les avantages des algorithmes de consensus traditionnels [2]. CBBA est une solution entièrement distribuée pour mettre en œuvre une variante d'enchères combinatoires à faible coût de calcul. Il suit une séquence répétée en deux phases. Tout d'abord, pendant la *phase d'enchérissement*, chaque agent construit un lot unique de tâches qu'il souhaite se voir attribuer, en respectant le coût marginal associé à l'inclusion de l'élément considéré dans son plan actuel. Ensuite, pendant la *phase de consensus*, les agents comparent leurs offres avec celles de leurs voisins. Si un agent est battu sur une tâche t , il abandonne la tâche et toutes les tâches ajoutées après elle dans le plan, car l'exclusion de t a rendu l'évaluation de leur coût marginal obsolète. Cet algorithme a été largement étudié et modifié pour améliorer ses performances et l'adapter à des scénarios spécifiques, comme l'allocation d'observations multi-satellites [18, 13, 19]. L'extension sur laquelle nous nous concentrons dans cet article est CBGA, *Consensus-based Group Auction* [9]. Ici, certaines tâches multi-agents nécessitent la participation de plusieurs agents pour être réalisées. Lorsqu'un nombre suffisant d'agents enchérisse pour une tâche, celle-ci peut être ajoutée aux lots. Pour ce faire, CBGA diffère de CBBA sur deux points principaux : (i) les structures de données d'offre stockent les agents qui ont fait une offre pour chaque tâche, au lieu de ne stocker que la meilleure valeur d'offre jusqu'à présent, et (ii) pendant le consensus, les récompenses des tâches sont calculées sur la base de la somme des meilleures offres des agents pour effectuer la tâche, en écartant les plus mauvaises lorsque plus d'agents que nécessaire ont fait une offre. Cependant, ni CBBA ni CBGA ne peuvent résoudre notre problème, car MACTA permet que certaines tâches composites soient exécutées de différentes manières (modes). Nous étendons donc CBGA au cadre des tâches composites multi-modes, dans la section suivante.

4.2 MM-CBGA

MM-CBGA suit la même logique que CBBA et CBGA, tel que décrit dans l'algorithme 2. Tout d'abord, les agents résolvent leurs problèmes privés,

Algorithme 2 : Solveur MM-CBGA

Données : Un MACTA $P = \langle \mathcal{A}, \mathcal{R}, \hat{\tau} \rangle$
Résultat : Un plan π^+ pour les requêtes externes,
 et un plan privé π_i^- pour chaque agent $i \in \mathcal{A}$

```

1 pour chaque  $i \in \mathcal{A}$  faire en parallèle
2    $\pi_i^- \leftarrow \text{solve}(P_i)$ 
3    $\mathcal{N}_i \leftarrow \{j \in \mathcal{A} \mid j \neq i, \text{can}(j) \cap \text{can}(i) \neq \emptyset\}$ 
4 tant que conflit faire
5   pour
6     chaque  $i \in \mathcal{A}$  faire en parallèle // Phase de bidding
7        $b_i, c_i \leftarrow \text{bid}(i)$  // voir Alg. 3
8       pour chaque  $j \in \mathcal{N}_i$  faire  $\text{send}(\langle b_i, c_i, s_i \rangle, j)$ 
9     pour chaque
10       $i \in \mathcal{A}$  faire en parallèle // Phase de consensus
11        pour chaque  $b_j$  reçue faire
12           $\text{consensus}(b_i, c_i, b_j, c_j)$  // voir Alg. 4
13 pour chaque  $i \in \mathcal{A}$  faire en parallèle  $\text{send}(\pi_i^+, \text{client})$ 
14 retourner  $\bigcup_{i \in \mathcal{A}} \pi_i^+$ 
    
```

en utilisant n'importe quel algorithme d'ordonnement (ligne 2) et se connectent aux agents qui peuvent insérer des tâches provenant des mêmes requêtes (ligne 3). Ensuite, chaque agent enchérit simultanément sur l'ensemble des modes, en fonction des offres reçues précédemment, et envoie les résultats à ses voisins (lignes 5-7). Ensuite, chaque agent résout les conflits qu'il peut avoir avec les offres de ses voisins (lignes 8-10). Lorsqu'il n'y a plus de conflit entre les agents, les allocations pour les tâches externes sont renvoyées au client (ligne 11).

Structures de données. Les principales structures de données utilisées par chaque agent i sont les *offres* (ou *bids*) et les *contributions*. Une *offre* est une structure à 3 dimensions où chaque cellule contient la récompense pour une requête donnée $\hat{\tau}$, un mode donné m et un agent donné j dans \mathcal{N}_i , noté $b_i[\hat{\tau}][m][j]$. La valeur d'une cellule d'offre peut être soit : (i) une valeur positive ou négative finie, lorsque l'agent sait ou évalue combien il gagne ou perd en programmant les tâches du mode dans ses créneaux ; (ii) l'infini négatif, s'il ne peut pas insérer les tâches du mode dans son plan, en raison de la limitation des ressources (plus de place dans ses créneaux) ; (iii) ou vide, s'il n'a pas encore enchéri sur le mode. Les offres sont mises à jour à chaque phase d'enchères, comme expliqué plus loin. Complémentaire à une offre, une *contribution* est l'ensemble des tâches que l'agent j est prêt à programmer pour une requête $\hat{\tau}$ donnée, et un mode m , noté $c_i[\hat{\tau}][m][j]$. A chaque offre correspond une contribution, c'est-à-dire l'ensemble des tâches dont l'insertion aboutit à l'offre proposée. Afin d'éviter les blocages dus à des informations obsolètes, chaque agent i stocke la date du dernier message qu'il a reçu de la part de chaque autre agent j , notée $s_i[j]$. Ces horodateurs seront également échangés afin d'éviter de prendre des décisions basées sur des informations

obsolètes. Enfin, chaque agent i conserve la trace de son *lot* (ou *bundle*), β_i , c-à-d. l'ensemble des modes externes actuellement prévus dans son plan.

Phase d'enchérissement (*bidding*). Contrairement à CBBA et CBGA, où les agents font des offres pour des tâches, l'idée dans MM-CBGA est que les agents font des offres sur les modes pour chaque demande dont ils ont connaissance. L'algorithme 2 présente comment chaque agent met en œuvre cette phase. L'agent tente d'intégrer des modes dans son lot tant que ses créneaux ne sont pas pleins (ligne 4). Pour chaque mode de chaque requête non satisfaite (lignes 6-7), l'agent calcule deux plans -avec et sans le mode-, pour évaluer le coût ou le gain marginal de l'intégration du mode donné dans ses créneaux actuels, en fonction de son lot actuel β_i . Le gain résultant et les tâches respectives constituent l'offre et la contribution de l'agent (lignes 11-12). Les deux plans sont calculés à l'aide d'un algorithme quelconque, noté *solve*, qui peut être MILP ou l'algorithme glouton présentés précédemment. Bien entendu, l'agent n'a pas à recalculer la solution du même sous-problème, en stockant la solution déjà calculée de chaque sous-problème. Le meilleur mode pour chaque demande doit être choisi, mais lorsqu'un agent n'a aucune connaissance des offres des autres agents requises pour un mode, il évaluera une *borne supérieure* pour l'intégration du mode (ligne 13).

$$\begin{aligned}
 \text{UB}(b_i[\hat{\tau}][m]) &\stackrel{\text{def}}{=} \sum_{b_i[\hat{\tau}][m][j] \neq \emptyset} b_i[\hat{\tau}][m][j] \quad (9) \\
 &+ \sum_{\tau \notin \bigcup_j c_i[\hat{\tau}][m][j]} \omega_\tau \quad (10)
 \end{aligned}$$

Cette borne supérieure consiste en l'agrégation des récompenses effectives des offres existantes, comme dans l'équation (9) et des récompenses hypothétiques maximales pour les offres manquantes sur certaines tâches du mode, comme dans l'équation (10). Dès qu'un agent n'est pas d'accord pour choisir le mode (offre $-\infty$), le mode est écarté pour intégration. La raison pour laquelle on utilise des récompenses hypothétiques et pas seulement des récompenses effectives est de permettre aux modes avec une récompense effective négative d'être considérés, s'ils ont le potentiel d'apporter un certain gain grâce à des récompenses non encore découvertes, comme illustré dans l'exemple 1. Dans le cas où cette borne supérieure est trop confiante, elle sera affinée pendant la phase de consensus, en mettant à jour la récompense avec les offres effectives. Dans le cas où plusieurs agents peuvent effectuer la même tâche pour un mode, et qu'il y a donc plus d'offres sur m pour $\hat{\tau}$, seules les meilleures offres, suffisantes pour remplir $\hat{\tau}$, sont utilisées dans la définition de UB (les moins bonnes sont ignorées).

Cette borne supérieure est utilisée pour déterminer

Algorithme 3 : La fonction bid

```

1 Fonction bid( $i$ )
2    $M \leftarrow \emptyset$  // modes déjà analysés
3    $R \leftarrow \emptyset$  // requêtes déjà analysées
4   tant que true faire
5      $\omega^* \leftarrow -\infty, \hat{\tau}^* \leftarrow \emptyset, m^* \leftarrow \emptyset$ 
6     pour chaque  $\hat{\tau} \in \hat{\mathcal{T}}_i^+ \setminus R$  faire // requêtes
7       pour chaque  $m \in M_{\hat{\tau}} \setminus \beta_i$  faire // modes
8          $\pi_i \leftarrow \text{solve}(P_i^- \uplus \beta_i)$ 
9          $\pi_i^m \leftarrow \text{solve}(P_i^- \uplus (\beta_i \cup \{m\}))$ 
10        si  $m \in \{\text{mode}(\tau) \mid (\tau, t) \in \pi_i^m\}$  alors
11          // Tâches
12          de  $m$ 's dans les créneaux de  $i$ 
13           $c_i[\hat{\tau}][m][i] \leftarrow \{\tau \mid (\tau, t) \in \pi_i^m\}$ 
14          // évaluer le gain marginal
15           $b_i[\hat{\tau}][m][i] \leftarrow \omega_{\pi_i^m} - \omega_{\pi_i}$ 
16           $\omega \leftarrow \text{UB}(b_i[\hat{\tau}][m])$ 
17          si  $\omega > \omega^*$  alors // meilleur mode
18             $\omega^* \leftarrow \omega, \hat{\tau}^* \leftarrow \hat{\tau}, m^* \leftarrow m$ 
19          sinon //  $m$  ne peut être planifié
20             $b_i[\hat{\tau}][m][i] \leftarrow -\infty$ 
21             $c_i[\hat{\tau}][m][i] \leftarrow \emptyset$ 
22
23        si  $\omega^* \neq -\infty$  alors // un mode a été trouvé
24        si  $m^* \subseteq \bigcup_j c_i[\hat{\tau}^*][m^*][j]$ 
25        alors // le mode est plein
26          si  $\sum_j b_i[\hat{\tau}^*][m^*][j] > 0$  alors
27             $\beta_i \leftarrow \beta_i \cup \{m^*\}$  // intégrer au lot
28
29           $M \leftarrow M \cup \{m^*\}$ 
30          // ne pas considérer  $m^*$  à nouveau
31           $R \leftarrow \{\hat{\tau} \mid M_{\hat{\tau}} \subseteq M\}$  // ne plus considérer  $\hat{\tau}$ 
32        sinon break
33
34 retourner  $b_i, c_i$ 

```

le prochain meilleur mode candidat à l'intégration dans le lot (lignes 14-15). Une fois que toutes les requêtes et tous les modes restants ont été traités, si un meilleur mode a été trouvé – c-à-d. s'il reste de la place dans les créneaux – et si ce mode est valide, il est inséré dans le lot (lignes 20-22). Un mode est valide lorsque toutes les tâches ont été effectivement choisies (ligne 20), et que sa récompense globale est positive (ligne 21). A la fin de la phase d'enchérissement, chaque agent aura construit son lot, ses offres et ses contributions. Ces deux dernières données sont ensuite envoyées aux agents voisins avec les horodatages, afin d'obtenir un consensus.

Phase de consensus. L'algorithme 4 décrit comment les agents résolvent les conflits pour aligner leurs offres. Lorsque les offres des agents sont en désaccord sur le meilleur mode pour une requête donnée (lignes 4-6), ils résolvent le conflit comme dans CBGA, c-à-d. qu'un agent met à jour ses offres lorsqu'il reçoit des offres de ses voisins avec des informations plus récentes (lignes 8-13). En fonction de ces offres mises à jour et d'un éventuel changement d'avis sur le meilleur mode, un agent peut supprimer des modes de son lot (lignes 14-16). La fonction discard supprime

Algorithme 4 : La fonction consensus

```

1 Fonction consensus( $b_i, c_i, b_j, c_j$ )
2   conflict  $\leftarrow$  false
3   pour chaque  $\hat{\tau} \in \hat{\mathcal{T}}_i^+ \cap \hat{\mathcal{T}}_j^+$  faire // requêtes en commun
4      $m_i \leftarrow \text{argmax}(b_i[\hat{\tau}])$ 
5      $m_j \leftarrow \text{argmax}(b_j[\hat{\tau}])$ 
6     si  $b_i[\hat{\tau}][m_i] \neq b_j[\hat{\tau}][m_j]$  alors // conflit
7       conflict  $\leftarrow$  true
8       pour chaque  $m \in M_{\hat{\tau}}$  faire // mettre à jour
9         pour chaque  $k \in b_i[\hat{\tau}][m][\cdot]$  faire
10          si  $b_j[\hat{\tau}][m][k] \neq \emptyset$  alors
11            si  $k \neq i$  et  $s_j[k] > s_i[k]$  alors
12               $b_i[\hat{\tau}][m][k] \leftarrow$ 
13                 $b_j[\hat{\tau}][m][k]$ 
14               $c_i[\hat{\tau}][m][k] \leftarrow$ 
15                 $c_j[\hat{\tau}][m][k]$ 
16
17        $m_i^* \leftarrow \text{argmax}_{m \in M_{\hat{\tau}}} \{\text{UB}(b_i[\hat{\tau}][m])\}$ 
18       si  $m_i^* \neq m_i$  et  $m_i \in \beta_i$  alors // nouveau meilleur
19         discard( $\beta_i, m_i$ ) // supprimer  $m_i$ 

```

du lot un mode et tous les modes insérés après ce mode, afin de reconsidérer l'allocation des tâches.

Convergence. Afin de converger, les algorithmes basés sur le consensus tels que CBBA et CBGA exigent que la récompense de l'insertion d'une tâche dans un ordonnancement ne puisse pas augmenter si d'autres tâches sont insérées avant elle. Ici, cette condition se traduit par : $\text{solve}(P_i^- \uplus \beta_i) \leq \text{solve}(P_i^- \uplus (\beta_i \cup \{m\}))$. Cette condition, appelée *gain marginal décroissant* [2], est satisfaite dans MM-CBGA puisque nous planifions toujours chaque tâche de chaque mode groupé au moment qui produit l'augmentation maximale du score. Ainsi, si nous essayons d'insérer les mêmes tâches dans un planning plus complet, il y aurait moins de fenêtres de temps disponibles et la tâche devrait être placée soit à un moment moins optimal, soit au même moment que dans le planning original si cette fenêtre de temps ne chevauche toujours pas une autre tâche, comme décrit dans [18].

5 Évaluation expérimentale

Les expérimentations ont été réalisées en Java 11.0.15 et exécutées sur un processeur 20 cœurs Intel(R) Xeon(R) E5-2660 v3 @ 2.60GHz, 62GB RAM, Ubuntu 18.04.5 LTS. Nous évaluons les performances de MM-CBGA en comparaison à deux concurrents de base : un solveur centralisé (centralized, qui est glouton dans notre cas) qui fournit des solutions de bonne qualité (proches de l'optimal) en un temps très limité, et une enchère séquentielle (ssi) qui fournit des solutions de qualité équivalente à CBGA dans des cas mono-mode. La procédure solve utilisée dans MM-CBGA et ssi est également l'algorithme glouton présenté dans la sec-

tion 3. Le temps de calcul rapporté ici est un temps de calcul mono-CPU (pas d'exécution distribuée).

5.1 Scénario d'observation de la Terre

Le scénario que nous utilisons pour évaluer MM-CBGA est fortement inspiré de EOSCSP (Earth Observation Satellite Constellation Scheduling Problems) [19], qui consiste initialement en un MACTA multi-mode non-composite (chaque requête est satisfaite par exactement une tâche parmi plusieurs possibles). Ainsi, nous avons généré des instances MACTA pour planifier des tâches d'observation sur un ensemble de satellites avec plusieurs propriétaires de créneaux orbitaux.

Nous considérons une constellation en orbite terrestre basse (500 km d'altitude). La constellation est composée de 4 plans orbitaux avec une inclinaison de 45 degrés. Chaque plan orbital contient deux satellites en opposition de phase, pour un total de 8 satellites. Ce paramètre de constellation est utilisé pour déterminer les propriétaires des créneaux orbitaux et les fenêtres temporelles des tâches en fonction de certains points d'intérêt (POI) sur la Terre, en utilisant une bibliothèque de mécanique spatiale dédiée. Le scénario que nous simulons s'étend sur 6 heures (21600 secondes). Quatre utilisateurs (nos agents, u_1 à u_4) possèdent jusqu'à 10 créneaux orbitaux chacun, et tenteront de répondre aux demandes émises par un cinquième utilisateur (u_0). La propriété des créneaux orbitaux est une fenêtre de visibilité sur des points d'intérêt choisis au hasard (10 parmi 27 villes européennes). Les créneaux sont attribués en utilisant une procédure *round-robin* : chaque utilisateur choisit la meilleure fenêtre de visibilité restante pour observer l'un de ses POIs, et laisse l'utilisateur suivant choisir son prochain créneau d'orbite, jusqu'à ce qu'il n'y ait plus de fenêtre, ou que la limite de 10 créneaux par utilisateur soit atteinte. En moyenne, sur toutes les instances que nous avons considérées, selon l'angle d'incidence maximum ($\theta_{\max} = \frac{\pi}{6}$) pour effectuer des observations de bonne qualité, la durée des créneaux exclusifs est d'environ 10 minutes (≈ 595 secondes).

Tous les utilisateurs émettent des requêtes. Ce nombre varie au cours des expérimentations pour évaluer son impact sur les performances de l'algorithme. Les propriétaires de créneaux orbitaux émettent de 2 à 20 requêtes dans leurs propres créneaux, tandis que l'utilisateur non propriétaire en émet de 4 à 80 (autant que tous les autres utilisateurs) dans des créneaux exclusifs, afin de stresser le système. Une requête est définie par un POI, et donc toutes les fenêtres de visibilité existantes pour acquérir ce POI sont des fenêtres de temps potentielles pour les tâches d'observation. Nous avons étudié deux configurations. La première ne considère qu'un seul mode par requête,

qui consiste à faire 5 observations distinctes des mêmes POIs, dans les 5 meilleures fenêtres de visibilité, selon l'angle d'observation. Ainsi, nous nous situons dans la configuration mono-mode des tâches composites, où MM-CBGA est équivalent à CBGA. Dans la deuxième configuration, nous considérons 5 modes par requête, ce qui consiste à dégrader le mode précédemment défini (5 observations) en supprimant 1 à 4 fenêtres. On obtient ainsi 5 modes avec respectivement 5, 4, 3, 2 et 1 tâche par mode. Les fenêtres de temps des tâches sont les mêmes que les fenêtres de visibilité : l'agent peut positionner ses observations n'importe où dans les créneaux de l'orbite. La durée des tâches est déterminée aléatoirement entre 20 et 40 secondes. La planification doit également respecter un temps de reconfiguration inter-tâches des satellites de 10 secondes, puisque les satellites ne peuvent effectuer qu'une seule observation à la fois, et doivent pivoter pour pointer vers le prochain POI. L'angle d'incidence (θ), qui a un impact sur la qualité des observations, est utilisé pour déterminer la récompense de chaque tâche τ , plus il est proche de 0° , mieux c'est : $\omega_\tau = \rho(1 - \frac{|\theta|}{\theta_{\max}})$, avec ρ un nombre aléatoire dans $[1;2]$. On obtient ainsi des récompenses moyennes d'environ 1,41 (min $\approx 0,89$, max $\approx 1,99$).

Nous avons exécuté 100 instances de MACTA générées aléatoirement avec une graine dans $[0;99]$ pour chaque paramètre, et nous avons tracé la moyenne, avec un intervalle de confiance de $[0,05;0,95]$. Les valeurs aléatoires sont choisies uniformément dans les intervalles fournis. Cela donne un total de 3000 instances.

5.2 Analyse de performance

Pour évaluer les performances de MM-CBGA, nous analysons les métriques de la figure 3, pour deux configurations : 1 mode par requête (ligne du haut) et 5 modes par requête (ligne du bas). Sur chaque ligne, nous représentons le ratio de qualité par rapport au solveur centralisé, le pourcentage de requêtes satisfaites, le nombre de tâches planifiées, le nombre de messages échangés, la charge de communication (en Mo), et le temps de calcul total (en s), avec un nombre croissant de requêtes.

Requêtes mono-mode. Nous examinons tout d'abord les résultats sur des MACTA avec un seul mode par requête. Dans cette situation, les problèmes à résoudre sont équivalents à ceux résolus avec CBGA. Comme prévu, la qualité des solutions de MM-CBGA est équivalente à celle d'une enchère séquentielle (ssi). Puisqu'il n'y a qu'un seul mode par requête, consistant en cinq tâches d'observation, centralized ne peut pas trouver de solutions satisfaisant toutes les requêtes. De plus, avec de nombreuses requêtes (plus de 160), sur ce système à res-

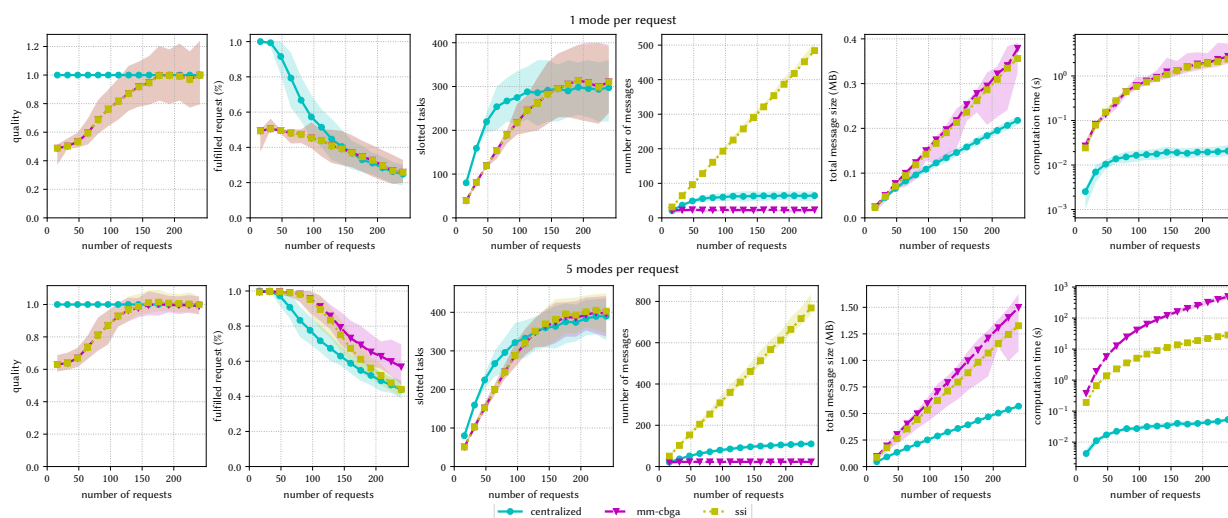


FIGURE 3 – Performances pour les configurations 1 mode par requête (haut) et 5 modes par requête (bas).

sources limitées, centralized est même surpassé par MM-CBGA en termes de requêtes (et donc de tâches) planifiées, car ce dernier permet de planifier des requêtes à récompense non immédiate. Si l'on considère les métriques opérationnelles concernant la communication et le calcul, comme le réseau entre les agents est stable, MM-CBGA ne nécessite qu'un petit nombre constant de messages, alors que ssi nécessite un nombre de messages linéaire par rapport au nombre de modes. Cependant, les structures de données échangées dans MM-CBGA sont quadratiques, ce qui entraîne une charge de communication similaire entre MM-CBGA et ssi. La quantité d'informations est également analysée pour centralized : en effet, les propriétaires de créneaux doivent envoyer toutes leurs données au solveur. En termes de calcul, MM-CBGA, qui nécessite la construction et l'agrégation de données plus importantes (modes fois agents), est plus lent de deux ordres de grandeur que centralized. Ainsi, dans ce cadre mono-mode, MM-CBGA se comporte de manière très similaire à ssi en termes de qualité, de communication et de calcul. En effet, il est équivalent au CBGA. Mais, contrairement à ssi, MM-CBGA est entièrement décentralisé : le WDP est distribué entre tous les agents.

Requêtes multi-mode. Dans cette configuration, chaque requête peut être satisfaite par 5 modes avec une dégradation de 5 tâches à 1 tâche. Ainsi, il y a de la place pour de la relaxation afin de satisfaire plus de requêtes. Ceci est illustré par le pourcentage de requêtes satisfaites, qui ne chute pas aussi rapidement que dans le cadre mono-mode. Il est plus intéressant de noter que MM-CBGA et ssi réalisent plus de requêtes que centralized, pour une qualité moindre sur les petits problèmes (< 150 requêtes). En effet, les deux approches basées sur les enchères remplissent plus de modes avec moins de tâches pour satisfaire plus de requêtes. Sur des instances plus

grandes (> 150 requêtes), les courbes de qualité et de satisfaction de ssi et centralized convergent, pour atteindre des performances équivalentes, tandis que MM-CBGA sert encore plus de demandes pour une récompense globale équivalente, par l'utilisation de UB. Sur le plan de la communication, la taille des messages augmente par rapport au paramètre mono-mode pour MM-CBGA, en raison de la présence de modes multiples dans les offres. Pour la même raison, MM-CBGA a besoin de plus de temps pour calculer les offres et résoudre les conflits, ce qui se traduit par un temps de calcul plus élevé (3 ordres de grandeur de plus que centralized, et 1 ordre de grandeur que ssi). Une fois encore, rappelons que MM-CBGA est entièrement décentralisé et que nous présentons un temps mono-CPU. En résumé, dans un cadre multi-mode, MM-CBGA a tendance à répondre à plus de requêtes avec des modes plus petits que centralized, et fournit même des résultats avec une récompense équivalente à centralized sur des instances plus grandes. Le coût de la décentralisation, de la construction incrémentale des lots et de la confidentialité résulte en une qualité inférieure sur des instances plus faciles/petites, et une charge de communication et de calcul plus élevée, en général.

6 Conclusions

Nous avons abordé le problème de l'allocation de requêtes nécessitant la réalisation de plusieurs tâches atomiques et ayant plusieurs modes à réaliser sur des créneaux privés appartenant à des agents coopératifs. Nous avons modélisé ce problème d'allocation (MACTA) et proposé un nouvel algorithme de consensus (MM-CBGA) pour le résoudre de manière décentralisée, en maximisant la récompense globale et sans dévoiler les plans privés.

Nous avons évalué ses performances sur des problèmes de planification de tâches d'observation de la Terre (EOSCSP). MM-CBGA affiche des performances équivalentes à celles des enchères séquentielles (SSI) sur les paramètres mono- et multi-mode, et atteint la même qualité qu'un solveur glouton centralisé sur des instances plus grandes et plus difficiles. En termes de calcul, MM-CBGA nécessite moins d'étapes pour converger mais plus de temps que SSI sur des instances plus grandes. Ce temps de calcul reste inférieur à 5 minutes, mais il est encore possible de l'améliorer en parallélisant certaines opérations. Pour la communication, MM-CBGA nécessite moins de messages, mais plus d'octets, en raison des informations supplémentaires requises pour résoudre les conflits entre les offres. MM-CBGA s'avère être un ajout intéressant à la famille des algorithmes de consensus, comblant le vide en termes de tâches multi-modes et multi-agents.

Nous identifions des améliorations potentielles, notamment en concevant de meilleures heuristiques et des bornes supérieures plus justes, qui pourraient être conçues pour des types spécifiques de tâches ou de modes (e.g., les modes inclus dans d'autres). De plus, il existe d'autres techniques pour coordonner les actions des agents, par exemple les DCOP, à explorer [6]. Ces algorithmes ont été utilisés avec des performances équivalentes à CBBA sur des tâches atomiques multi-mode [19]. Enfin, nous avons considéré des problèmes statiques, alors que les algorithmes de consensus sont adaptés aux problèmes dynamiques, où les demandes et les agents peuvent apparaître ou disparaître. Nous étudierons les performances de MM-CBGA avec des carnets de commande dynamiques en ligne.

Remerciements. Ces travaux ont été menés grâce au financement du gouvernement français dans le contexte du Programme d'Invertissements d'Avenir, au travers du projet BPI PSPACE "LiChIE" coordonné par Airbus Defence and Space.

Références

- [1] D.-H. Cho, J.-H. Kim, H.-L. Choi et J. Ahn. « Optimization-Based Scheduling Method for Agile Earth-Observing Satellite Constellation ». In : *Journal of Aerospace Information Systems* 15.11 (2018), p. 611-626.
- [2] H. Choi, L. Brunet et J. P. How. « Consensus-Based Decentralized Auctions for Robust Task Allocation ». In : *IEEE Trans. Robotics* 25.4 (2009), p. 912-926.
- [3] J. Coelho et M. Vanhoucke. « Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers ». In : *European Journal of Operational Research* 213.1 (2011), p. 73-82.
- [4] P. Cramton, Y. Shoham et R. Steinberg, éd. *"Combinatorial Auctions"*. MIT Press, 2010.
- [5] M. Dias, R. Zlot, N. Kalra et A. Stentz. « Market-Based Multirobot Coordination: A Survey and Analysis ». In : *Proceedings of the IEEE* 94.7 (2006), p. 1257-1270.
- [6] F. Fioretto, E. Pontelli et W. Yeoh. « Distributed Constraint Optimization Problems and Applications: A Survey ». In : *Journal of Artificial Intelligence Research* 61 (2018), p. 623-698.
- [7] B. P. Gerkey et M. J. Mataric. « A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems ». In : *The International Journal of Robotics Research* 23.9 (2004), p. 939-954.
- [8] Gurobi Optimization. *Gurobi Optimizer*. 2022.
- [9] S. Hunt, Q. Meng, C. J. Hinde et T. Huang. « A Consensus-Based Grouping Algorithm for Multi-agent Cooperative Task Allocation with Complex Requirements ». In : *Cogn. Comput.* 6.3 (2014), p. 338-350.
- [10] IBM Corporation. *IBM ILOG CPLEX Optimization Studio*. 2022.
- [11] A. M. Khamis, A. Hussein et A. M. Elmoogy. « Multi-robot Task Allocation: A Review of the State-of-the-Art ». In : *Cooperative Robots and Sensor Networks 2015*. T. 604. Studies in Computational Intelligence. Springer, 2015, p. 31-51.
- [12] S. Koenig, C. A. Tovey, M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. J. Kleywegt, A. Meyerson et S. Jain. « The Power of Sequential Single-Item Auctions for Agent Coordination ». In : *AAAI 2006*. AAAI Press, 2006, p. 1625-1629.
- [13] M. Lee, S. J. Kim, H.-Y. Kim et H.-L. Choi. « Consensus-based Task Scheduling Algorithm for Agile Earth Observation Satellites with Different Authorities ». In : *ASCEND 2021*. 2021.
- [14] E. Manisterski, E. David, S. Kraus et N. R. Jennings. « Forming Efficient Agent Groups for Completing Complex Tasks ». In : *AAMAS*. 2006, 834-841.
- [15] S. Maqrot, S. Roussel, G. Picard et C. Pralet. « Bundle Allocation with Conflicting Preferences Represented as Weighted Directed Acyclic Graphs – Application to Orbit Slot Ownership ». In : *PAAMS 2022*. T. 13616. LNAI. Springer, 2022, p. 280-293.
- [16] S. Maqrot, S. Roussel, G. Picard et C. Pralet. « Orbit Slot Allocation in Earth Observation Constellations ». In : *PAIS 2022*. T. 351. Frontiers in Artificial Intelligence and Applications. IOS Press, 2022, p. 3-16.
- [17] N. Michael, M. M. Zavlanos, V. Kumar et G. J. Pappas. « Distributed multi-robot task assignment and formation control ». In : *2008 IEEE International Conference on Robotics and Automation*. 2008, p. 128-133.
- [18] S. Phillips et F. Parra. « A Case Study on Auction-Based Task Allocation Algorithms in Multi-Satellite Systems ». In : *AIAA Scitech 2021 Forum*. 2021.
- [19] G. Picard. « Auction-based and Distributed Optimization Approaches for Scheduling Observations in Satellite Constellations with Exclusive Orbit Portions ». In : *AAMAS*. 2022, p. 1056-1064.
- [20] G. Picard. « Multi-Agent Consensus-based Bundle Allocation for Multi-Mode Composite Tasks ». In : *AAMAS*. 2023.
- [21] S. Squillaci, S. Roussel et C. Pralet. « Parallel Scheduling of Complex Requests for a Constellation of Earth Observing Satellites ». In : *PAIS 2022*. T. 351. IOS Press, 2022, p. 100-113.
- [22] X. Wang, G. Wu, L. Xing et W. Pedrycz. « Agile Earth observation satellite scheduling over 20 years: formulations, methods and future directions ». In : *CoRR* abs/2003.06169 (2020).

Consommation adaptative par négociation continue

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

Nous étudions ici le problème de l'allocation de jobs concurrents, composés de tâches situées, sous-jacent au déploiement distribué du patron de conception MapReduce sur une grappe de serveurs. Afin de mettre en œuvre notre stratégie multi-agents qui vise à minimiser le délai moyen de réalisation des jobs (flowtime), nous proposons une architecture composite d'agent qui permet la concurrence des négociations et des consommations. Nos expérimentations montrent que, lorsqu'elle est exécutée en continue lors du processus de consommation, notre stratégie de réallocation : (1) améliore le flowtime; (2) ne pénalise pas la consommation; (3) est robuste aux aléas d'exécution.

Mots-clés : Résolution collective de problèmes, Négociation multi-agents, Architecture d'agent

Abstract

In this paper, we study the problem of allocating concurrent jobs, composed of situated tasks, underlying the distributed deployment of the Map-Reduce design pattern on a cluster. In order to implement our multi-agent strategy that aims at minimizing the mean flowtime of jobs, we propose a composite agent architecture that allows negotiation and consumption concurrency. Our experiments show that, when executed continuously during the consumption process, our reallocation strategy : (1) improves the flowtime; (2) does not penalise the consumption; (3) is robust to execution hazards.

Keywords: Distributed Problem Solving, Agent-based Negotiation, Agent Architecture

1 Introduction

Les sciences des données exploitent de larges volumes de données sur lesquelles des calculs sont effectués en parallèle par différents nœuds. Ces applications mettent à l'épreuve l'informatique distribuée en ce qui concerne l'allocation de tâches et l'équilibrage de charge. C'est le

cas de l'application pratique que nous considérons dans cet article : le modèle de traitement le plus répandu pour traiter des données massives sur une grappe de serveurs, c'est-à-dire le patron de conception MapReduce [10]. Les jobs, qui doivent être exécutés le plus tôt possible, sont composés d'un ensemble de tâches, exécutées par les différents nœuds. L'exécution d'une tâche consiste à traiter des ressources situées sur les nœuds. Comme plusieurs ressources sont requises pour réaliser une tâche sur un nœud, son exécution peut nécessiter 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 parmi des exécutants multiples [2]. L'approche centrée individus permet la distribution d'heuristiques pour des problèmes impraticables à cause de la combinatoire des ordonnancements, autorisant ainsi 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.).

Dans [3], nous avons proposé une stratégie multi-agents de réallocation de tâches pour un ensemble de jobs devant être exécutés le plus tôt possible. Afin de minimiser le délai moyen de réalisation des jobs (*flowtime*), les agents, qui sont coopératifs, négocient pour déterminer les prochaines tâches à déléguer, voire à échanger. Cette stratégie nécessite le déploiement distribué d'agents autonomes qui consomment les tâches et échangent de manière continue certaines d'entre elles pour équilibrer l'allocation courante. Dans cet article, nous formalisons les opérations de consommation de tâches et celles de réallocation et nous proposons une architecture composite d'agent qui permet la concurrence des négociations et des consommations. Selon le principe de la séparation des préoccu-

pations, un premier agent composant est dédié à la consommation (i.e. l'exécution) des tâches, un second aux négociations des réallocations et un troisième à la coordination locale de ces opérations à travers la gestion du lot de tâches. La difficulté réside dans la conception des comportements des agents composants qui ne partagent pas un état global du système (e.g. l'allocation) mais disposent de connaissances locales et partielles. Nos expérimentations montrent que, lorsqu'elle est exécutée en continue lors du processus de consommation, notre stratégie de réallocation ne pénalise pas la consommation et peut améliorer le *flowtime* jusqu'à 37 %, même lorsque les agents ont une connaissance imparfaite de l'environnement d'exécution comme lors d'aléas (i.e. le ralentissement de nœuds).

Après un aperçu des travaux connexes dans la section 2, nous rappelons dans la section 3 la formalisation du problème d'allocation de jobs composés de tâches situées. La section 4 formalise les opérations de consommation/réallocation. Nous décrivons ensuite, dans la section 5, comment le processus de consommation et celui de réallocation sont entrelacés, puis nous détaillons notre architecture d'agents dans la section 6. La section 7 présente nos résultats expérimentaux. La section 8 résume notre contribution et présente nos perspectives.

2 Travaux connexes

De nombreux travaux ont abordé le problème de la réallocation de tâches parmi de multiples exécutants. L'approche centrée individus permet de surmonter les limites des solutions centralisées : l'impossibilité de résoudre les problèmes à grande échelle et la faible réactivité aux changements [2]. Les problèmes d'allocation dynamique des tâches nécessitent notamment de proposer des processus dynamiques qui s'ajustent en permanence aux changements de l'environnement d'exécution ou de performance des exécutants [11]. La plupart de ces travaux s'appuie sur l'algorithme à base de consensus (CBBA – *Consensus Based Bundle Algorithm*) [7] qui est une méthode multi-agents d'assignation en deux phases qui consiste à : (a) sélectionner les tâches à négocier à travers un processus d'enchère ; (b) déterminer les offres qui remportent ces enchères en résolvant les conflits potentiels. En particulier, notre architecture d'agent modulaire s'inspire largement de [1]. Toutefois, nos agents ne visent pas à minimiser le *makespan* (i.e. le temps d'exécution global) mais le *flowtime*. De plus, nous avons préféré ici un pro-

tocole de négociation bilatérale qui permet, en sélectionnant l'interlocuteur, de faire des propositions ciblées et donc de réduire les coûts computationnel et communicationnel liés à la négociation. Finalement, la simulation de l'environnement d'exécution nous permet d'en contrôler les perturbations.

Chen et *al.* envisagent des problèmes d'allocation dynamique de tâches où les tâches sont libérées à des moments incertains [6]. Ils proposent d'ajuster l'allocation des tâches de façon continue en combinant le réordonnement local des agents avec la réallocation des tâches entre agents. De manière similaire, notre stratégie multi-agents s'appuie sur une stratégie de consommation pour définir l'ordonnement local des tâches et sur une stratégie de négociation des tâches à réallouer. Contrairement à [6], nous faisons ici l'hypothèse que l'ensemble des jobs sont initialement connus, mais nos agents sont susceptibles d'avoir une connaissance imparfaite de l'environnement d'exécution.

La plupart des travaux qui considèrent que les perturbations de l'environnement d'exécution font varier le coût des tâches s'appuient sur des techniques de recherche opérationnelle comme l'analyse de sensibilité pour évaluer la robustesse des optima aux perturbations [15], des méthodes incrémentales pour réparer l'allocation optimale initiale lorsque les coûts changent [14] ou l'optimisation combinatoire pour exploiter les mesures de dégradation [13]. De manière similaire, notre stratégie mesure notamment l'écart entre les progrès attendus et ceux observés en vue de modifier l'allocation. Toutefois, notre approche centrée individus permet de résoudre des problèmes à grande échelle.

Creech et *al.* aborde le problème de l'allocation des ressources et de la hiérarchisation des tâches dans les systèmes multi-agents distribués pour des environnements dynamiques [8]. Ils proposent un algorithme d'optimisation de l'allocation des ressources multi-groupes (MG-RAO) qui combine des algorithmes de mise à jour et de priorisation et qui utilise des techniques d'apprentissage par renforcement. À l'inverse des techniques d'apprentissage, notre solution ne nécessite aucun modèle préalable, ni des données, ni de l'environnement, et aucune phase d'exploration car cela ne serait pas pertinent pour l'application pratique qui nous concerne. En effet, le volume de données rend les pré-traitements et l'exploration trop coûteux. De plus, la variabilité des données les rend rapidement obsolète.

Nos précédentes expérimentations ont montré que la durée moyenne de réalisation atteinte par notre stratégie est meilleure que celle obtenue avec les techniques d'optimisation sous contraintes distribuée (DCOP) et reste proche de celle obtenue avec une heuristique classique, avec dans tous les cas un temps de réordonnement significativement réduit [4]. Nous montrons dans cet article comment déployer cette stratégie de manière continue au cours du processus de consommation.

3 Problème

Cette section présente la formalisation, introduite dans [3], du problème d'allocation des jobs concurrents composés de tâches situées.

Un système distribué est composé d'un ensemble de nœuds de calcul capables d'exécuter des tâches. Ces tâches requièrent des ressources, transférables et non consommables, réparties parmi différents nœuds de ressources.

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

- \mathcal{P} est un ensemble de p nœuds de calcul ;
- \mathcal{N}_r est un ensemble de r nœuds de ressource ;
- $\mathcal{E} : \mathcal{P} \times \mathcal{N}_r \rightarrow \{\top, \perp\}$ est une propriété de voisinage qui évalue si un nœud de calcul de l'ensemble \mathcal{P} est local à un nœud de ressource dans \mathcal{N}_r ;
- $\mathcal{R} = \{\rho_1, \dots, \rho_k\}$ est un ensemble de ressources ayant des tailles $|\rho_i|$. La localisation des ressources, qui sont éventuellement répliquées, est déterminée par la fonction :

$$l : \mathcal{R} \rightarrow 2^{\mathcal{N}_r} \quad (1)$$

Une ressource peut être locale ou distante d'un nœud de calcul, selon sa présence ou non sur un nœud de ressource dans le voisinage du nœud de calcul. À partir des fonctions \mathcal{E} et l , nous définissons le prédicat de localité :

$$\begin{aligned} \forall v_c \in \mathcal{P}, \forall \rho \in \mathcal{R}, \text{local}(\rho, v_c) \text{ ssi} \\ \exists v_r \in l(\rho) \text{ t.q. } \mathcal{E}(v_c, v_r) \end{aligned} \quad (2)$$

Les ressources sont accessibles pour tous les nœuds de calcul, même celles sur les nœuds de ressource distants.

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.

L'exécution d'un job (sans date butoir) consiste à exécuter l'ensemble de ses tâches pour produire un résultat.

Définition 2 (Job/Tâche). *Soit \mathcal{D} un système distribué. On considère un ensemble de ℓ jobs $\mathcal{J} = \{J_1, \dots, J_\ell\}$. Chaque job J_i , associé à la date de libération $t_{J_i}^0$, est un ensemble non vide de k_i tâches $J_i = \{\tau_1, \dots, \tau_{k_i}\}$.*

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 $\text{job}(\tau)$ 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 pour un nœud v_i est une estimation *a priori* de son temps d'exécution.

Définition 3 (Coût d'une tâche pour un nœud). *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 :*

$$\begin{aligned} c(\tau, v_j) &= \sum_{\rho_j \in \mathcal{R}_\tau} c(\rho_j, v_j) \\ \text{avec } c(\rho_j, v_i) &= \begin{cases} |\rho_j| & \text{si local}(\rho_j, v_i) \\ \kappa \times |\rho_j| & \text{avec } \kappa > 1 \text{ sinon.} \end{cases} \end{aligned} \quad (3)$$

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

$$\forall T \subseteq \mathcal{T}, c(T, v_i) = \sum_{\tau \in T} c(\tau, v_i) \quad (4)$$

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

Définition 4 (STAP). *Un problème d'allocation de tâches situées est un quadruplet STAP = $\langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ 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 5 (Allocation). Une allocation pour un problème STAP à l'instant t est un vecteur de m lots de tâches ordonnées

$\vec{A}_t = ((B_{1,t}, \prec_1), \dots, (B_{m,t}, \prec_m))$ où chaque lot $(B_{i,t}, \prec_i)$ est l'ensemble des tâches $(B_{i,t} \subseteq \mathcal{T})$ affectées au nœud v_i à l'instant t , associé à un ordre total strict $(\prec_i \subseteq \mathcal{T} \times \mathcal{T})$. $\tau_j \prec_i \tau_k$ signifie que si $\tau_j, \tau_k \in B_{i,t}$ alors τ_j est exécutée avant τ_k par v_i . L'allocation \vec{A}_t vérifie pour l'instant t :

$$\forall \tau \in \mathcal{T}, \exists v_i \in \mathcal{N}, \tau \in B_{i,t} \quad (5)$$

$$\forall v_i \in \mathcal{N}, \forall v_j \in \mathcal{N} \setminus \{v_i\}, B_{i,t} \cap B_{j,t} = \emptyset \quad (6)$$

Toutes les tâches sont allouées (Eq. 5) et chacune n'est allouée qu'à un seul nœud (Eq. 6). Par souci de concision, on note :

- $\vec{B}_{i,t} = (B_{i,t}, \prec_i)$, le lot trié de v_i ;
- $\min_{\prec_i} B_{i,t}$, la prochaine tâche à exécuter par v_i .

Pour évaluer la qualité d'une allocation de tâches, nous considérons le délai moyen de réalisation (*flowtime*), qui mesure le temps moyen écoulé entre la date de libération des jobs et leur date d'achèvement.

Définition 6 (Flowtime). Soient STAP un problème et \vec{A}_t une allocation à l'instant t . On définit :

- le délai d'attente de la tâche τ dans le lot $\vec{B}_{i,t}$,

$$\Delta(\tau, v_i) = \sum_{\tau' \in B_{i,t} | \tau' \prec_i \tau} c(\tau', v_i) \quad (7)$$

- la durée de réalisation de la tâche $\tau \in \mathcal{T}$ pour l'allocation \vec{A}_t ,

$$C_\tau(\vec{A}_t) = \Delta(\tau, v(\tau, \vec{A}_t)) + t - t_{job(\tau)}^0 + c(\tau, v(\tau, \vec{A}_t)) \quad (8)$$

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

$$C_J(\vec{A}_t) = \max_{\tau \in J} \{C_\tau(\vec{A}_t)\} \quad (9)$$

- le délai moyen de réalisation de \mathcal{J} pour \vec{A}_t ,

$$C_{mean}(\vec{A}_t) = \frac{1}{\ell} C(\vec{A}_t) \quad (10)$$

avec $C(\vec{A}_t) = \sum_{J \in \mathcal{J}} C_J(\vec{A}_t)$

Plus particulièrement le délai d'attente (équation 7) correspond au délai d'attente à partir de

l'instant courant t avant que la tâche τ ne soit traitée. Il est à noter que les temps de réalisation, et par conséquent le *flowtime*, dépendent de l'ordre d'exécution des tâches sur chacun des nœuds.

4 Opérations

Nous formalisons ici les opérations de consommation et de réallocation de tâches.

La **consommation** d'une tâche par un nœud consiste pour ce dernier à retirer cette tâche de son lot pour l'exécuter. Cette opération mène à une nouvelle instance de problème où cette tâche a été retirée. En d'autres termes, l'accomplissement d'une tâche est un événement disruptif qui modifie non seulement l'allocation des tâches mais également le problème sous-jacent.

Définition 7 (Consommation de tâche). Soient STAP = $\langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ un problème d'allocation de tâches et \vec{A}_t une allocation. La consommation par un nœud consommateur v_i à un instant t , dont le lot n'est pas vide ($B_{i,t} \neq \emptyset$), aboutit à l'allocation $\vec{A}_t' = \lambda(v_i, \vec{A}_t)$ pour le problème STAP' = $\langle \mathcal{D}, \mathcal{T}', \mathcal{J}', c \rangle$ où :

$$\mathcal{T}' = \mathcal{T} \setminus \{\min_{\prec_i} B_{i,t}\}$$

$$\mathcal{J}' = \begin{cases} \mathcal{J} \setminus \{job(\min_{\prec_i} B_{i,t})\} & \text{si } job(\min_{\prec_i} B_{i,t}) = \\ \{\min_{\prec_i} B_{i,t}\} & \text{sinon} \\ \mathcal{J} & \end{cases}$$

Dans ce dernier cas :

$$\forall J_j \in \mathcal{J} \exists J_j' \in \mathcal{J}' \text{ t.q. } J_j' = \begin{cases} J_j \setminus \{\min_{\prec_i} B_{i,t}\} \\ \text{si } job(\min_{\prec_i} B_{i,t}) = J_j \\ J_j \text{ sinon} \end{cases}$$

et $\vec{A}_t' = (\vec{B}_{1,t}', \dots, \vec{B}_{m,t}') avec$

$$\vec{B}_{j,t}' = \begin{cases} \vec{B}_{i,t} \ominus \min_{\prec_i} B_{i,t} & \text{si } j = i \\ \vec{B}_{j,t} & \text{sinon} \end{cases}$$

Lorsqu'une tâche est consommée, elle est retirée du problème résultant non seulement dans l'ensemble des tâches mais également du job correspondant. Ce dernier peut également être retiré si la tâche était la seule (la dernière) du job. L'allocation résultante est également modifiée. La tâche est retirée du lot où elle se trouvait.

Les tâches sont destinées à être consommées une à une jusqu'à atteindre l'allocation vide.

À l'évidence, la consommation d'une tâche ne peut augmenter le *flowtime* à un instant t . En effet, la consommation d'une tâche fait décroître localement le *flowtime*, à l'instant t ,

$$\sum_{J \in \mathcal{J}} C_J(\lambda(v_i, \vec{B}_{i,t})) < \sum_{J \in \mathcal{J}} C_J(\vec{B}_{i,t}) \quad (11)$$

Cela n'est cependant pas toujours vrai au cours du temps car les coûts effectifs des tâches peuvent être différents des coûts estimés. Si une tâche s'avère plus coûteuse que prévu lors de son exécution, le *flowtime* peut augmenter après sa consommation, comme dans l'exemple 1.

Exemple 1. Soit le problème STAP = $\langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ avec :

- $\mathcal{D} = \langle \mathcal{P}, \mathcal{N}_r, \mathcal{E}, \mathcal{R} \rangle$, un système distribué avec un unique nœud de calcul $\mathcal{P} = \{v_1\}$ associé au seul nœud de ressource $\mathcal{N}_r = \{v_1^r\}$, tel que $\mathcal{E}(v_1, v_1^r) = \top$ et une unique ressource $\mathcal{R} = \{\rho_1\}$ sur le nœud de ressource v_1^r ;
- deux tâches $\mathcal{T} = \{\tau_1, \tau_2\}$;
- un unique job $\mathcal{J} = \{J_1\}$ libéré à $t_{J_1}^0 = 0$ composé des deux tâches $J_1 = \{\tau_1, \tau_2\}$;
- la fonction de coût c telle que $c(\tau_1, v_1) = 2$ et $c(\tau_2, v_1) = 4$.

L'allocation $\vec{A}_0 = (\vec{B}_{1,t})$ avec $\vec{B}_{1,t} = (\tau_1, \tau_2)$. Selon l'équation 10, le *flowtime* est $C_{mean}(\vec{A}_0) = C_{J_1}(\vec{A}_0) = C_{\tau_2}(\vec{A}_0) = \Delta(\tau_2, v_1) + t + t_{J_1}^0 + c(\tau_2, v_1) = c(\tau_1, \vec{A}_t) + 0 + 0 + c(\tau_2, v_1) = 2 + 4 = 6$.

Si la consommation de τ_1 se termine à l'instant $t_1 = 3$, cela signifie que cette tâche s'avère plus coûteuse que prévu lors de son exécution. Par conséquent, le *flowtime* de $\vec{A}_{t_1} = (\vec{B}_{v_1, t_1})$ avec $B_{v_1, t_1} = (\tau_2)$ est $C_{mean}(\vec{A}_{t_1}) = C_{J_1}(\vec{A}_{t_1}) = t_1 + t_{J_1}^0 + c(\tau_2, v_1) = 3 + 0 + 4 = 7 > C_{mean}(\vec{A}_0)$.

Une **réallocation bilatérale** est une opération lors de laquelle une ou plusieurs tâches sont déplacées d'un lot à un autre.

Définition 8 (Réallocation bilatérale). Soit $\vec{A}_t = (\vec{B}_{1,t}, \dots, \vec{B}_{m,t})$ une allocation pour le problème STAP = $\langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ à l'instant t . 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 ($T_1 \subseteq B_{i,t}$ et $T_2 \subseteq B_{j,t}$) aboutit à

l'allocation $\gamma(T_1, T_2, v_i, v_j, \vec{A}_t)$ avec les m lots $\gamma(T_1, T_2, v_i, v_j, \vec{B}_{k,t})$ définis tels que :

$$\gamma(T_1, T_2, v_i, v_j, \vec{B}_{k,t}) = \begin{cases} \overline{B_{i,t} \ominus T_1 \oplus T_2} & \text{si } k = i, \\ \overline{B_{j,t} \ominus T_2 \oplus T_1} & \text{si } k = j, \\ \overline{B_{k,t}} & \text{sinon} \end{cases} \quad (12)$$

Si T_2 est vide, on parle de *délégation*. Sinon, une *réallocation bilatérale* est un échange de tâches.

Nous nous restreignons ici aux échanges mais les réallocations multilatérales mériteraient d'être explorées.

Contrairement à la plupart des autres travaux (e.g. [9]), nos agents ne sont pas individuellement rationnels mais ils ont un but commun qui prime sur leur intérêt individuel : réduire le *flowtime*.

Définition 9 (Réallocation bilatérale socialement rationnelle). Soit \vec{A}_t une allocation à l'instant t pour le problème STAP = $\langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$. La réallocation bilatérale $\gamma(T_1, T_2, v_i, v_j, \vec{A}_t)$ est socialement rationnelle ssi le *flowtime* décroît,

$$C(\gamma(T_1, T_2, v_i, v_j, \vec{A}_t)) < C(\vec{A}_t) \quad (13)$$

Une allocation est dite **stable** s'il n'existe aucune réallocation bilatérale socialement rationnelle. Dans [4], nous avons démontré la terminaison du processus qui itère ce type de réallocations.

5 Processus

Afin d'exécuter de manière concurrente le processus de consommation et celui de réallocation, nous considérons deux types d'agents : (a) les agents de nœud, chacun d'entre eux représentant un nœud de calcul en gérant son lot de tâches (cf. section 6); (b) le superviseur qui synchronise les phases du processus de négociation.

Le processus de consommation se résume à l'exécution concurrente ou séquentielle des différentes tâches par les nœuds de calcul sous la supervision de leur agent. Le processus de réallocation est constitué de multiples réallocations locales qui sont le résultat de négociations bilatérales entre agents de nœud, réalisées de manière séquentielle ou concurrente. Ces processus sont complémentaires. Tandis que les consommations se déroulent en continu, les agents négocient leurs lots de tâches jusqu'à atteindre une allocation stable. La consommation d'une tâche

peut rendre instable une allocation et ainsi déclencher de nouvelles négociations. Le processus de consommation se termine quand toutes les tâches sont exécutées. L'allocation finale, qui est vide, met un terme au processus.

Il est important de noter que ce système multi-agents est intrinsèquement adaptatif. En effet, si le coût d'une tâche s'avère plus élevé que prévu, parce que le temps d'exécution a été sous-estimé ou parce que le nœud l'exécutant est ralenti, alors le processus de réallocation qui se déroule en continu permet de corriger l'allocation en prenant en compte le temps effectivement mesuré après la réalisation des tâches.

La **stratégie de consommation** des agents, détaillée dans [3], spécifie l'ordonnancement des tâches, au sein du lot de tâches, pour leur exécution par le nœud dont ils ont la charge. Afin de réduire la durée de réalisation des jobs, cette stratégie exécute les tâches des jobs les moins coûteux avant celles des jobs les plus coûteux.

La **stratégie de négociation** des agents de nœud, également détaillée dans [3], s'appuie sur un modèle des pairs, notamment une base de croyances, construit à partir des messages échangés et grâce auquel elle détermine si une réallocation est socialement rationnelle, selon les croyances de l'agent. Les agents disposent : (a) d'une stratégie d'offre qui propose des réallocations bilatérales ; (b) d'une règle d'acceptabilité qui évalue si une proposition reçue est ou non socialement rationnelle avant de l'accepter ou de la refuser ; et (c) d'une stratégie de contre-offre qui sélectionne une contre-partie à une délégation afin de proposer un échange de tâches.

Le processus de négociation se décompose en deux phases successives : (1) les agents proposent des délégations qu'ils croient socialement rationnelles et qui sont acceptées ou refusées par leurs pairs ; (2) les agents proposent des délégations qui ne sont pas nécessairement socialement rationnelles mais qui sont susceptibles de déclencher des contre-offres et ainsi des échanges socialement rationnels. Les phases de négociations s'alternent successivement, de façon concurrente à la consommation.

6 Architecture

Pour concevoir un agent de nœud, nous avons adopté une architecture modulaire qui permet la concurrence des négociations et des consommations.

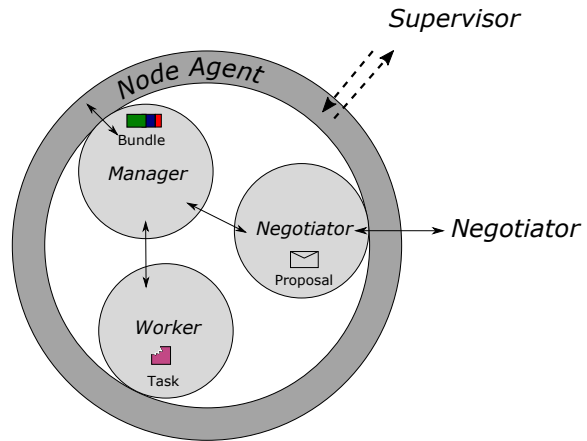


FIGURE 1 – Architecture d'un agent de nœud

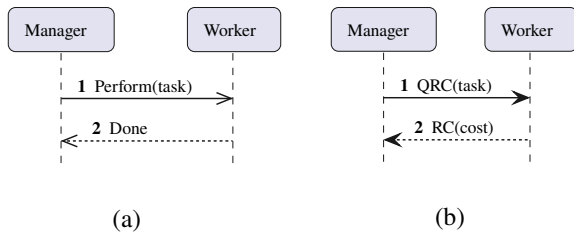
Un agent de nœud est un agent composite constitué de trois agents composants (cf. figure 1), chacun ayant un rôle limité :

- le **worker** exécute (consomme) les tâches ;
- le **negotiator** maintient une base de croyances pour négocier des tâches avec ses pairs ;
- le **manager** gère le lot de tâches du nœud de calcul pour ordonnancer leur exécution par le **worker** en y ajoutant ou supprimant les tâches selon les réallocations bilatérales marchandées par le **negotiator**.

Afin de prioriser la consommation des tâches, le **manager**, dès qu'il est informé que le **worker** est libre, fournit à ce dernier la prochaine tâche à exécuter conformément à la stratégie de consommation, quitte à annuler la réallocation de cette tâche en cours négociation. Cette tâche n'est alors plus éligible pour une potentielle réallocation.

Nous représentons ici les interactions entre les agents composants sous la forme de diagrammes d'interaction où les flèches pleines représentent des appels synchrones (comme dans la figure 2b), les flèches ouvertes des messages asynchrones (comme dans la figure 2a) et les lignes pointillées des messages de réponse.

Après que le **manager** a confié au **worker** une tâche, ce dernier signale au **manager** quand elle est accomplie (cf. figure 2a). Pour prioriser la consommation plutôt que la négociation, la demande de la prochaine tâche à accomplir par le **worker** au **manager** est prioritaire et préempte les interactions de ce dernier avec le **negotiator**. Pour raffiner son estimation du délai d'attente des tâches dans son lot, le **manager** peut demander au **worker** une estimation du temps d'exécution restant pour la tâche en cours (cf. figure 2b).

FIGURE 2 – Interactions *manager/worker*

Lors d'une première phase de négociation, les agents marchandent des délégations (cf. figure 3). Afin de confirmer une telle réallocation bilatérale, le *negotiator* de l'agent proposant demande de manière synchrone au *manager* de mettre à jour le lot de tâches afin que lui-même puisse mettre à jour sa base de croyances avant de s'engager dans de nouvelles négociations. Après cette confirmation, le *negotiator* du répondant en fait de même. Les étiquettes QRC indiquent que le *manager* interagit avec le *worker* selon le protocole de la figure 2b pour prendre en compte le temps d'exécution restant pour la tâche en cours. Lors d'une seconde phase de négociation, les agents marchandent des échanges de tâches et les interactions sont similaires.

Malgré notre architecture modulaire, la principale difficulté subsiste dans la conception des comportements des agents, qui sont spécifiés dans [5] par des automates¹, et dont la complexité est mesurée dans la table 1 en nombre d'états, de transitions et de lignes de code.

Agent	États	Transitions	Lignes
<i>worker</i>	2	7	173
<i>manager</i>	5	23	465
<i>negotiator</i>	9	74	1306
superviseur	9	69	626

TABLE 1 – Complexité des comportements

Le *worker* est : soit disponible; soit en train d'exécuter une tâche et il peut donc estimer le temps d'exécution restant pour la tâche en cours.

Le *manager* gère le lot de tâches et coordonne les opérations de consommation des tâches réalisées par le *worker* avec celles de réallocations marchandées par le *negotiator*. Quand ce dernier lui signale qu'il n'a plus de délégation socialement rationnelle à proposer et qu'il attend les propositions des autres agents de nœud, le *manager* en informe le superviseur. Il continue également de

distribuer les tâches à exécuter au *worker* jusqu'à ce que son lot soit vide. Informé qu'aucun agent de nœud ne détecte d'opportunité de réallocation, le superviseur enclenche le changement de phase de négociation. Enfin, le superviseur clôt le processus quand il apprend par les *managers* que toutes les tâches ont été consommées.

Le *negotiator* répond aux propositions de ses pairs et met à jour sa base de croyances, ce qui lui permet de détecter des opportunités de réallocation. Après avoir proposé une délégation, il attend, avant une date butoir, une acceptation, un refus ou une contre-proposition. Lorsque le *negotiator* a accepté une proposition ou fait une contre-proposition, il attend la confirmation ou l'abandon de son interlocuteur (si la tâche a été consommée depuis). Lorsque l'agent a confirmé son acceptation d'une contre-offre, il attend également la double-confirmation de son homologue. Quand la stratégie d'offre ne suggère aucune délégation, la base de croyances est mise à jour jusqu'à ce qu'une nouvelle opportunité soit trouvée.

7 Expérimentations

Nos expériences visent à valider que, lorsqu'elle est exécutée en continue lors du processus de consommation, la stratégie de réallocation : (1) améliore le *flowtime*; (2) ne pénalise pas la consommation; (3) est robuste aux aléas d'exécution (i.e. le ralentissement de nœuds). Nous présentons ici nos métriques, notre protocole expérimental et nos résultats².

Plutôt que le temps estimé d'exécution des tâches par les nœuds (cf. équation 3), nous considérons le **coût simulé** $c^S(\tau, v)$ comme le coût effectif de la réalisation de la tâche τ par le nœud v :

- avec une connaissance parfaite de l'environnement d'exécution,

$$c^{SE}(\tau, v_i) = c(\tau, v_i) \quad (14)$$

- avec le ralentissement de la moitié des nœuds,

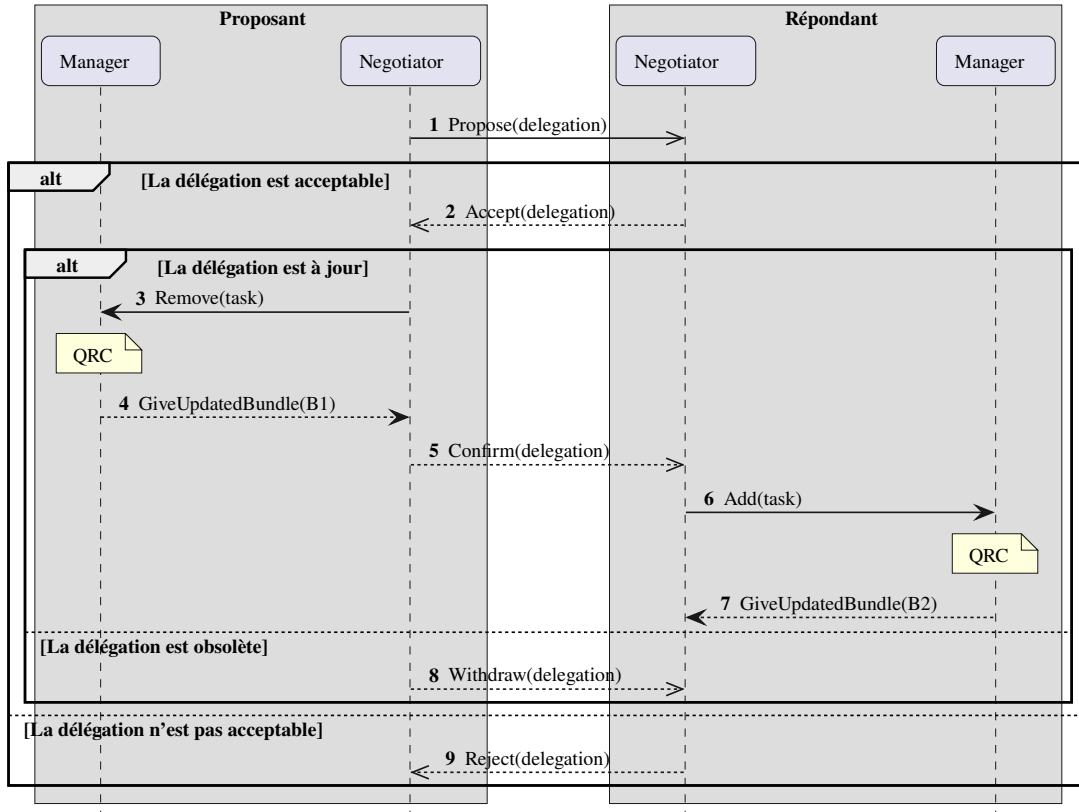
$$c^{SH}(\tau, v_i) = \begin{cases} 2 \times c(\tau, v_i) & \text{si } i \bmod 2 = 1 \\ c(\tau, v_i) & \text{sinon.} \end{cases} \quad (15)$$

C'est pourquoi nous distinguons :

- le **flowtime simulé** $C_{mean}^S(\vec{A}_t)$ calculé à partir d'une allocation \vec{A}_t selon les coûts simulés ;

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

2. Ces expérimentations sont reproductibles à partir des instructions suivantes : <https://gitlab.univ-lille.fr/maxime.morge/smastaplus/-/tree/master/doc/experiments>


 FIGURE 3 – Interactions entre le *manager* et le *negotiator* lors de la première phase de négociation

- le **flowtime réalisé** $C_{mean}^R(\vec{A}_t)$ calculé à partir des temps de réalisation des tâches effectivement mesurés.

On définit le **taux d'amélioration de performance** :

$$\Gamma = \frac{C_{mean}^R(\vec{A}_0) - C_{mean}^R(\vec{A}_e)}{C_{mean}^R(\vec{A}_0)} \quad (16)$$

où \vec{A}_e est l'allocation des tâches au moment de leur exécution et \vec{A}_0 est l'allocation initiale. Il est à noter que si aucune réallocation n'a lieu pendant le processus, $\vec{A}_e = \vec{A}_0$. Le taux d'amélioration est positif si le *flowtime* réalisé de l'allocation atteinte par le processus de réallocation est meilleur (c'est-à-dire plus faible) que celui de l'allocation initiale.

Notre prototype [5] est implémenté avec le langage de programmation Scala et la bibliothèque Akka [12] adaptée aux applications orientées messages, fortement concurrentes, distribuées et robustes. Les expériences ont été réalisées sur une lame munie de 20 CPUs avec 512Go de RAM. Le fait que, dans nos expériences, la différence entre le *flowtime* réalisé et le *flow-*

time simulé de l'allocation initiale ($C_{mean}^R(\vec{A}_0) - C_{mean}^S(\vec{A}_0)$), qui mesure le coût de l'infrastructure, est négligeable, conforte ce choix technologique. Le protocole expérimental consiste, pour les différentes expériences, à générer aléatoirement 25 allocations initiales pour des problèmes STAP distincts. Nous avons fixé empiriquement $\kappa = 2$ comme une valeur réaliste pour capturer le surcoût induit par la récupération des ressources non locales dans un réseau homogène. Nous considérons $m = 8$ nœuds, $l = 4$ jobs, $n \in [40; 320]$ tâches avec 10 ressources par tâche. Chaque ressource p_i est répliquée 3 fois et $|\rho_i| \in [0; 500]$. Afin de ne pas déclencher des négociations inutiles dues à l'asynchronisme des opérations de consommations, nous considérons dans nos expérimentations qu'une réallocation bilatérale est socialement rationnelle si elle fait décroître d'au moins une seconde le *flowtime*.

Hypothèse 1 : la stratégie de réallocation améliore le *flowtime*. Nous considérons ici que les allocations initiales sont aléatoires et que les agents ont une connaissance parfaite de l'environnement d'exécution (c^{SE}). La figure 4 montre les médianes et les écarts types de nos métriques

en fonction du nombre de tâches. Nous observons que le *flowtime* réalisé de la réallocation est meilleur que le *flowtime* réalisé de l'allocation initiale et borné par le *flowtime* simulé de la réallocation (si un oracle calcule la réallocation en temps constant). Notre stratégie améliore le *flowtime* en réallouant en continu lors du processus de consommation les tâches non locales dont la délégation réduit le coût. Le taux d'amélioration de performance (Γ) se situe entre 20 % et 37 %.

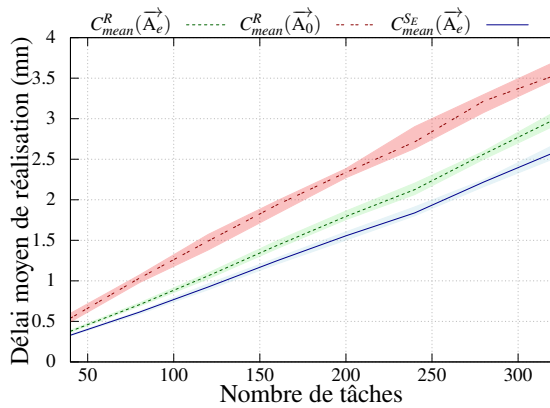


FIGURE 4 – Depuis une allocation aléatoire

Hypothèse 2 : la stratégie de réallocation ne pénalise pas la consommation. Nous considérons ici que les allocations initiales sont stables. Dans la figure 5 le *flowtime* réalisé de la réallocation est similaire au *flowtime* réalisé de l'allocation initiale. Le surcoût de la négociation est négligeable car aucune négociation n'est déclenchée lorsque les agents estiment que l'allocation est stable.

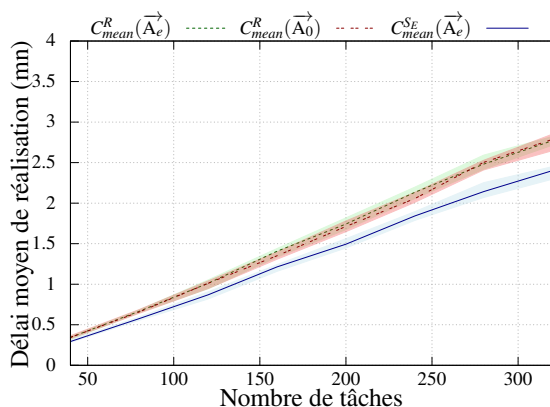


FIGURE 5 – Depuis une allocation stable

Hypothèse 3 : la stratégie de réallocation s'adapte aux aléas d'exécution. Nous considérons ici le coût effectif des tâches qui simule le

ralentissement de la moitié des nœuds, c^{SH} . Nous observons dans la figure 6 que les délais moyens de réalisation ont doublé à cause des aléas d'exécution. De plus, le *flowtime* réalisé de la réallocation reste meilleur que le *flowtime* réalisé de l'allocation initiale malgré une connaissance imparfaite de l'environnement d'exécution des agents. Prendre en compte les temps d'exécution effectifs des tâches déjà réalisées permet au taux d'amélioration de performance (Γ) de se situer entre 30 % et 37 %.

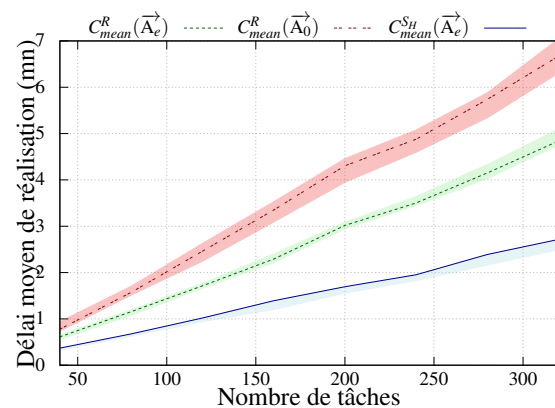


FIGURE 6 – Avec aléas d'exécution

8 Discussion

Afin de concevoir des agents autonomes qui réalisent de manière concurrente des opérations de consommation et de réallocation, nous avons proposé une architecture modulaire d'agent composé de trois agents composants : le *worker* qui exécute les tâches ; le *negotiator* qui mène des réallocations avec ses pairs ; et le *manager* qui coordonne localement ces opérations en gérant le lot de tâches. Sans pour autant connaître l'état global du système, i.e. l'allocation, ces agents disposent de connaissances locales (e.g. la tâche courante, le lot de tâches) et de croyances qui guident leur comportement dans les interactions.

Nos expérimentations montrent que le taux d'amélioration de performance dû à notre stratégie de réallocation, lorsqu'elle est exécutée en continue lors du processus de consommation, peut atteindre 37 %. De plus, le surcoût lié aux négociations est négligeable car, lorsque l'allocation est stable, elles sont suspendues. De plus, même si un ou plusieurs nœuds sont ralentis, notre stratégie de réallocation s'adapte au contexte d'exécution en distribuant plus de tâches aux nœuds qui ne sont pas ralentis, car

elle tient compte du temps d'exécution effectif des tâches déjà réalisées, sans pour autant nécessiter de phase d'apprentissage.

Une analyse de sensibilité pour étudier l'influence du facteur de réplification, du surcoût induit par la récupération des ressources non locales (κ) ou du délai d'annulation de la négociation (*timeout*) va au-delà de la portée de cet article, mais mériterait une étude approfondie. Nous souhaitons également évaluer la réactivité de notre stratégie face à la libération de *jobs* au fil de l'eau.

Plus généralement, nos travaux futurs porteront sur l'intégration de la réallocation des tâches dans un processus d'approvisionnement qui ajoute ou supprime des nœuds de calcul au cours de l'exécution en fonction des besoins des utilisateurs afin de proposer une stratégie multi-agents élastique de passage à l'échelle.

Remerciements. Nous adressons nos remerciements aux relecteurs pour leur travail minutieux et leurs précieux conseils.

Références

- [1] Quentin BAERT, Anne-Cécile CARON, Maxime MORGE, Jean-Christophe ROUTIER et Kostas STATHIS. « Un système multi-agent adaptatif pour la réallocation de tâches au sein d'un job MapReduce ». In : *Revue Ouverte d'Intelligence Artificielle* (2022), p. 557-585.
- [2] Ellie BEAUPREZ, Luc BIGAND, Anne-Cécile CARON, Maxime MORGE et Jean-Christophe ROUTIER. « Réaffectation de tâches de la théorie à la pratique : état de l'art et retour d'expérience ». In : *Actes des JFSMA*. Cépaduès, 2021, p. 51-60.
- [3] Ellie BEAUPREZ, Anne-Cécile CARON, Maxime MORGE et Jean-Christophe ROUTIER. « Échange de tâches pour la réduction de la durée moyenne de réalisation ». In : *Actes des JFSMA*. Cépaduès, 2022, p. 19-28.
- [4] Ellie BEAUPREZ, Anne-Cécile CARON, Maxime MORGE et Jean-Christophe ROUTIER. « Délégation de lots de tâches pour la réduction de la durée moyenne de réalisation ». In : *ROIA* (2023). À paraître, p. 1-29.
- [5] Ellie BEAUPREZ et Maxime MORGE. *Scala implementation of the Extended Multi-agents Situated Task Allocation*. <https://gitlab.univ-lille.fr/maxime.morge/smastaplus>. 2020.
- [6] Yin CHEN, Xinjun MAO, Fu HOU, Qiuzen WANG et Shuo YANG. « Combining re-allocating and re-scheduling for dynamic multi-robot task allocation ». In : *Proc. of SMC*. 2016, p. 395-400.
- [7] Han-Lim CHOI, Luc BRUNET et Jonathan P How. « Consensus-based decentralized auctions for robust task allocation ». In : *IEEE transactions on robotics* 25.4 (2009), p. 912-926.
- [8] Niall CREECH, Natalia Criado PACHECO et Simon MILES. « Resource allocation in dynamic multiagent systems ». In : *CoRR* abs/2102.08317 (2021).
- [9] Anastasia DAMAMME, Aurélie BEYNIER, Yann CHEVALEYRE et Nicolas MAUDET. « The Power of Swap Deals in Distributed Resource Allocation ». In : *Proc. of AAMAS*. 2015, p. 625-633.
- [10] J. DEAN et S. GHAWAT. « MapReduce : Simplified Data Processing on Large Clusters ». In : *Proc. of OSDI*. 2004, p. 137-150.
- [11] Kristina LERMAN, Chris JONES, Aram GALSTYAN et Maja J MATARIĆ. « Analysis of dynamic task allocation in multi-robot systems ». In : *The International Journal of Robotics Research* 25.3 (2006), p. 225-241.
- [12] LIGHTBEND. *Akka is the implementation of the Actor Model on the JVM*. <http://akka.io>. 2020.
- [13] Siddharth MAYYA, Diego S D'ANTONIO, David SALDAÑA et Vijay KUMAR. « Resilient task allocation in heterogeneous multi-robot systems ». In : *IEEE Robotics and Automation Letters* 6.2 (2021), p. 1327-1334.
- [14] G Ayorkor MILLS-TETTEY, Anthony STENTZ et M Bernardine DIAS. « The dynamic hungarian algorithm for the assignment problem with changing costs ». In : *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-27* (2007).
- [15] Changjoo NAM et Dylan A. SHELL. « When to do your own thing : Analysis of cost uncertainties in multi-robot task allocation at run-time ». In : *Proc. of ICRA*. 2015, p. 1249-1254.

Modèles formels multi-agents

Un concept de solutions avec un biais d'exploration pour les jeux de coalitions stochastiques répétés

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

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

Résumé

Classiquement, en formation de coalitions, les agents connaissent à l'avance les utilités déterministes qu'ils vont obtenir des coalitions. Relâcher ces deux hypothèses (déterminisme et connaissance a priori) nous place dans un cadre de jeux de coalitions stochastiques répétés. Les agents doivent décider à chaque pas de temps quelle coalition former sur la base d'informations limitées. Ils obtiennent alors des observations qui permettent de mettre à jour leurs connaissances. Nous proposons un concept de solutions qui intègre explicitement une notion d'exploration pour permettre aux agents de parfois former des coalitions ayant une faible utilité mais qui seraient intéressantes à former pour obtenir plus d'informations. Nous comparons ce concept à une approche gloutonne et mettons en lumière son efficacité en fonction de la structuration des utilités réelles, inconnues des agents.

Mots-clés : Formation de coalitions, Théorie des jeux coopératifs, Décision séquentielle

Abstract

Classically, in coalition formation, agents know in advance the deterministic utilities they will obtain from coalitions. Relaxing these two assumptions (determinism and a priori knowledge) takes place in a framework of repeated stochastic coalition games. The agents must decide at each time step which coalition to form on the basis of limited information. They then obtain observations that allow them to update their knowledge. We propose a solution concept that explicitly integrates a notion of exploration bias to allow agents to sometimes form coalitions that have a low utility but that would be interesting to form to obtain more information. We compare this concept to a greedy approach and highlight its effectiveness with respect to the structure of the real utilities, unknown to the agents.

Keywords: Coalition Formation, Cooperative Game Theory, Sequential Decision

1 Introduction

Dans un système multi-agents (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. Cependant, la majorité des travaux sur la formation de coalitions font deux hypothèses fortes. La première est que les agents ont une connaissance parfaite a priori de l'intérêt à former une coalition, c'est-à-dire du gain obtenu en la formant. La seconde hypothèse est que ce gain est déterministe. Ces deux hypothèses ne semblent pas adaptées à des problèmes réels où le gain exact obtenu par une coalition n'est connu qu'a posteriori. De plus, si cette même coalition se reforme par la suite, ce gain n'a pas de raison d'être strictement le même, dû à des facteurs internes ou des externalités.

À titre d'exemple, considérons le cadre de l'*Industrie 4.0* où des robots de différentes natures doivent se coordonner. Les problématiques induites par ce problème sont nombreuses, comme celle des déplacements, de l'interaction planifiée et, ce qui nous intéresse ici, la constitution d'équipe en environnement incertain. En effet, certaines tâches ne pouvant être réalisées seul, les agents de cette industrie peuvent être amenés à former des groupes de manière répétée. Cependant, des facteurs internes et externes peuvent influencer sur la qualité des résultats produits par ces groupes. Par exemple, un facteur interne peut être les compétences individuelles des agents dont les effets peuvent être stochastiques, couplées à leur capacité à mieux interagir avec certains agents plutôt qu'avec d'autres. Un facteur externe peut être un effet environnemental indépendant des agents, comme l'arrivée d'une sinistre dans les locaux de l'un d'entre eux. Il est donc approprié de relâcher ces deux hypothèses de déterminisme et connaissance a priori.

Cependant, cela soulève de nouvelles questions. Si les agents n'ont plus de connaissance a priori sur les coalitions, comment peuvent-ils en obtenir ? Si le gain produit par les coalitions est stochastique, comment peuvent-ils l'estimer ? Une façon de modéliser de tels jeux de coalitions aux hypothèses relâchées est d'utiliser des jeux répétés, qui permettent d'observer le résultat d'un même jeu séquentiellement. Ainsi, les agents peuvent observer l'état du jeu à différents moments et sont capables d'en extraire de l'information. Néanmoins, l'objectif principal de la formation de coalitions est de partitionner les agents en coalitions. Dans ce nouveau contexte où les utilités des coalitions sont stochastiques et inconnues des agents, comment décider des coalitions à former ? Ce choix peut être décomposé en plusieurs questions. Comment les agents peuvent-ils favoriser la formation d'une coalition plutôt qu'une autre en fonction de ce qu'ils savent d'elles ? À quel moment considèrent-ils qu'ils en savent suffisamment sur une coalition pour en évaluer correctement son utilité ? Comment les agents peuvent-ils décider collectivement des interactions qui pourraient être acceptées par tous ?

Cet article se place dans la continuité de nos travaux précédents publiés aux JFSMA, à savoir un état de l'art sur la formation de coalitions [13] et un protocole distribué de formation de coalitions classique (i.e. déterministe et non répété) [14]. Nous proposons alors dans cet article un nouveau concept de solutions pour la formation de coalitions stochastique répétée, fondé sur un principe exploration-exploitation, principe bien connu de l'apprentissage par renforcement. Pour cela, nous redéfinissons un concept de solutions existant, en y intégrant une notion d'intérêt à l'exploration afin de permettre aux agents de former des coalitions stables tout en obtenant plus d'information via un compromis entre l'exploitation d'un gain plutôt connu et l'exploration de coalitions dont le gain est inconnu ou très incertain. Nous montrons que notre concept de solutions est très efficace sur des fonctions caractéristiques non structurées, et est meilleur qu'une stratégie ϵ -glouton hormis dans le cas d'une fonction caractéristique très structurée.

Dans la suite, nous présentons en section 2 les jeux de coalitions et les travaux relatifs à la formation de coalitions stochastique répétée. Nous décrivons en section 3 un nouveau concept de solutions intégrant explicitement une notion d'exploration, ainsi qu'un exemple de définition de cette dernière. Enfin, la section 4 est consacrée aux résultats expérimentaux.

2 État de l'art

Dans cette section, nous présentons succinctement les fondements de la formation de coalitions classique ainsi que des approches stochastiques et répétées de ce problème. Pour une vue d'ensemble plus complète, nous renvoyons le lecteur vers l'article de synthèse [13].

2.1 Formation de coalitions classique

Quand les agents coopèrent, ils forment une *coalition*. Celle-ci produit une certaine *utilité*.

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 partition des agents en coalitions est appelée une *structure de coalitions* et une *solution* à un jeu de coalitions est définie comme suit.

Definition 2 (Solution) *Une solution à un jeu de coalitions \mathcal{G} est un tuple $S_{\mathcal{G}} = \langle \mathcal{CS}, \vec{x} \rangle$ où \mathcal{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, une solution doit être *acceptable* par tous les agents, c'est-à-dire qu'ils ne doivent pas vouloir former ou rejoindre une autre coalition (où ils gagneraient plus par exemple). Une telle solution est dite *stable*. Différents critères de stabilité existent et l'ensemble des solutions respectant un de ces critères est appelé un *concept de solutions*.

Intéressons-nous au concept du *cœur* et sa généralisation, l' *ϵ -cœur* [18, 20]. Le cœur est l'ensemble des solutions $\langle \mathcal{CS}, \vec{x} \rangle$ 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*.

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

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

Cette variante autorise à réduire le gain des agents d' ϵ afin de rendre la solution stable. Ici, ϵ est la pire perte de gain parmi tous les agents qui assure la stabilité. Dans la suite, nous considérons uniquement l' ϵ -cœur, car toute solution à un jeu appartient à un ϵ -cœur pour un ϵ , et plus précisément, nous nous intéressons à l' ϵ -cœur ayant le plus petit ϵ pour lequel une solution existe, appelé le *dernier cœur* [9].

2.2 Fonctions caractéristiques stochastiques

Comme mentionné précédemment, le fait que l'utilité de la coalition soit déterministe n'est pas toujours réaliste. Dans la littérature, certains travaux proposent de relâcher cette contrainte et d'intégrer la stochasticité dans les fonctions caractéristiques [11, 8, 15]. Cependant, la nature de l'incertitude dans ces modèles diffère.

Par exemple, Jeong et Shoham ont proposé une distribution de probabilité sur des mondes représentant des jeux de coalitions, chacun d'entre eux ayant une fonction caractéristique déterministe [15]. Les vecteurs de gains sont alors des distributions de l'utilité des coalitions dans chaque monde possible, et les agents expriment des préférences sur ces distributions.

Chalkiadakis et Boutilier ont considéré une fonction caractéristique déterministe modélisée dans un environnement stochastique avec des processus de décision markoviens partiellement observables [8]. Les agents ont des croyances sur les capacités des autres et la même structure de coalitions peut conduire à différents états du monde. Ici, les auteurs considèrent un concept de stabilité appelé le noyau bayésien [7, 10].

Charnes et Granot considèrent simplement que la valeur d'une coalition est une variable aléatoire : [12]. La fonction caractéristique est alors réécrite comme $v : 2^N \rightarrow \mathcal{X}_{2^N}$. Ainsi, lorsqu'une coalition est formée, l'utilité produite est déterminée par la variable aléatoire, qui suit une loi normale. Dans ce modèle, ils calculent leurs vecteurs de gains en associant à chaque agent d'une coalition une part égale de l'espérance de la variable aléatoire associée à la coalition.

Dans cet article, l'objectif est que notre modèle puisse représenter des situations de stochasticité hétérogène. Par exemple, où l'incertitude pourrait provenir de la fiabilité des agents qui la composent pour une coalition, et être due à l'environnement pour une autre. Les deux premiers modèles présentés ci-dessus ne correspondent pas à notre exigence, puisque la stochasticité provient

d'éléments particuliers spécifiques. Le modèle de Jeong et Shoham nécessite la définition de mondes dans lesquels les fonctions caractéristiques sont déterministes, et la nature de la stochasticité ne peut provenir que de l'incertitude sur le monde réel. Si cela nous permet de modéliser des fonctions caractéristiques stochastiques, cela ne correspond pas à une nature hétérogène de l'incertitude. En ce qui concerne le modèle de Chalkiadakis et Boutilier, il repose sur l'incertitude de l'environnement et les croyances sur les compétences des agents. Ce modèle est plus permissif que le précédent, mais pas encore assez général de notre point de vue car les compétences des agents sont toujours déterministes en elles-mêmes. C'est pourquoi nous nous positionnons dans la continuité des travaux de Charnes et Granot, ce qui nous permet de modéliser l'hétérogénéité de la stochasticité par l'utilisation d'une variable aléatoire.

2.3 Jeux de coalitions répétés

Si nous relâchons l'hypothèse de connaissance parfaite de la fonction caractéristique, qu'elle soit stochastique ou non, les agents ne savent rien a priori des utilités produites par les coalitions. Il devient alors intéressant d'utiliser un jeu répété [4]. En effet, dans un contexte de formation de coalitions, répéter le jeu permet d'observer les utilités des coalitions formées, d'estimer la fonction caractéristique et de trouver une solution optimale stable dans le temps.

Par exemple, Blankenburg *et al.*, Louati *et al.* et Bettinelli *et al.* ont proposé des modèles de jeux de coalitions répétés ainsi que des protocoles fondés sur des notions de confiance [6, 17] ou de proximité sociale [5] pour déterminer avec quels agents former des coalitions. Ces protocoles s'appuient sur des étapes similaires : (1) communications à propos des croyances sur les compétences, la similarité ou la confiance des agents ; (2) formation (parfois dynamique) des coalitions ; (3) paiement des agents si nécessaire avant l'achèvement de la tâche pour encourager le bon comportement ; (4) exécution et évaluation des tâches ou des services. De cette façon, les agents apprennent un degré de fiabilité ou de pertinence envers les autres en observant les utilités des coalitions auxquelles ils participent. Toutefois, il s'agit d'approches *gloutonnes* qui ne cherchent pas à former des coalitions pour obtenir plus d'information.

Nous pouvons également mentionner à nouveau Chalkiadakis et Boutilier [8] dont le modèle est

basé sur un processus d'apprentissage par renforcement bayésien, qui peut être vu comme un jeu répété. À chaque pas de temps, les agents doivent choisir de former une coalition. Une fois les coalitions formées, ils prennent une action qui provoque une transition stochastique à un état du monde. Les agents observent ce qu'il s'est passé et mettent à jour leurs croyances. Cela leur permet d'apprendre les compétences des autres agents mais aussi le modèle de transition stochastique entre les états.

Konishi et Debraj [16] ont proposé un processus répété de formation de coalitions dans le but d'étudier les équilibres dans ces processus. À chaque pas de temps, les coalitions doivent choisir une action commune à réaliser. Elles peuvent ne rien faire, ou se déplacer dans un état autorisé (qui correspond à la description d'une structure de coalitions, y compris le vecteur de gains). Les états autorisés d'une coalition, appelés *mouvements coalitionnels*, sont limités à des mouvements Pareto-efficaces afin de permettre la convergence.

Les deux premiers modèles sont intéressants mais sont principalement axés sur l'apprentissage des compétences ou des fiabilités individuelles influant sur une fonction caractéristique déterministe. Le troisième modèle est également intéressant mais la contrainte de mouvements Pareto-efficaces autorise les agents à faire des mouvements irrationnels. Par conséquent, nous allons plutôt nous éloigner de ces modèles pour apprendre une fonction caractéristique stochastique telle que définie par Charnes et Granot. Dans les sections suivantes, nous présentons notre modèle ainsi qu'un concept de solutions intégrant une notion d'intérêt à l'exploration, puis un protocole expérimental pour le mettre en pratique, et enfin quelques résultats.

3 Un ϵ -cœur avec exploration

Afin de répondre à notre problématique, nous définissons en premier lieu un modèle de jeux de coalitions stochastiques répétés. Nous présentons un concept de solutions intégrant un équilibre exploration-exploitation, principe souvent utilisé dans les bandits manchots. En effet, dans les deux cadres, les objets de la décision (respectivement les coalitions¹ et les bras des bandits) génèrent un gain (ou une utilité) lorsque choisis,

1. Nous pourrions envisager de considérer un bras non pas par coalition mais par structure de coalitions. Toutefois, cela conduirait à devoir envisager un nombre de bras très supérieur aux $2^{|N|}$ coalitions.

ce gain dépendant d'une distribution de probabilité inconnue des agents. Une différence néanmoins est que dans les bandits manchots, un seul bras est tiré à chaque pas de temps, tandis que dans la formation de coalitions, cela concerne (potentiellement) plusieurs coalitions, et toutes ne peuvent pas être formées en même temps. Cet équilibre est défini de manière générique afin de prendre en compte différentes caractérisations d'intérêt, par exemple un biais d'exploration qui décrit un intérêt à l'information, ou bien un biais de réputation, qui décrit un intérêt lié à la fiabilité des agents.

3.1 Jeux de coalitions stochastiques répétés

Le modèle de jeux de coalitions stochastiques répétés que nous définissons est inspiré des travaux de Charnes et Granot.

Definition 4 (RSCG) Soit $\mathcal{G} = \langle N, \mathbb{T}, v, \hat{v}, i \rangle$ un jeu de coalitions stochastique répété où :

- $N = \{a_1 \dots a_n\}$ est un ensemble d'agents,
- $\mathbb{T} \subset \mathbb{N}^+$ est un ensemble de pas de temps distincts,
- $v : 2^N \rightarrow \mathcal{X}^{2^N}$ est une fonction caractéristique qui à chaque coalition associe une variable aléatoire. Pour une coalition $C \subseteq 2^N$ donnée, nous notons $v(C) = \mathcal{X}^C$. Cette fonction caractéristique est inconnue des agents.
- $\hat{v} : 2^N \times \mathbb{T} \rightarrow \hat{\mathcal{X}}^{2^N}$ est une fonction caractéristique qui associe à chaque coalition au pas de temps $t \in \mathbb{T}$ une estimation de l'utilité.
- $i : 2^N \times \mathbb{T} \rightarrow \mathbb{R}$ une fonction d'intérêt qui à chaque coalition associe un intérêt quantitatif à un pas de temps donné. Nous notons $i(C, t)$ l'intérêt de la coalition C à un pas de temps t .

À chaque pas de temps, les agents de N doivent décider d'une solution au jeu, malgré le fait qu'ils ne connaissent pas a priori la fonction caractéristique v . Une solution est, comme dans un contexte déterministe, un tuple constitué d'une structure de coalitions et d'un vecteur de gains. Cependant, le gain des agents est un gain estimé *ex ante* basé sur ce qu'ils savent de v .

Definition 5 (Solution à un RSCG) Une solution S^t à un pas de temps $t \in \mathbb{T}$ à un jeu \mathcal{G} est un tuple $S^t = \langle \mathcal{CS}^t, \bar{x}^t \rangle$ tel que :

- \mathcal{CS}^t est une structure de coalitions de N ,
- $\bar{x}^t = \{x_1^t, \dots, x_n^t\}$ est un vecteur de gains tel que $x_i^t \geq 0$ est le gain de l'agent a_i calculé selon l'utilité estimée de la coalition à laquelle il appartient dans la structure \mathcal{CS}^t .

Nous proposons d'adapter le concept de solutions de l' ϵ -cœur en considérant que la valeur d'une coalition, c'est-à-dire son intérêt à être formée à un pas de temps donné, dépend de deux éléments : une estimation de son utilité dont les gains des agents sont directement dérivés, et d'un intérêt que les agents ont de la former afin d'obtenir plus d'informations sur son utilité réelle.

Comme mentionné ci-dessus, le gain des agents pour une solution donnée est une estimation. Une fois la solution trouvée et les coalitions formées, les utilités réelles produites par les coalitions sont le résultat de processus stochastiques paramétrés par la fonction caractéristique. Nous supposons que ces utilités sont observées par tous les agents. Notons X_t^C l'observation de l'utilité produite par la coalition C au pas de temps t .

Definition 6 (Observations) Soit \mathcal{O}_t un ensemble d'observations au pas de temps t correspondant à l'ensemble des coalitions formées à chaque pas de temps avant t et leurs utilités réelles produites :

$$\mathcal{O}_t = \{(C, t', X_{t'}^C) : C \subseteq 2^N, t' \in \mathbb{T}, t' < t\}$$

Par la suite, notons $\mathcal{O}_t(C)$ l'ensemble des observations au pas de temps t associé à la coalition $C \subseteq 2^N$. Cet ensemble d'observations permet de mettre à jour les connaissances des agents sur la fonction caractéristique.

Dans la suite, nous faisons l'hypothèse que les agents estiment l'utilité des coalitions comme des lois normales. Ainsi, pour une coalition donnée $C \subseteq 2^N$, $\hat{v}(C, t)$ est caractérisé par la espérance et la variance d'une loi normale sur l'ensemble des observations.

Definition 7 (Estimation de l'utilité) Au pas de temps $t \in \mathbb{T}$ et pour la coalition $C \subseteq 2^N$, la valeur estimée de C , $\hat{v}(C, t)$, est donnée par $\hat{\mu}(C, t)$ son espérance et $\hat{\sigma}^2(C, t)$ sa variance, calculées à partir des observations $\mathcal{O}_t(C)$.

L'incertitude portant sur l'utilité produite par les coalitions une fois formées, une solution doit tenir compte de cette incertitude pour être stable.

3.2 Intérêt des coalitions

La nature exacte de l'intérêt que les agents ont pour une coalition peut dépendre du problème. Toutefois, cet intérêt a pour objectif de rendre

possible l'exploration d'autres solutions potentiellement intéressantes pour les agents mais qui pourraient être jugées non stables au sens d'un concept de solutions classique.

Remarquons que dans la formation de coalitions, nous devons comparer des structures de coalitions, ce qui implique donc de comparer des coalitions différentes. Par exemple dans le cœur, vérifier la stabilité d'une solution consiste à comparer l'utilité d'une coalition à la somme des gains individuels des agents de cette même coalition où qu'ils soient dans la solution. Nous devons considérer une forme d'intérêt nous permettant de telles comparaisons, c'est-à-dire calculer à partir de l'intérêt d'une coalition l'intérêt individuel des agents qui la composent.

Definition 8 (Intérêt individuel) L'intérêt individuel $i_j(C_j, t)$ d'un agent $a_j \in C_j$ à un pas de temps t est :

$$i_j(C_j, t) = \frac{i(C_j, t)}{|C_j|}$$

Cette répartition égalitaire est une des nombreuses façons de distribuer l'intérêt et, contrairement à la valeur de Shapley, représente le fait que chaque agent d'une coalition possède le même intérêt pour cette coalition, indépendamment des autres coalitions auxquelles ils peuvent appartenir. De plus, plus une coalition contient d'agents, plus leur intérêt individuel sera faible. Cette répartition va donc avantager les coalitions de faible cardinalité, car plusieurs observations distinctes peuvent rapporter plus d'information qu'une seule. Cet intérêt individuel permet de définir l'intérêt d'une coalition au regard d'une structure de coalitions donnée, indépendamment du fait que les agents de cette coalition soient ensemble ou non dans la structure.

Definition 9 (Intérêt collectif) L'intérêt collectif $i^{CS}(C, t)$ des agents d'une coalition C au regard d'une structure de coalitions CS à un pas de temps t est :

$$i^{CS}(C, t) = \sum_{a_j \in C} i_j(C_j^{CS}, t)$$

où C_j^{CS} est la coalition de a_j dans CS et C n'appartient pas nécessairement à CS .

3.3 λ -cœur

Afin d'intégrer cet intérêt des coalitions au concept de solutions, nous devons l'agréger à

l'utilité. Pour rester générique dans un premier temps, nous considérons de manière abstraite un opérateur d'agrégation noté \oplus . Selon la nature exacte de l'intérêt, cet opérateur peut prendre différentes formes, par exemple une *addition*, une *multiplication* ou encore un *maximum*.

Les différents éléments décrivant l'intérêt des agents étant définis, nous pouvons désormais construire notre concept de solutions, le λ -cœur, fondé sur un principe exploration-exploitation. Pour cela, nous adaptons le concept de l' ϵ -cœur en intégrant l'opérateur de composition ainsi que la fonction d'intérêt. Nous ajoutons donc d'un côté de l'inéquation, l'intérêt d'une coalition à l'espérance d'utilité de la coalition, et de l'autre côté l'intérêt collectif à la somme des gains des agents au regard de la solution considérée.

Definition 10 Une solution $\langle \mathcal{CS}^t, \vec{x}^t \rangle$ appartient au λ -cœur si et seulement si $\forall C \subseteq N$:

$$x^t(C) \oplus i^{\mathcal{CS}^t}(C, t) \geq \hat{\mu}(C, t) \oplus i(C, t) - \lambda$$

avec $x^t(C) = \sum_{a_i \in C} x_i^t$

De manière similaire à l' ϵ -cœur, le dernier cœur pour ce concept du λ -cœur est défini comme étant celui ayant le plus petit λ pour lequel une solution existe. Nous pouvons désormais proposer un exemple d'instanciation de ce concept de solutions en définissant l'intérêt comme un biais d'exploration, et l'opérateur d'agrégation comme étant une *addition*.

3.4 Exemple d'intérêt : biais d'exploration

Une notion d'intérêt pertinente est celle de l'exploration, que nous retrouvons dans le problème des bandits manchots. Pour ce problème, de nombreuses stratégies ont été proposées, et notamment les stratégies fondées sur une *borne supérieure de confiance* appelée *UCB* (pour *Upper Confidence Bound*) [1]. Parmi les stratégies fondées sur ce principe, il existe *UCB-V*, qui a été proposée pour le problème des bandits manchots par Audibert *et al.* [2]. Celle-ci décrit un biais d'exploration prenant en compte la variance des distributions de probabilité sous-jacentes des bras du bandit manchot a été prouvée plus efficace que la stratégie *UCB-I* [3]. Nous adaptons donc *UCB-V* pour l'appliquer aux jeux de coalitions stochastiques répétés.

Definition 11 (Biais d'exploration UCB-V)

Le biais d'exploration *UCB-V* pour une coa-

lition C à un pas de temps t est défini comme suit :

$$i(C, t) = \sqrt{\frac{2\hat{\sigma}^2(C, t)\eta}{|O_t(C)| + 1}} + c \frac{3b\eta}{|O_t(C)| + 1}$$

avec $\eta = \zeta \cdot \log(|O_t| + 1)$

Certaines constantes doivent être définies. La constante b définit la borne supérieure des gains du problème, cela est donc dépendant de ce dernier. Cependant, nous pouvons faire l'hypothèse que les utilités sont normalisées sur l'intervalle $[0, 1]$ comme dans les bandits manchots, et ainsi définir $b = 1$. Les constantes ζ et c sont des paramètres de contrôle de l'exploration (en particulier ζ). Nous reprenons ici les valeurs de l'article originel, dans lequel Audibert *et al.* montrent l'efficacité de ces constantes lorsqu'elles sont donc définies comme $\zeta = 1, 2$ et $c = 1$.

4 Expérimentations

Nous évaluons de manière empirique les performances de notre concept de solutions.

4.1 Protocole expérimental

Dans un premier temps, nous construisons 200 couples de jeux différents avec des fonctions caractéristiques uniques, pour 6 agents. Chaque couple de jeux est construit avec deux structures de fonctions caractéristiques différentes. La première fonction caractéristique est tirée selon le modèle *NDCS* (*Normally Distributed Coalition Structures*) [19]. Ce modèle permet de construire des fonctions caractéristiques structurées, mais sans contraindre fortement le modèle comme avec des structures monotones ou superadditives [9]. Ainsi, l'espérance d'utilité μ_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 $[0, 1]$. La deuxième fonction caractéristique n'est pas structurée, car tirée aléatoirement de façon uniforme pour chaque coalition. Ainsi, l'espérance d'utilité μ_C de chaque coalition $C \subseteq N$ est tirée selon une loi uniforme $\mathcal{U}(0, 1)$. Dans les deux modèles de structuration, les variances σ_C^2 de chaque coalition C sont tirées selon la loi uniforme $\mathcal{U}(0, \frac{\mu_C}{2})$. Chaque fonction caractéristique est ensuite normalisée sur l'intervalle $[0, 1]$.

Dans un second temps, afin de créer une série de jeux des plus aux moins structurés, pour

chaque couple de jeux, nous créons des jeux intermédiaires à l'aide d'une transformation linéaire en appliquant un facteur de transformation $w \in [0, 1]$. Ainsi, un facteur de transformation de 0 correspond au jeu structuré NDCS, tandis que le facteur 1 correspond au jeu structuré aléatoirement. Un jeu est créé par pas de 0.05 pour w entre les deux jeux du couple, ce qui correspond donc à 19 jeux intermédiaires supplémentaires. Notre concept de solutions est donc évalué sur 4200 jeux et sur 100 pas de temps chacun.

Exemple 1 Soient C et C' deux coalitions, et v_1 et v_2 deux fonctions caractéristiques respectivement structurées aléatoirement et NDCS :

$$v_1 = \{C = \mathcal{N}(0.6, 0.2), C' = \mathcal{N}(0.1, 0.4)\}$$

$$v_2 = \{C = \mathcal{N}(0.2, 0.4), C' = \mathcal{N}(0.5, 0.1)\}$$

Pour un facteur de transformation de 0.4, les utilités de C et C' sont telles que :

$$v_{(1,2)}^{0.4} = \{C = \mathcal{N}(0.36, 0.32),$$

$$C' = \mathcal{N}(0.34, 0.22)\}$$

Pour un facteur de transformation de 1, la fonction caractéristique résultante est v_1 :

$$v_{(1,2)}^1 = \{C = \mathcal{N}(0.6, 0.2),$$

$$C' = \mathcal{N}(0.1, 0.4)\}$$

Enfin, pour un facteur de transformation de 0, la fonction caractéristique résultante est donc v_2 :

$$v_{(1,2)}^0 = \{C = \mathcal{N}(0.2, 0.4),$$

$$C' = \mathcal{N}(0.5, 0.1)\}$$

Ces jeux sont également joués avec la stratégie ϵ -glouton, qui est une stratégie de référence dans le domaine des bandits manchots [21]. Elle décrit également un équilibre exploration-exploitation, en explorant aléatoirement avec la probabilité ϵ , et en exploitant avec la probabilité $1 - \epsilon$. Dans notre implémentation, l'exploitation consiste à utiliser le concept du dernier cœur avec une valeur ϵ utilisée de 0.05.

4.2 Mesures de performance

La première mesure est le *regret cumulé*. Celui-ci mesure l'évolution du regret instantané (c'est-à-dire la différence entre le bien-être social maximal² du jeu et la somme des utilités réelles espérées des coalitions de la structure formée au

² Donné par la structure de coalitions qui maximise la somme des utilités de ses coalitions.

pas de temps t) au cours du temps. Cela permet de mettre en lumière la convergence du regret, c'est-à-dire le pas de temps à partir duquel les stratégies ont atteint leur équilibre exploration-exploitation et produisent donc un regret instantané constant. À un pas de temps t , le regret cumulé est la somme des regrets instantanés de chaque pas de temps $t' \leq t$. L'objectif de cette mesure est de pouvoir observer les différences entre les stratégies et non la valeur absolue. Formellement :

Definition 12 (Regret cumulé) Soit la solution optimale $S^* = (CS^*, \bar{x}^*)$ au sens du bien-être social, le regret cumulé à un pas de temps t , noté R_c^t , est défini tel que :

$$R_c^t = \sum_{t'=0}^t \left(\sum_{C^* \in CS^*} \mu_{C^*} - \sum_{C \in CS^t} \mu_C \right)$$

Enfin, afin d'évaluer l'apprentissage que les agents font de la fonction caractéristique réelle au cours du temps, nous utilisons l'*erreur moyenne absolue* (MAE) sur les utilités estimées et réelles des coalitions. Plus la MAE est proche de 0, plus la fonction caractéristique estimée est précise. La MAE est définie telle que :

Definition 13 (Erreur moyenne absolue)

Soient v et \hat{v} deux fonctions caractéristiques, l'erreur moyenne absolue (MAE) D_{MAE}^t entre v et \hat{v} au pas de temps t est définie telle que :

$$D_{MAE}^t = \frac{\sum_{C \in 2^N} |\hat{\mu}(C, t) - \mu_C|}{|2^N|}$$

4.3 Résultats

Les figures 1 et 2 montrent respectivement l'évolution des moyennes de l'erreur d'apprentissage et du regret cumulé de l'ensemble des jeux pour une configuration donnée (c'est-à-dire un facteur de transformation linéaire w) au cours des 100 pas de temps. La figure 3 synthétise les résultats avec le pourcentage relatif d'efficacité du λ -cœur contre ϵ -glouton pour les différents facteurs.

Concernant l'erreur d'apprentissage en figure 1, un premier point à souligner est que plus la fonction caractéristique est structurée (donc plus le facteur de transformation w est proche de 0), moins l'erreur d'apprentissage est grande. De manière générale, la stratégie ϵ -gloutonne est

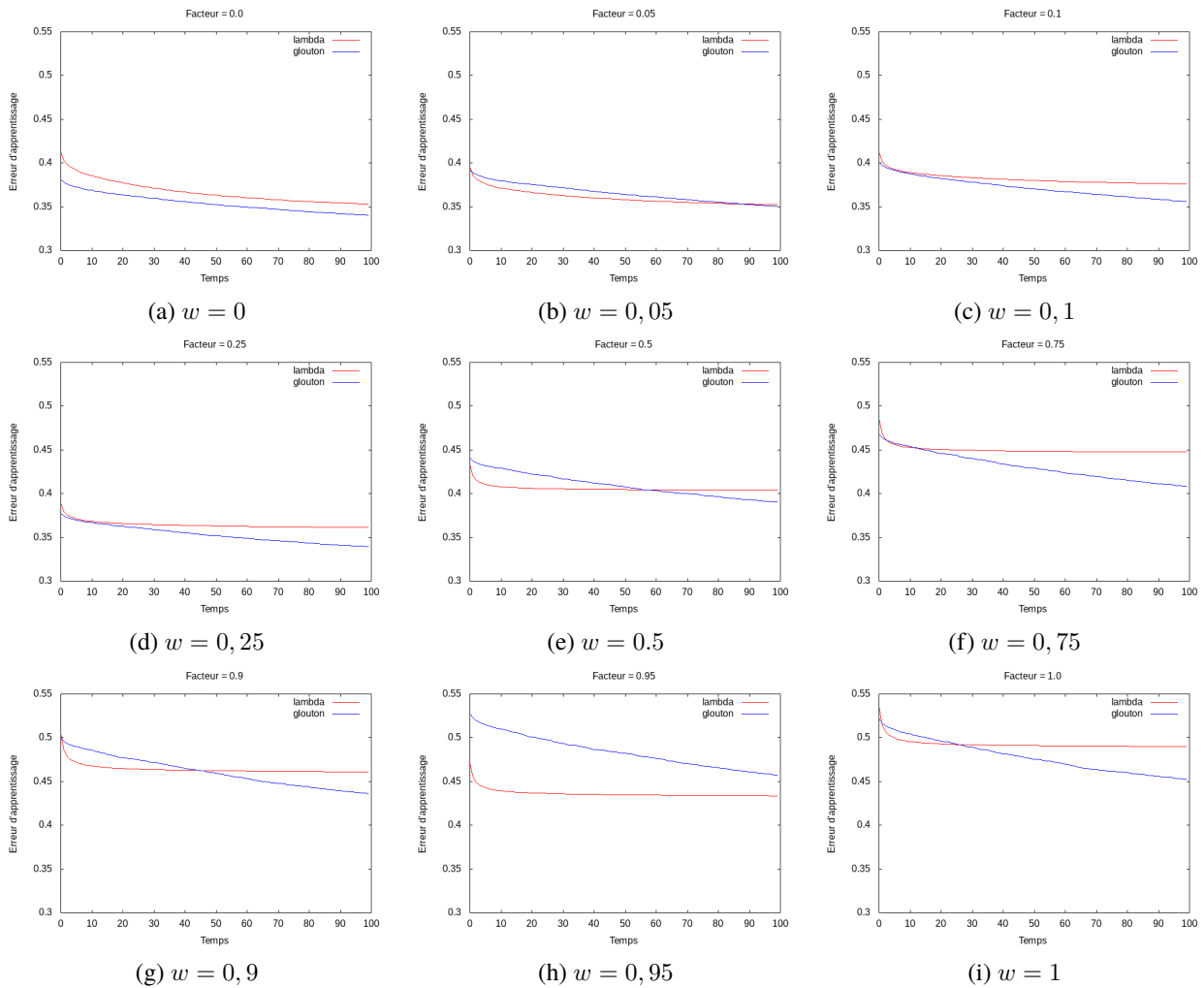


FIGURE 1 – Erreur d’apprentissage moyenne pour 6 agents sur 4200 jeux

celle qui apprend le mieux, avec quelques exceptions comme pour $w = 0,95$ où le λ -cœur permet d’apprendre mieux, ou bien $w = 0,05$ où les résultats des deux méthodes sont très proches. Toutefois, nous pouvons voir graphiquement une différence de comportement entre elles selon la structuration des fonctions caractéristiques. En effet, plus les fonctions caractéristiques sont structurées, plus la stratégie ϵ -gloutonne apprend entre le début et la fin des expérimentations. Par exemple, son erreur d’apprentissage diminue respectivement de 10,59%, 11,45% et de 13,20% pour $w = 0$, $w = 0,5$ et $w = 1$. Remarquons que cette diminution est quasi-linéaire avec la variation du facteur w . Concernant le λ -cœur, nous pouvons voir que l’apprentissage converge rapidement, dû au terme d’exploration $UCB-V$, et cela de plus en plus rapidement à mesure que les fonctions caractéristiques sont déstructurées. Par exemple,

pour $w = 0$, l’erreur d’apprentissage diminue tout au long de l’expérimentation, tandis que pour $w = 1$, l’erreur cesse quasiment de diminuer après le pas de temps $t = 20$. D’un point de vue plus général pour les deux méthodes, plus les fonctions caractéristiques sont déstructurées, plus l’erreur d’apprentissage est initialement importante.

Intéressons-nous ensuite au regret cumulé moyen en figure 2. Pour un facteur de transformation $w = 0$, c’est-à-dire avec une structure purement NDCS, le regret cumulé moyen est en faveur de la stratégie ϵ -gloutonne, tout comme pour un $w = 0,05$. Toutefois, à partir de $w = 0,1$, le λ -cœur est plus performant en termes de regret, et l’écart est plus important pour des valeurs de w plus grande. À partir de ces résultats, nous pouvons déduire que la stratégie ϵ -gloutonne est performante sur des fonctions

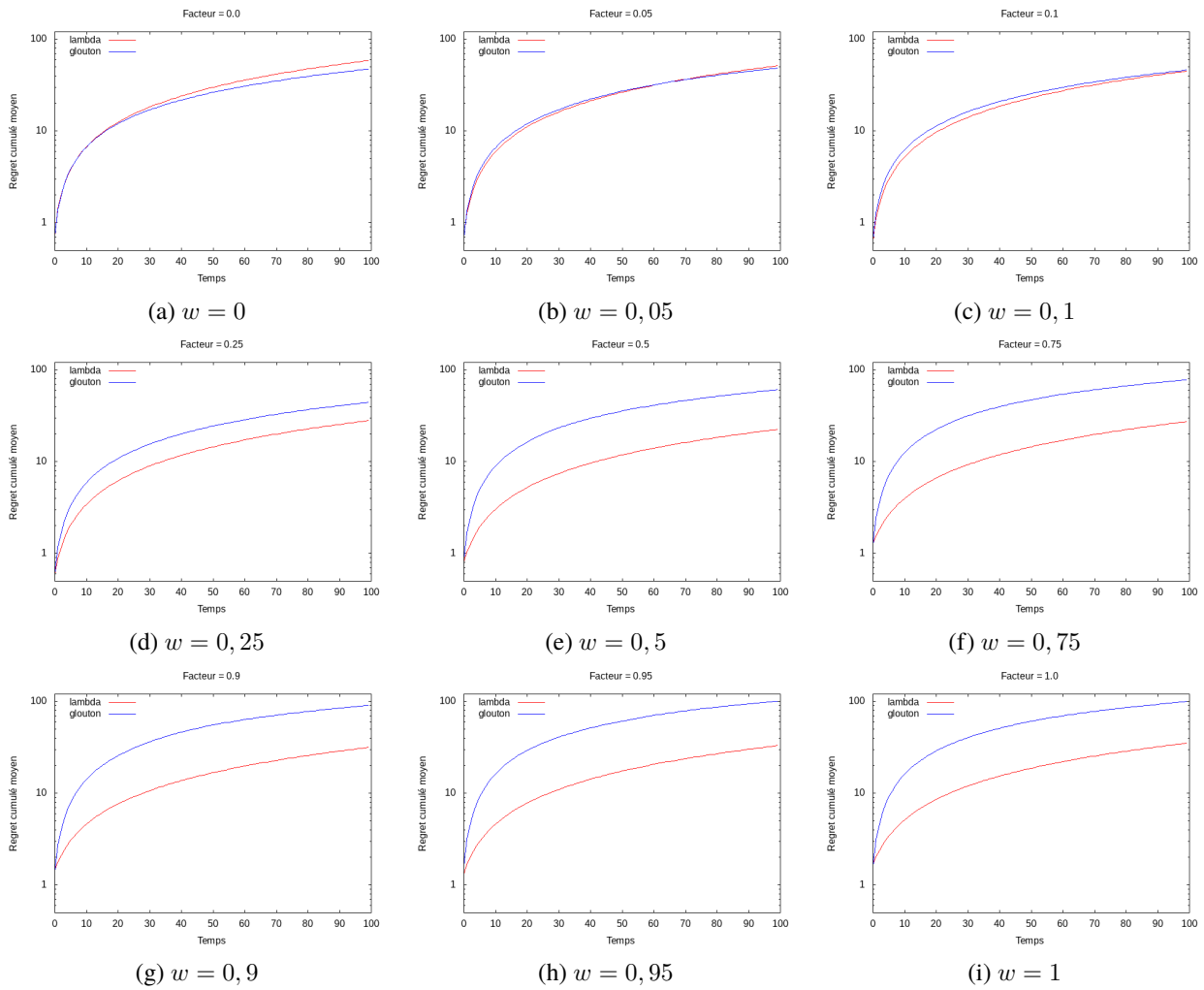


FIGURE 2 – Regret cumulé moyen pour 6 agents sur 4200 jeux

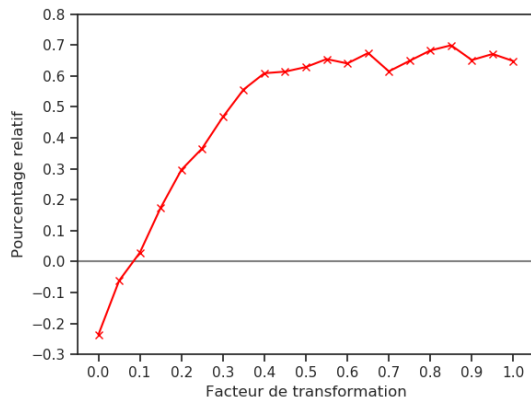


FIGURE 3 – Pourcentage relatif moyen d'efficacité de λ -cœur contre la stratégie ϵ -gloutonne

caractéristiques structurées mais que moins il y a de structuration, moins elle l'est. Il faut toutefois souligner que lorsque la stratégie ϵ -gloutonne est surpassée par le λ -cœur, c'est principalement ce dernier qui gagne en performance plus que la stratégie ϵ -gloutonne n'en perd. En effet, cette dernière obtient un regret cumulé moyen de 45,91 pour $w = 0$ et de 47,13 pour $w = 0,1$, soit une différence de 1,22. De son côté, le λ -cœur obtient un regret cumulé moyen de 58,28 pour $w = 0$ et de 44,58 pour $w = 0,1$, soit une différence de 13,70. Cet écart est de 16,07 pour $w = 0,25$, de 37,91 pour $w = 0,5$, de 35,11 pour $w = 0,75$, jusqu'à une différence de 64,39 pour $w = 1$. Dans ce dernier cas, le regret cumulé moyen pour le λ -cœur est de 34,98 alors qu'il est de 99,37 pour la stratégie ϵ -gloutonne. L'efficacité relative de λ -cœur contre la stratégie ϵ -gloutonne est mise en lumière sur la figure 3. Sur cette dernière, nous pouvons voir que l'écart

en faveur du λ -cœur ne fait qu'augmenter jusqu'à $w = 0,4$ puis se stabilise. Pour $w = 0$, λ -cœur est 23,66% moins efficace que ϵ -glouton. Il devient 2,9% plus efficace à partir de $w = 0,1$, jusqu'à 60,87% pour $w = 0,4$. Ensuite, pour $w \geq 0,4$, l'efficacité relative en faveur de λ -cœur se stabilise autour de 65%, avec un maximum de 69,90% pour $w = 0,85$. Ainsi, le concept de solutions λ -cœur se montre très performant sur des fonctions caractéristiques déstructurées, et reste plus performant que la stratégie ϵ -gloutonne tant que la structuration n'est pas importante. Il est toutefois nécessaire de noter que le λ -cœur est plus efficace sur des fonctions caractéristiques légèrement structurées. Par exemple, il obtient un regret cumulé de 22,39 avec $w = 0,5$, tandis que pour $w = 1$ celui-ci est de 34,98 (avec un minimum pour $w = 0,45$ avec 21,94 de regret cumulé).

5 Conclusion

Nous avons proposé le concept de solutions λ -cœur fondé sur un équilibre exploration-exploitation par l'intégration d'une notion d'intérêt pour les agents. En fixant cet intérêt à un biais d'exploration $UCB-V$ et définissant la composition comme étant une addition, nous avons montré que ce concept de solutions est performant sur les jeux de coalitions stochastiques répétés, notamment lorsque les fonctions caractéristiques ne sont pas très fortement structurées. Toutefois, le calcul du λ -cœur est long en raison de biais d'exploration. En effet, ce biais conduit le dernier cœur à avoir une valeur λ élevée, et donc à davantage parcourir l'espace des solutions car une approche naïve de ce calcul consiste à chercher des λ -cœur en incrémentant itérativement la valeur de λ . Il serait pertinent de travailler sur une approche distribuée ou décentralisée, mais qui introduit une nouvelle problématique : celle de l'observabilité partielle, propre à chaque agent, de la fonction caractéristique.

Références

- [1] R. Agrawal. Sample mean based index policies by $\mathcal{O}(\log n)$ regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4) :1054–1078, 1995.
- [2] J.Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theor. Comput. Sci.*, 410(19) :1876–1902, 2009.
- [3] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2) :235–256, 2002.
- [4] J.-P. Benoit and V. Krishna. Finitely repeated games. *Foundations in Microeconomic Theory*, pages 195–212, 1984.
- [5] M. Bettinelli, M. Ocello, and D. Genthial. ABSG : un modèle d'agent socialement inspiré pour la formation de coalitions. In *29e JFSMA*, pages 53–60, 2021.
- [6] B. Blankenburg, R. K. Dash, S. D. Ramchurn, M. Klusch, and N. R. Jennings. Trusted kernel-based coalition formation. In *4th AAMAS*, pages 989–996, 2005.
- [7] G. Chalkiadakis and C. Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *3rd AAMAS*, pages 1090–1097, 2004.
- [8] G. Chalkiadakis and C. Boutilier. Sequential decision making in repeated coalition formation under uncertainty. In *7th AAMAS*, pages 347–354, 2008.
- [9] 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.
- [10] G. Chalkiadakis, E. Markakis, and C. Boutilier. Coalition formation under uncertainty : Bargaining equilibria and the Bayesian core stability concept. In *6th AAMAS*, pages 1–8, 2007.
- [11] A. Charnes and D. Granot. Prior solutions : Extensions of convex nucleus solutions to chance-constrained games. Technical report, Texas Univ., 1973.
- [12] A. Charnes and D. Granot. Coalitional and chance-constrained solutions to n-person games. i : The prior satisficing nucleolus. *SIAM J. Appl. Math.*, 31(2) :358–367, 1976.
- [13] J. Guéron and G. Bonnet. De la diversité des jeux de coalitions à utilité transférable. In *29e JFSMA*, pages 149–158, 2021.
- [14] J. Guéron and G. Bonnet. Un protocole de concessions monotones pour la formation distribuée de coalitions. In *30e JFSMA*, pages 31–40, 2022.
- [15] S. Jeong and Y. Shoham. Bayesian coalitional games. In *23rd AAAI*, pages 95–100, 2008.
- [16] H. Konishi and D. Ray. Coalition formation as a dynamic process. *Journal of Economic theory*, 110(1) :1–41, 2003.
- [17] A. Louati, J. El Haddad, and S. Pinson. Formation de coalitions pour une composition de services web fondée sur la confiance dans les réseaux sociaux. In *25e JFSMA*, pages 149–158, 2017.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] L. Tran-Thanh, A. Chapman, E. M. De Cote, A. Rogers, and N. R. Jennings. Epsilon-first policies for budget-limited multi-armed bandits. In *24th AAAI*, pages 1211–1216, 2010.

Vérification formelle de propriétés de vivacité pour des SMA stochastiques à l'aide de GDT

Mathias Déhais^a Bruno Mermet^b Grégory Bonnet^a
 mathias.dehais@unicaen.fr bruno.mermet@unicaen.fr gregory.bonnet@unicaen.fr

^aNormandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC

^bNormandie Univ, UNIHAVRE, ENSICAEN, CNRS, GREYC

Résumé

Cet article traite de la preuve formelle de systèmes multi-agents dont les comportements peuvent être caractérisés de manière probabiliste. Nous nous fondons sur le modèle GDT4MAS qui aborde la vérification formelle de la spécification jusqu'à la génération d'obligations de preuve en s'appuyant sur la logique du premier ordre, ce qui lui confère une expressivité importante. Le modèle GDT4MAS ne traite cependant pas des comportements stochastiques et ne permet pas de prouver des propriétés de vivacité. Nous étendons le modèle GDT4MAS en permettant de modéliser des agents pouvant réaliser des actions aux effets stochastiques et redéfinissons les opérateurs dans ce cadre. Afin de prouver des propriétés de vivacité, nous nous appuyons sur des méthodes de variant adaptées à notre cadre et montrons comment prouver par exemple la terminaison presque sûre d'un SMA ou la récurrence d'une propriété particulière.

Mots-clés : Vérification formelle, Systèmes multi-agents, Comportements stochastiques

Abstract

This article deals with the formal proof of multi-agent systems whose behaviors can be characterized probabilistically. We consider the GDT4MAS model which addresses formal verification from specification to proof obligation generation by relying on first-order logic, which gives it an important expressiveness. However, the GDT4MAS model does not deal with stochastic behavior and does not allow the proof of liveness properties. We then extend GDT4MAS to model agents that can perform actions with stochastic effects and redefine the operators in this framework. To prove liveness properties, we rely on variant methods adapted to our framework and show how to prove the MAS almost sure termination, or the recurrence of a property.

Mots-clés : Formal verification, Multi-agent systems, Stochastic behaviours

1 Introduction

Les systèmes multi-agents (SMA) consistent en des agents percevant, agissant et interagissant avec leur environnement. Nous nous intéressons dans cet article aux systèmes multi-agents stochastiques (SMAS), des SMA dans lesquels le comportement et l'ordonnancement des agents peuvent être caractérisés de manière probabiliste. Il est important de noter que cette notion de stochasticité peut être introduite et observée dans les SMA à différents endroits, résumés dans la table 1. En effet, le choix de la prochaine action pour chaque agent peut suivre une certaine loi de probabilité, qui peut dépendre de l'environnement. De plus, le résultat de l'action en lui-même peut être de nature probabiliste tout comme l'ordre d'exécution des agents. Enfin, le système peut être ouvert, c'est-à-dire des agents peuvent apparaître ou disparaître selon une certaine loi de probabilité. Dans les systèmes centralisés, la stochasticité est étudiée depuis longtemps. Parallèlement, plusieurs modèles ont été développés pour assurer la vérification formelle des SMA. La combinaison des deux commence à être étudiée avec des méthodes comme le *model checking* mais n'a, à notre connaissance, pas encore été abordée via la preuve de théorèmes.

Source potentielle de stochasticité	Considérée ?
Choix de l'action suivante	oui
Résultat d'une action	oui
Ordre d'exécution des agents	non
Création / destruction d'agents	non

TABLE 1 – Sources de stochasticité potentielles considérées dans cet article.

Dans cet article, nous considérons le modèle GDT4MAS utilisé pour prouver formellement des SMA, et nous l'étendons aux sources de stochasticité de la table 1. Nous proposons égale-

ment une adaptation de méthodes connues pour prouver des propriétés de vivacité. Cet article est organisé comme suit. Dans la section 2, nous présentons l'état de l'art sur la vérification formelle des systèmes multi-agents et sur les systèmes stochastiques. La section 3 présente le modèle GDT4MAS et la section 4 son extension probabiliste. La section 5 aborde la preuve des propriétés de vivacité.

2 État de l'art

2.1 Vérification formelle de SMA

La preuve de systèmes multi-agents a déjà été abordée sous plusieurs angles, dont le *model checking*. Par exemple Lomuscio *et al.* [17, 16] ont développé MCMAS, un *model checker* conçu pour vérifier des SMA. Dans cet article, nous nous concentrons sur la preuve de théorèmes car le *model checking* pose des problèmes de recherche exhaustive. La méthode B est l'une des principales méthodes utilisant la démonstration de théorème [1]. Cette méthode, et son extension B-événementiel, reposent sur le principe de raffinement, c'est-à-dire travailler à partir d'un niveau très abstrait et spécifier le système vers un niveau plus concret en s'assurant que chaque étape du processus est vérifiée et correcte. Une problématique identique à la nôtre s'est alors posée, i.e. considérer des comportements multi-agents ainsi que stochastiques. Pour l'aspect multi-agent, les travaux en B-événementiel [7] utilisent des solutions ad hoc et il y existe peu d'approches différentes, dont les principales sont données ci-après. Esteva *et al.* [5] ont proposé ISLANDER, conçu pour la spécification et la vérification des institutions électroniques gérées par des agents. Les institutions représentent les règles du jeu dans une société, bien adaptées pour faire face à la conception de SMA ouverts. ISLANDER se concentre sur les aspects sociaux en se référant à l'infrastructure des institutions plutôt qu'aux aspects internes des agents. Les agents sont abstraits et l'utilisateur peut choisir leur architecture et leur langage. Contrairement à ISLANDER, Bracciali *et al.* [23] ont proposé PROSOCS, qui spécifie les aspects internes des agents et utilise un modèle d'agence KGP (Knowledge, Goals, Plan). Mermet et Simon [19] ont proposé GDT4MAS, qui s'appuie sur la logique du premier ordre pour spécifier et vérifier des SMA. GDT4MAS spécifie le comportement de l'agent avec des opérateurs et des actions, en s'appuyant sur les objectifs de chaque agent pour les spécifier. Des

obligations de preuve peuvent être générées afin de prouver formellement que le comportement de l'agent est correct. Ceci est rendu possible en spécifiant des arbres de décomposition de buts (GDT, *Goal Decomposition Tree*). Un GDT est la spécification de la manière dont l'objectif principal d'un agent doit être atteint, par une combinaison d'actions via des opérateurs, formant un arbre. Ces trois modèles n'intègrent pas de stochasticité. De plus, dans ISLANDER, seule la partie des comportements décrits par l'institution peut être vérifiée et non tous les comportements des agents. Dans PROSOCS, seules les propriétés exprimées en logique propositionnelle peuvent être vérifiées. Ceci est moins expressif que la logique du premier ordre qui est utilisée dans GDT4MAS. De plus, le modèle GDT4MAS prend en compte l'ensemble du processus de preuve de théorème : spécification, génération d'obligations de preuve et preuve en elle-même. Par conséquent, dans cet article, nous nous concentrons sur l'extension de GDT4MAS aux comportements stochastiques et à la preuve de propriétés de vivacité.

2.2 Vérification formelle stochastique

Hors du domaine des SMA, il existe de très nombreux travaux concernant la vérification formelle de systèmes stochastiques. Kwiatkowska *et al.* [12] ont développé, et étudient toujours [14], le *model checking* probabiliste avec le logiciel PRISM. Ce travail repose sur des chaînes de Markov en temps discret, des processus de décision de Markov et des jeux multi-joueurs stochastiques. La logique sous-jacente utilisée pour raisonner sur ces structures est classiquement PCTL, une extension probabiliste de la logique temporelle arborescente (CTL). Ils ont récemment étendu leurs travaux à des logiques plus puissantes telles que ATL* et PCL (*Probabilistic Strategy Logic*), une logique proche de la logique du premier ordre. Ils modélisent également des jeux, en raisonnant avec la logique rPATL, qui introduit la notion de récompense [13]. Le lecteur intéressé peut se référer à [6]. Ces méthodes sont destinées à explorer tous les cas possibles et à donner des probabilités explicites que les différentes propriétés d'un système soient vraies. En ce qui concerne la vérification de SMA, le principal problème reste le passage à l'échelle. McIver et Morgan [18] ont étudié les systèmes stochastiques à la fois vus comme des programmes pouvant être décrits dans un langage opératoire, et comme des fonctions. Dans cette théorie, les prédicats sont

étendus avec la notion d'*expectations*. Une *expectation* permet de raisonner sur la probabilité qu'une propriété soit vraie après l'exécution d'un programme stochastique. Plus généralement, les recherches récentes [8] ont tendance à s'appuyer sur les travaux de McIver et Morgan ainsi que la méthode B [3].

Aouadhi *et al.* [2] ont étendu la méthode B-événementiel. Ce modèle ne s'appuie pas sur les travaux de McIver et Morgan [18] mais introduit des probabilités explicitement dans les machines B, en implémentant des événements probabilistes. Un événement probabiliste est un événement qui peut avoir plusieurs résultats possibles avec des probabilités associées à chaque résultat.

2.3 Vérification formelle de SMAS

Les travaux précédents ont été adaptés aux SMA, l'aspect *model checking* est donc bien développé depuis plusieurs dizaines d'années, [4] et la vérification de propriétés de vivacité est également un sujet de recherche depuis de nombreuses années, [14]. Récemment, Lomuscio et Pirovano [15] ont traité de la vérification formelle pour SMAS en raisonnant sur un fragment de la logique PCTL. Pirovano [22] a amélioré la méthode afin de travailler sur le problème du passage à l'échelle. *Model checking* et preuve de théorèmes ont toujours avancé côte à côte, partageant les mêmes objectifs avec des outils distincts. Bien que des avancées aient été faites, le passage à l'échelle reste toujours un problème en *model checking*, c'est pourquoi il nous paraît important de poursuivre les travaux par la preuve de théorèmes.

2.4 Propriétés de vivacité

Les propriétés que nous souhaitons vérifier sur des systèmes sont généralement divisées en *propriétés de sûreté*, c'est-à-dire que quelque chose ne se produira jamais, et en *propriétés de vivacité*, c'est-à-dire que quelque chose finira par se produire. Le modèle GDT4MAS permet la preuve de propriétés de sûreté en permettant la vérification de la préservation des invariants tout au long de l'exécution du système, mais la preuve de propriété de vivacité n'a pas encore été implémentée. En B événementiel, les propriétés de vivacité sont prises en charge à l'aide de *model-checking*. Hoang et Abrial [9] permettent certaines preuves en B-événementiel grâce à des méthodes de variant. Hudon *et al.*

ont également développé une extension de B-événementiel, unit-B [10], pour tenter d'instaurer les preuves de propriétés de vivacité au coeur du modèle.

3 Introduction à GDT4MAS

Étant donnés les éléments précédents, nous choisissons d'étendre le modèle GDT4MAS avec des considérations probabilistes. Nous présentons d'abord des éléments sur le modèle général. Nous renvoyons les lecteurs intéressés vers [19] pour plus de détails. Comme dans des modèles similaires, comme B-événementiel, le processus de preuve se déroule en 3 phases : (1) la spécification du système qui comprend notamment la définition des invariants ; (2) la création d'obligations de preuve qui, une fois prouvées, garantissent que toutes les propriétés spécifiées sur le système sont vraies ; (3) la preuve des propriétés, déléguée à un prouveur externe comme PVS.

3.1 Logique temporelle sous-jacente

GDT4MAS utilise la logique temporelle linéaire (LTL) pour donner une sémantique opérationnelle aux GDT. Étant donné un ensemble de variables booléennes V , les formules LTL sur V sont les formules construites à partir des variables de V , les connecteurs propositionnels usuels ($\neg, \wedge, \vee, \rightarrow$) et les opérateurs standards \square, \diamond et \circ . Les formules LTL sont évaluées par rapport aux *traces* qui sont des séquences infinies de mondes $(\omega_1, \omega_2, \omega_3, \dots)$, où chaque monde est étiqueté avec une interprétation des variables. Étant donné un monde ω , $Path(\omega)$ est l'ensemble de toutes les traces possibles commençant par le monde ω . La validité d'une formule LTL ϕ est évaluée par rapport à une trace τ et un monde ω dans celle-ci, selon la sémantique standard, notée $\tau, \omega \models \phi$. Soit ω un monde et ψ une formule LTL, alors $\omega \models \psi$ si et seulement si $\tau, \omega \models \psi$ pour tous les $\tau \in Path(\omega)$.

3.2 Agent et arbre de décomposition de buts

Dans GDT4MAS, le comportement des agents est représenté par des GDT. Ce sont des arbres dont les nœuds sont des buts, définis par une condition de satisfaction et associés soit à des actions atomiques, soit à des décompositions en sous-but. Le GDT d'un agent spécifie l'ensemble de son comportement, et la condition de satisfaction de son nœud racine est donc son objectif principal. De manière intéressante, il

est possible de spécifier plusieurs agents ayant le même comportement avec un même GDT, ce qui permet de vérifier tous les agents avec les mêmes preuves. Nous ne donnons ci-dessous que les notions pertinentes pour cet article. La logique des prédicats est utilisée dans la phase de spécification essentiellement pour exprimer les conditions de satisfaction des objectifs, les conditions préalables des actions et les conditions de branchement. Nous notons cette logique \mathcal{L} . La phase de conception nécessite de pouvoir exprimer des objectifs dont le succès dépend de l'évolution de certains prédicats, comme incrémenter une valeur. Pour cela, nous utilisons une extension, notée \mathcal{L}' , de la logique choisie dans laquelle nous nous référons à deux instants dans le temps. Dans \mathcal{L}' , les variables primées désignent les valeurs dans le second instant tandis que celles non primées désignent le premier.

Définition 1 (Environnement). Un *environnement* est un tuple $\epsilon = (V_\epsilon, i_\epsilon)$, où V_ϵ est un ensemble de variables, et $i_\epsilon \in \mathcal{L}$ désigne les invariants de l'environnement.

Les agents interagissent avec l'environnement en modifiant les variables d'environnement. Les invariants d'environnement sont des formules sur ces variables dont il faut prouver qu'elles restent vraies tout au long de l'exécution afin de vérifier que la spécification est correcte.

Définition 2 (Agent). Soit ϵ un environnement. Un *agent* A relatif à ϵ est un tuple :

$$(V_i(A), V_\epsilon(A), init_A, i_A, Actions_A, Beh_A)$$

Chaque agent est décrit grâce à un ensemble de variables internes $V_i(A)$, un ensemble de variables d'environnement $V_\epsilon(A)$ sur lesquelles il peut agir, et un ensemble d'actions $Actions_A$ qu'il peut effectuer. $init_A$ correspond à une fonction qui associe une valeur initiale aux variables internes. i_A est les invariants de l'agent que nous voulons vérifier. Un comportement Beh_A modélisé par un GDT décrit alors son comportement.

Définition 3 (GDT). Soit ϵ un environnement et A un agent. Un *GDT* pour A dans ϵ est un triplet $(pre_T, tc_T, Root_T)$, où $pre_T, tc_T \in \mathcal{L}$ et $Root_T$ est un nœud interne (donc un arbre). pre_T est la condition préalable et tc_T est le contexte de déclenchement, ce qui signifie que le GDT ne peut s'activer que s'il est vrai.

Les GDT sont constitués de feuilles et de nœuds internes. Dans un GDT, les nœuds correspondent

aux buts de l'agent. Une condition de satisfaction (SC) est associée à chaque nœud N . Intuitivement, un but est satisfait si et seulement si sa SC est rendue vraie. La SC du nœud à la racine est alors le but principal de l'agent, décomposé grâce à des opérateurs en sous-buts. Enfin, toujours pour le processus de preuve, il est nécessaire de savoir ce qu'il se passe si l'exécution du nœud échoue. Cela se traduit par une Propriété Garantie en cas de d'Échec (GPF).

La sémantique opérationnelle des GDT est donnée par des règles en LTL. Afin d'exprimer l'évolution de l'exécution des GDT le modèle utilise des atomes LTL. Soit N un nœud, les atomes $\{init, in, end\}$ dénotent respectivement que l'on commence à exécuter N , que l'on est en train d'exécuter N et que l'on a fini de l'exécuter.

Définition 4 (nœud interne). Soit A un agent dans un environnement ϵ . Un *nœud interne* N (d'un GDT pour A dans ϵ) est un tuple :

$$(name_N, Op_N, Children, sc_N, gpf_N)$$

où $sc_N, gpf_N \in \mathcal{L}'$, Op_N est un opérateur de décomposition, et $Children_N$ est une séquence de nœuds internes et de nœuds feuilles dont la longueur correspond à l'arité de Op_N .

Chaque nœud interne d'un GDT est associé à une décomposition en sous-buts via un opérateur. Par exemple, l'opérateur SEQAND est un opérateur AND logique paresseux et ordonné. Comme autre exemple, l'opérateur OR permet de choisir de manière non déterministe entre les sous-objectifs, et de continuer jusqu'à ce que l'un réussisse. Les *nœuds feuilles* se comportent comme des nœuds internes sauf qu'ils contiennent une *action* et pas d'opérateur de branchement.

4 GDT4MAS probabiliste

Pour le modèle probabiliste, il est nécessaire d'étendre la logique LTL avec un opérateur probabiliste.

4.1 LTL Probabiliste

Comme dans la logique PCTL, où la mesure de probabilité est définie sur des ensembles de formules de chemins, nous considérons les probabilités dans la logique LTL en s'intéressant tout d'abord à l'opérateur $\mathcal{P}_{\sim\lambda}$ sur les formules LTL, où $\sim \in \{<, \leq, \geq, >, =\}$ et λ est un seuil de probabilité. Les formules de cette logique doivent être interprétées sur des chaînes de

Markov discrètes, qui pourraient être instanciées pour chaque SMAS. Tous les mondes possibles sont des états et les transitions probabilistes sont données par les probabilités associées aux actions et aux opérateurs. Ensuite, nous définissons une mesure de probabilité μ_ω sur un ensemble de traces, qui est d'abord définie sur toute trace finie commençant par ω , puis étendue de manière unique à toutes les traces commençant par ω . Pour plus de détails, le lecteur peut se référer à [11]. Ainsi, étant donné un monde ω , une formule LTL ψ , $\lambda \in [0, 1]$, nous avons :

$$\omega \models \mathcal{P}_{\sim\lambda}(\omega, \psi) \text{ si } \mu_\omega(\{\tau \in \text{Path}(\omega) \mid \tau \models \psi\}) \sim \lambda$$

qui spécifie de manière informelle la probabilité que la formule ψ soit valide pour $\sim \lambda$.

Exemple 1. Soit un SMAS qui inclut un nœud N et un monde initial ω_{init} . Alors, $\omega_{init} \models \mathcal{P}_{>0.5}(\omega_{init}, \diamond in_N)$ signifie que la probabilité que le système exécute finalement le nœud N est supérieure à 0,5.

Dans la version probabiliste de GDT4MAS, nous rajoutons les opérateurs probabilistes P_{OR} et P_{AND} , qui sont des versions probabilistes des opérateurs OR et AND.

Définition 5 (Opérateur probabiliste OR). L'opérateur *probabiliste OR*, noté P_{OR} , décompose un but en k sous-buts, les exécutions sont choisies de manière probabiliste jusqu'à ce que l'un d'eux réussisse. L'ordre est donné par un ensemble de probabilités P_{op} . À chaque sous-nœud N_i une probabilité $P_{N_i} \in P_{op}$ est associée et il faudra prouver que la somme des P_{N_i} vaut 1. Si toutes les exécutions échouent, l'exécution de l'objectif parent se termine. Pour marquer qu'un sous-but a été tenté, une variable temporelle $done_{N_i}$ est utilisée. La sémantique de cet opérateur est donnée par une conjonction de formules dont nous ne donnons ici que celles contenant des considérations probabilistes :

1. $\Box(\forall(i \in \llbracket 1, k \rrbracket)).(init_N \rightarrow \mathcal{P}_{=P_{N_i}}(init_{N_i}))$
2. $\Box(\forall(i \in \llbracket 1, k \rrbracket)).(end_{N_i} \wedge \neg sat_{N_i} \wedge \exists(j \in \llbracket 1, k \rrbracket)).\neg done_{N_j} \rightarrow \mathcal{P}_{=P_j}(\bigcirc_A(init_{N_j}))$

où :

$$P_j = P_{N_j} \times \frac{1}{\sum_{\substack{l \in \llbracket 1, k \rrbracket \\ \neg done_{N_l}}} P_{N_l}}$$

correspond à la probabilité que N_j soit exécuté au monde suivant sachant quels nœuds ont déjà été exécutés.

La formule 1 signifie que chaque sous-nœud N_i a une probabilité P_{N_i} de s'exécuter en premier. La formule 2 signifie que lorsqu'un sous-nœud termine son exécution sans atteindre le sous-objectif, la probabilité qu'un nœud donné N_i soit exécuté vaut P_j , sa probabilité initiale divisée par la somme des probabilités des sous-nœuds non-exécutés.

L'opérateur P_{AND} est défini de manière parfaitement similaire à l'exception que l'exécution de l'objectif parent se termine avec succès si tous les nœuds réussissent et échoue si l'exécution de l'un des nœuds fils échoue.

Définition 6 (Action probabiliste). Une *action probabiliste* a est un tuple :

$$(pre_a, POST_a, GPF_a, P_a)$$

où pre_a est la formule qui doit être vraie avant qu'une action ait lieu. $POST_a = \{post_{a,i} \mid i \in \llbracket 1, n \rrbracket\}$ avec $n \in \mathbb{N}$ étant le nombre de réussites. Chaque $post_{a,i}$ représente une formule qui devient potentiellement vraie si l'action réussit. $GPF_a = \{gpf_{a,i} \mid i \in \llbracket 1, k \rrbracket\}$ avec $k \in \mathbb{N}$ étant le nombre de résultats infructueux. Chaque $gpf_{a,i}$ représente une formule qui peut devenir fautive si l'action échoue. À chaque résultat est assignée une fonction, qui est contenue dans l'ensemble $P_a = \{p_{post_{a,i}} \mid i \in \llbracket 1, n \rrbracket\} \cup \{p_{gpf_{a,i}} \mid i \in \llbracket 1, k \rrbracket\}$ où n et k sont tels que définis précédemment. Notons par la suite $OUTCOME_a$ l'union de $POST_a$ et GPF_a . Notons F_A l'ensemble des applications de $V_i(A)$ vers les valeurs. Chaque $p_{post_{a,i}}$ et $p_{gpf_{a,i}}$ est une fonction de F_A dans \mathbb{R} , ce qui signifie que la probabilité de chaque résultat dépend de la valeur des variables.

La sémantique d'une action probabiliste a est qu'un agent ne peut tenter d'effectuer a à un instant donné que si, et seulement si, pre_a est vraie au même instant. Le résultat de l'action est probabiliste, et les probabilités $p_i \in P_a$ dépendent des valeurs des variables. Un seul résultat de $OUTCOME_a$ est garanti comme vrai après l'exécution de a et il est choisi en fonction des probabilités P_a .

Remarque 1. Les résultats de $POST_a$ et GPF_a ne sont pas exclusifs. En effet, plusieurs résultats peuvent être vrais par effet de bord mais un seul est garanti vrai selon les probabilités P_a .

Remarque 2. Une action peut ne jamais échouer. C'est pourquoi il est possible d'avoir $GPF_a = \emptyset$. Cependant, il doit y avoir au moins un résultat où elle réussit.

Afin de vérifier des propriétés de vivacité, nous aurons besoin de déterminer si la probabilité qu'une action s'exécute une fois son GDT activé soit strictement positive.

Définition 7 (Action accessible). Une action a est dite *accessible* si la probabilité d'exécuter le nœud feuille correspondant est non nulle. Notons N le nœud feuille associé à a et tc , le contexte déclencheur du GDT contenant N . L'action a est accessible si, et seulement si, $\Box(tc \Rightarrow \mathcal{P}_{>0}(\Diamond init_N))$.

Nous pouvons désormais définir un système multi-agents stochastique.

Définition 8 (Système multi-agents stochastique). Un *système multi-agents stochastique* est un système multi-agents dont les agents peuvent avoir des comportements décrits par des actions probabilistes et des opérateurs probabilistes.

Remarque 3. Un GDT peut contenir à la fois des actions ou des opérateurs probabilistes et non déterministes comme dans le modèle de base.

Prenons en exemple un SMAS avec deux types d'agents, un producteur et un consommateur.

Variabes. Soit $S_E \in \mathbb{N}$, une variable d'environnement qui représente le stock consommé par le *Consommateur*. Soit $S_{pro} \in \mathbb{N}$, une variable interne du *Producteur* qui représente son stock. Soit $S_{cons} \in \mathbb{N}$, une variable interne du *Consommateur* qui représente son stock.

Contextes de déclenchement.

$$tc_{pro} = (S_{pro} > 0) \text{ et } tc_{cons} = (S_E > 1)$$

Comportements. Les GDT des deux agents sont illustrés en figure 1. Ici, (a_1, a_2, a_3, a_4) font référence aux actions sur les nœuds feuilles. Les détails des actions sont donnés sur la table 2.

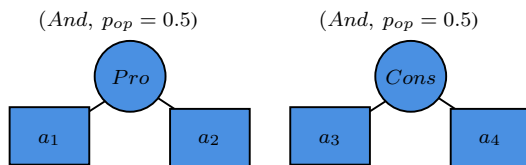


FIGURE 1 – GDT Producteur / Consommateur

4.2 Obligations de preuve et vérification

Les différentes obligations de preuve (PO) sont générées grâce à des schémas de preuve. Fondamentalement, un schéma de preuve est un patron

TABLE 2 – Actions des deux agents

a_1	$post_1 = (S'_E = S_E + 1)$
a_2	$post_1 = (S'_{pro} = S_{pro} - 1)$ $p_{post1} = \frac{2}{3}$ $gpf_1 = (S'_{pro} = S_{pro})$ $p_{gpf1} = \frac{1}{3}$
a_3	$pre = S_E > 0$ $post_1 = (S'_E = S_E - 2)$ $p_{post1} = \frac{S_E}{1+S_E}$ $post_2 = (S'_E = S_E - 1)$ $p_{post2} = \frac{1}{1+S_E}$
a_4	$post_1 = (S'_{cons} = S_{cons} + 1)$

de PO, et ils doivent être conçus de telle sorte que si toutes les PO sont instanciées et vérifiées, l'ensemble du système est correctement prouvé. L'objectif principal de la génération de PO est de vérifier que chaque invariant reste vrai pendant l'exécution du système. De manière générale, cette vérification peut être effectuée automatiquement par un démonstrateur de théorème comme PVS. Les lecteurs intéressés peuvent se référer à [21].

Un *contexte* est ajouté à presque toutes les obligations de preuve. En effet, certaines informations peuvent être déduites de chaque nœud et peuvent être utilisées pour générer des obligations de preuve plus facilement prouvables, puisque les contextes et les GPF seront des hypothèses supplémentaires.

Définition 9 (Contexte). Soit T un GDT, et soit N un nœud de T . Le *contexte* de N , noté c_N , est une formule de \mathcal{L} qui est logiquement équivalente à la conjonction de toutes les formules $\psi \in \mathcal{L}$ telles que pour tous les mondes ω satisfaisant $init_N$, ω satisfait ψ .

Nous renvoyons le lecteur à [20] pour plus de détails sur le calcul des contextes. Le point essentiel dans cet article est que les contextes fournissent des informations qui peuvent aider durant la phase de preuve.

La version probabiliste du modèle nécessite l'introduction d'invariants liés à la définition stochastique des actions et des opérateurs. En effet, il faut préserver le fait que les probabilités soient des nombres réels compris entre 0 et 1 et qui somment à 1 pour chaque action et opérateur à chaque instant. Ces invariants sont donnés dans la table 3.

TABLE 3 – Invariants relatifs aux probabilités

Nom	formule
INV.proba1	$\forall p \in P_a, 0 \leq p \leq 1$
INV.proba2	$\sum_{p,p \in P_a} = 1$
INV.proba3	$\forall p \in P_{op}, 0 \leq p \leq 1$
INV.proba4	$\sum_{p,p \in P_{op}} = 1$

Nous n'avons pas besoin d'introduire explicitement de nouveaux schémas de preuve dans le modèle pour prouver que les probabilités sont correctes. En effet, nous nous appuyons sur le fait que les invariants doivent être préservés par les PO. Cependant, nous devons adapter les schémas de preuve existants pour que notre dernière phrase soit vraie. En ce qui concerne les PO associées aux actions, nous devons montrer que leurs nœuds feuilles respectent les invariants d'agents et d'environnement. Cela est capturé par des schémas de preuve d'action définis comme suit :

Définition 10 (Schémas de preuve d'action). L'exécution d'une action doit préserver les invariants. Pour chaque $outcome_a \in OUTCOME_a$:

$$i_\epsilon \wedge i_A \wedge c_N \wedge outcome'_a \models i'_\epsilon \quad (1)$$

$$i_\epsilon \wedge i_A \wedge c_N \wedge outcome'_a \models i'_A \quad (2)$$

où c_N est le contexte d'un nœud, c'est-à-dire toutes les formules qui sont garanties vraies au moment de l'exécution du nœud N .

Ces schémas de preuve signifient que, quel que soit le déroulement de l'exécution de l'action, celle-ci doit préserver les invariants. Il faut aussi montrer que, si l'action réussit, la condition de satisfaction associée aux nœuds feuilles est vérifiée (et la GPF sinon).

Nous devons également montrer que les actions induisent correctement le succès ou l'échec du nœud feuille correspondant. Ceci est capturé par des schémas de preuve de nœuds feuilles définis comme suit :

Définition 11 (Schémas de preuve de nœuds feuille).

$$i_\epsilon \wedge i_A \wedge c_N \models pre_a \quad (3)$$

Pour chaque $post_a \in POST_a$,

$$i_\epsilon \wedge i_A \wedge c_N \wedge post'_a \models sc'_N \quad (4)$$

Pour chaque $gpf_a \in GPF_a$,

$$i_\epsilon \wedge i_A \wedge c_N \wedge gpf'_a \models gpf'_N \vee sc'_N \quad (5)$$

Ces schémas de preuve signifient que si une action réussit (ou échoue), alors le nœud feuille correspondant doit voir sa condition de satisfaction (ou sa GPF) vérifiée.

Reprenons l'exemple du SMAS producteur/consommateur donné en section 4.

Preuve de probabilités. Il nous faut montrer que les probabilités somment bien à 1 pour chaque action et restent comprises entre 0 et 1. Il est facile de vérifier que les probabilités somment toujours à 1 dans les deux types d'agents. Pour a_3 , nous utilisons *pre* pour affirmer que les probabilités restent entre 0 et 1 et cela est trivial pour a_2 . Pour des raisons de place, nous ne détaillons pas ici les PO instanciées.

Nous avons décrit le modèle GDT4MAS adapté aux systèmes multi-agents stochastiques. Nous pouvons maintenant permettre la vérification de certaines preuves de vivacité en nous appuyant sur la définition de l'accessibilité d'une action, qui est caractérisée grâce à des distributions de probabilités.

5 Propriétés de vivacité

Nous tirons parti du modèle probabiliste pour permettre la preuve de propriétés de vivacité grâce à la notion d'action accessible. Nous proposons une approche similaire à celle Hoang et Abrial [9] puisque cela ne nécessite pas d'ajouter plus d'outils au modèle.

5.1 Méthodes de variant

Nous adaptons ici l'approche de Hoang et Abrial [9] à notre modèle. Nous utilisons toujours la logique LTL, mais cette fois, les éléments de base du langage sont les propriétés $P \in \mathcal{L}$ puisque les propriétés que nous considérons ici reposent sur les variables des systèmes et non sur l'ensemble $\{in, sat, end, \dots\}$ présenté de manière informelle en définition 3.2. Ces propriétés sont utilisées comme des formules d'état. Soit $P \in \mathcal{L}$, les mondes ω satisfaisant la propriété P sont appelés P -mondes. Pour une trace $\sigma = (w_0, w_1, \dots)$, $\sigma \models P$ si et seulement si w_0 est un monde dans lequel P est vérifiée.

Définition 12. Un SMAS S vérifie une formule temporelle ϕ ($S \models \phi$) si, et seulement si, toutes ses traces possibles vérifient ϕ .

Nous écrivons $S \vdash \phi$ pour signifier que $S \models \phi$ est démontrable.

Afin de pouvoir prouver les propriétés de vivacité, nous nous appuyons sur les règles de Hoang et Abrial. Cependant, nous devons les adapter à notre modèle. Premièrement, nous avons besoin de la propriété suivante pour prouver que notre méthode est correcte.

Propriété 1. Une exécution d'un GDT consiste toujours en un nombre fini d'actions

Démonstration. (Esquisse). Tous les opérateurs du modèle, même ceux que nous n'avons pas introduits dans cet article, ne permettent de revenir sur un même nœud qu'un nombre de fois fini. Pour plus d'informations, nous renvoyons le lecteur à [19]. \square

Ensuite, nous devons définir (pour pouvoir le prouver) le fait qu'un système mène d'un P_1 -monde à un P_2 -monde, avec $P_1, P_2 \in \mathcal{L}$.

Définition 13 (Mener à). Un SMAS S mène de P_1 à P_2 si pour toute trace possible, $S \models \square(P_1 \rightarrow \bigcirc P_2)$, c'est-à-dire si le SMAS est dans un P_1 -monde, chaque action possible mène à un P_2 -monde. Cette propriété se vérifie en prouvant que pour chaque action a dans S , pour chaque $outcome_a \in OUTCOME_a$:

$$pre_a \wedge C_a \wedge outcome'_a \wedge P_1 \Rightarrow P_2'$$

Ensuite, nous devons définir (toujours pour pouvoir le prouver) le fait qu'un système converge en P , ce qui signifie que chaque trace infinie se termine par une quantité infinie de P -monde.

Définition 14 (Convergence en P). Un SMAS S est convergent en $P \in \mathcal{L}$ si toute trace infinie de celui-ci se termine par une suite infinie de P -mondes. Cette propriété peut se vérifier en suivant une méthode de variant, c'est-à-dire :

- Soit une expression entière V , le variant.
- Lorsque le SMAS est dans un P -monde, prouver que pour chaque action a :

$$pre_a \wedge C_a \wedge P \Rightarrow V \in \mathbb{N} \quad (6)$$

et pour chaque $outcome_a \in OUTCOME_a$:

$$pre_a \wedge C_a \wedge outcome'_a \wedge P \Rightarrow V' \leq V \quad (7)$$

- Prouver qu'il existe au moins une action accessible dans chaque GDT pour laquelle, dans un monde $\neg P$:

$$pre_a \wedge C_a \wedge (post'_a \vee gpf'_a) \wedge \neg P \Rightarrow V' < V \quad (8)$$

Proposition 1. La méthode de la définition 14 prouve la convergence d'un SMAS S vers une propriété $P \in \mathcal{L}$.

Démonstration. Supposons qu'une trace se termine par une suite infinie de $\neg P$ -mondes. V a donc une probabilité 1 d'être diminué infiniment souvent. En effet, tout d'abord il y a une action accessible par GDT qui fait diminuer le variant (voir condition 8 ci-dessus). Comme l'exécution de chaque GDT est finie, d'après la propriété 1, et que la trace est infinie par hypothèse, il y a une infinité d'activation de GDT. Comme pour chaque exécution de GDT, il y a une probabilité non nulle de diminuer le variant et qu'aucune action n'augmente le variant (voir condition 7 ci-dessus), il est infiniment diminué de manière presque sûre. Cependant, puisque V est un entier dans les $\neg P$ -mondes, cela constitue une contradiction. Par conséquent, il ne peut y avoir de trace qui se termine par une suite infinie de $\neg P$ -mondes. \square

Par la suite, notons $S \vdash\downarrow P$ le fait que « S est convergent en P » est prouvable.

Définition 15 (Sans blocage). Un SMAS S est sans blocage en P si toute trace finie de S ne se termine pas en un P -monde.

Pour prouver une telle propriété, nous écrivons que si tous les contextes C_N de tous les nœuds feuilles sont faux, alors le SMAS est bloqué. Cependant, l'inverse n'est pas vrai car le système peut s'arrêter pour d'autres raisons. Une condition suffisante de non-blocage en P est donc :

Propriété 2. Soit S un SMAS, avec k types d'agents, alors si $P \Rightarrow \bigvee_i(tc_i)$, avec tc_i étant les contextes de déclenchement alors S est sans blocage en P .

Démonstration. Si un seul contexte de déclenchement est vrai, il suffit de s'assurer que le SMAS n'est pas bloqué, donc si P implique qu'au moins un contexte de déclenchement est vrai, cela signifie qu'il ne peut y avoir de trace finie qui se termine par un P -monde. \square

Remarque 4. C'est une condition suffisante mais elle n'est pas nécessaire puisque tous les contextes de déclenchement peuvent être faux à un instant donné alors que les agents ont encore des actions à exécuter.

Par la suite, notons par $S \vdash\bigcirc P$ le fait que « S est sans blocage dans n'importe quel P -monde est prouvable ».

Règles pour les propriétés de vivacité. Nous adaptons désormais les méthodes proposées par Hoang et Abrial aux GDT4MAS probabilistes. Nous pouvons maintenant appliquer les règles pour prouver les propriétés de vivacité comme la *persistance*, c'est-à-dire que quelque chose doit finalement rester vrai pour toujours. Nous faisons pour cela référence à la règle développée par Hoang et Abrial donnée en table 4.

TABLE 4 – Règle pour la propriété d'apparition

$$\frac{S \vdash \downarrow P \quad S \vdash \circlearrowleft \neg P}{S \vdash \diamond \square P}$$

Cette règle signifie que si un système converge en P et est sans blocage en $\neg P$ alors il en découle que P finira par rester vrai indéfiniment. Nous considérons principalement cette règle dans la suite, mais le lecteur peut se référer à [9] pour les autres règles. Si l'utilisateur spécifie le variant et les actions qui le font décroître, ces preuves sont automatisables de la même manière que les preuves classiques. En effet, une fois le variant et les actions qui le font décroître connues, un logiciel comme PVS peut prouver les points présentés en définition 14 et en propriété 2.

5.2 Le cas de la terminaison

La terminaison, qui consiste à prouver que le système finira par se terminer, est toujours une question importante en preuve formelle. Nous proposons ici une première approche pour prouver la terminaison d'un SMAS. Remarquons que les spécifications stochastiques données en section 4 n'interviennent pas dans cette approche, mais elles seront nécessaires pour prouver davantage de cas de terminaison dans le futur.

Nous caractérisons qu'un SMAS s'est terminé lorsque tous les contextes de déclenchement sont faux et qu'aucun agent n'a d'action en attente, c'est-à-dire que toutes les exécutions de GDT sont terminées et qu'aucune ne peut commencer. Bien que notre modèle ne permette pas d'écrire formellement la fin d'exécution d'un GDT dans le cas général, nous pouvons néanmoins exprimer cette propriété en nous appuyant uniquement sur les contextes de déclenchement.

Proposition 2. *Soit S un SMAS qui possède k types d'agents, c'est-à-dire qu'il y a k GDT différents. Notons tc_1, \dots, tc_k le contexte de déclenchement des différents GDT. S se termine finalement si, et seulement si, $S \models \diamond \square (\neg tc_1 \wedge \dots \wedge \neg tc_k)$.*

Ainsi, un SMAS S se termine finalement si S vérifie que tous les contextes de déclenchement finiront par rester faux.

Démonstration. (\Rightarrow) Supposons que le SMAS se termine dans le futur alors il existe un monde ω dans lequel les contextes de déclenchement sont faux. (\Leftarrow) Supposons que $S \models \diamond \square (\neg tc_1 \wedge \dots \wedge \neg tc_k)$. L'exécution d'un GDT est finie selon la propriété 1. Puisqu'il existe un monde à partir duquel les contextes de déclenchement restent faux, il existe bien un monde futur dans lequel toutes les exécutions sont terminées. \square

Prouver $S \models \diamond \square (\neg tc_1 \wedge \dots \wedge \neg tc_k)$ nécessite d'écrire un variant unique sur tous les GDT, ce qui n'est pas très pratique. Utilisons alors le fait que $\diamond \square \neg tc_1 \wedge \dots \wedge \diamond \square \neg tc_k$ est équivalent à $\diamond \square (\neg tc_1 \wedge \dots \wedge \neg tc_k)$, puis prouvons chacun d'eux séparément. Selon la règle de la table 4, pour prouver $\diamond \square \neg tc$ il faut d'abord prouver $S \vdash \downarrow \neg tc$ en définissant un variant. Ensuite, il faut prouver $S \vdash \circlearrowleft tc$, et cela est toujours trivial puisqu'il s'agit de la définition d'un SMAS sans blocage. En effet, si un contexte de déclenchement est vrai, le SMAS ne peut pas être bloqué. Il ne reste que la première propriété à prouver.

Remarque 5. Si nous montrons qu'un GDT finit par se terminer, alors il est certain qu'il existe un monde dans lequel le GDT est définitivement terminé. Ainsi, nous pouvons ne pas considérer les actions du GDT pour le reste de la preuve, et donc la simplifier.

Remarque 6. Si un variant ne contient que des variables internes d'un agent, seules ses actions peuvent être prises en compte pour la preuve de convergence. En effet, nous pouvons supposer que cet agent s'exécute autant de fois que nécessaire car, s'il ne le fait pas, c'est qu'il est terminé et aucun autre agent ne peut modifier le variant.

Reprenons l'exemple du SMAS producteur/consommateur donné en section 4.

Preuve de terminaison. Nous voulons prouver que le SMAS finit par se terminer. Pour ce faire, nous devons prouver que $S \models \diamond \square \neg tc_{pro}$ et que $S \models \diamond \square \neg tc_{cons}$.

Concentrons-nous sur le *Producteur*. Nous voulons prouver $S \models \diamond \square \neg (S_{pro} \leq -1)$. Pour ce faire, nous devons prouver :

$$S \models \downarrow (S_{pro} \leq -1) \quad (9)$$

$$S \models \circlearrowleft (S_{pro} > -1) \quad (10)$$

Comme indiqué en section 5.2, (10) est déjà prouvée. Pour la preuve que S est convergent en $(\neg tc_{pro})$, considérons le variant $V = S_{pro}$. Nous devons prouver qu’aucune action du *Producteur* n’augmente le variant, ce qui est vrai par définition. Nous devons alors montrer qu’au moins une action du *Producteur* diminue le variant, ce qui est vrai pour l’action a_2 .

Nous avons prouvé que $S \vdash \diamond \Box \neg tc_{pro}$. Prouvons maintenant $S \vdash \diamond \Box \neg tc_{cons}$. Pour ce faire, nous répétons exactement le même processus, en nous appuyant sur le fait que nous avons déjà prouvé que le *Producteur* se termine. Dès lors, nous pouvons nous placer dans un monde où celui-ci est déjà terminé. Par contrainte de place, nous ne présentons pas ici les détails mais il s’agit de la même structure de preuve que précédemment où S_E remplace S_{pro} . Nous devons alors montrer qu’au moins une action diminue le variant, ce qui est vrai pour a_3 .

Nous avons maintenant prouvé que $S \models (\diamond \Box \neg tc_{c1}) \wedge (\diamond \Box \neg tc_{c2})$ ce qui implique $S \models \diamond \Box \neg tc_{c1} \wedge \neg tc_{c2}$. Par conséquent, le SMAS finit par se terminer.

6 Conclusion et perspectives

Nous avons étendu le modèle GDT4MAS avec des spécifications probabilistes. Pour maintenir la correction du modèle, nous en avons adapté les schémas de preuve et introduit de nouveaux invariants pour chaque agent. De plus, nous avons adapté une méthode de variant afin de pouvoir prouver certaines propriétés de vivacité. Les travaux futurs consisteront à utiliser ce modèle afin de prouver formellement des propriétés telles que « la probabilité d’une formule donnée est égale à une quantité donnée », comme cela se fait en *model checking*. De plus, de manière plus générale, nous voudrions considérer des systèmes ouverts, avec apparition ou disparition d’agents, des dépendances aux conditions initiales ainsi que la preuve de propriétés émergentes.

Références

- [1] J-R. Abrial, A. Hoare and P. Chapron, *The B-Book*, Cambridge University Press, 1996.
- [2] M. Aouadhi, A fully probabilistic extension of Event-B, Technical Report, 2016.
- [3] M. Aouadhi, *Introduction de raisonnement probabiliste dans la méthode B événementiel*, PhD. thesis, 2017.
- [4] C. Baier and M. Kwiatkowska, Automatic verification of liveness properties of randomized systems, *PODC*, pp. 295, 1997.
- [5] M. Esteva, D. Cruz, and C. Sierra, ISLANDER : An electronic institutions editor, *AAMAS*, pp. 1045–1052, 2002.
- [6] V. Forejt, M. Kwiatkowska, G. Norman, and D. Parker, Automated verification techniques for probabilistic systems, *SFM*, pp. 53–113, 2011.
- [7] Z. Graja, F. Migeon, M.-P. Gleize et A. H. Kacem , Vers une modélisation formelle basée sur le raffinement des systèmes multi-agents auto-organisateurs, *JFSMA*, 2014.
- [8] T.S. Hoang, *The development of a probabilistic B-method and a supporting toolkit*, PhD. thesis, 2006.
- [9] T.S. Hoang and J-R. Abrial, Reasoning about liveness properties in Event-B, *ICFEM*, pp. 456–471, 2011.
- [10] S. Hudon, T.S Hoang and J.S. Ostroff, The Unit-B Method, Refinement guided by Progress Concerns, *SoSyM*, pp.1091-1116,2016
- [11] J.G. Kemeny, J.L. Snell, and A.W. Knapp, *Denumerable Markov chains*, Springer, 1969.
- [12] M. Kwiatkowska, G.Norman, and D.Parker, PRISM 4.0, Verification of probabilistic real-time systems, *CAV*, vol. 6806, pp. 585–591, 2011.
- [13] M. Kwiatkowska, G. Norman, D. Parker, and G. Santos, Equilibria-based probabilistic model checking for concurrent stochastic games, *FM*, pp. 298–315, 2019.
- [14] M. Kwiatkowska, G. Norman, D. Parker, and G. Santos, Automatic verification of concurrent stochastic systems, *Formal Methods in System Design*, vol. 58, pp. 188-250, 2021.
- [15] A. Lomuscio and E. Pirovano, Parameterised verification of strategic properties in probabilistic multi-agent systems, *AAMAS*, pp. 762–770, 2020.
- [16] A. Lomuscio, H. Qu, and F. Raimondi, MCMAS : An opensource model checker for the verification of multi-agent systems, *STTT*, vol. 19, pp. 9-30, 2017.
- [17] A. Lomuscio, and F. Raimondi, MCMAS : A model checker for the verification of multi-agent systems, *TACAS*, pp. 450-454, 2006.
- [18] A. McIver and C. Morgan, *Abstraction, refinement and proof for probabilistic systems*, Springer, 2005.
- [19] B. Mermet and G. Simon, GDT4MAS : An extension of the GDT model to specify and to verify multiagent systems, *AAMAS*, pp. 505–512, 2000.
- [20] B. Mermet and G. Simon, A new proof system to verify GDT agents, *IDC*, pp. 181–187, 2013.
- [21] S. Owre, J.M. Rushby, and N. Shankar. PVS : A prototype verification system, *CADE* vol. 607, pp. 748-752, 1992.
- [22] E. Pirovano. *Parameterised model-checking of probabilistic multi-agent systems*, PhD. thesis, 2021
- [23] K. Stathis, A.C. Kakas, W. Lu, N. Demetriou, U. Endriss, and A. Bracciali, PROSOCS : A platform for programming software agents in computational logic, *AT2AI*, pp. 523–528, 2004.

Jeunes-chercheurs

Architecture distribuée de collaboration pour systèmes critiques : application aux systèmes navals

Paul Quentel
paul.quentel@imt-atlantique.fr

IMT Atlantique, Lab-STICC, F-29238 Brest, France
Thales Défense Mission Systems, Brest, France

1 Introduction et problématiques

L'évolution du contexte de défense aéronaval nécessite une modification majeure des architectures des systèmes de senseurs. Dans le cadre du combat collaboratif naval, nous souhaitons concevoir une architecture qui associe des senseurs hétérogènes et distribués sur différentes plateformes pour effectuer des actions collaboratives. Par exemple, deux senseurs contribueraient ensemble à l'obtention de mesures plus précises. Des travaux récents montrent que les agents intelligents sont la métaphore naturelle pour traiter de la résolution distribuée de problèmes complexes. Ainsi, l'approche multi-agent rendrait possible l'allocation dynamique de tâches à des senseurs en réseau dans un contexte de collaboration multi-plateforme et multi-senseur.

2 Contraintes et besoins

Notre système devra trouver une solution à la problématique de l'allocation de tâche sous certaines contraintes et avec des besoins définis :

- **Mise à l'échelle** : Nombre important de plateformes et de senseurs ;
- **Contraintes temporelles** : Rapidité de l'allocation de tâches ;
- **Robustesse et résilience** : Adaptation du système en cas de perte d'une plateforme ou d'un senseur en cours de mission ;
- **Communication** : Besoin en bande passante des agents pour les échanges entre les plateformes ;
- **Modularité et adaptabilité** : Faciliter l'intégration de nouveaux services avec une diversité plateformes ;
- **Efficacité** : Maximiser le nombre de tâches accomplies, prioriser les tâches importantes.

3 Concepts et expérimentations

Nous avons conçu trois types d'agents dont les échanges sont présentés en figure 1. Ces agents

possèdent tous une structure générique qui comprend une mémoire interne, des fonctions et un système de communication.

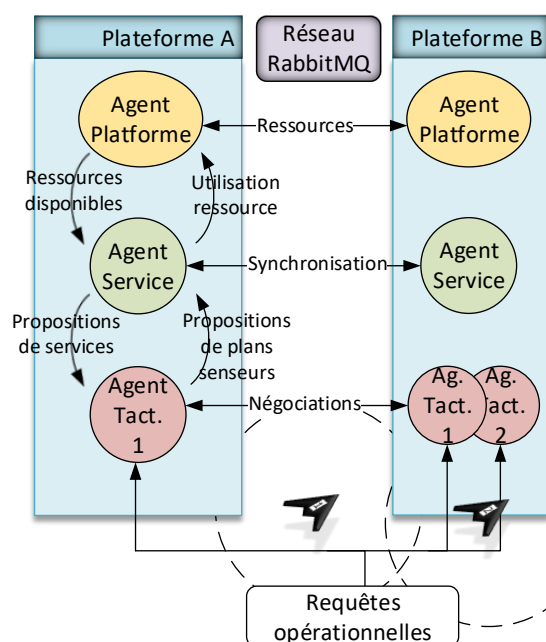


FIGURE 1 – Echanges entre agents

Au niveau des expérimentations, nous développons un banc d'essai pour l'évaluation de notre architecture multi-agent selon les critères vus en section 2. Dans ce but, nous souhaitons observer certaines métriques dans diverses configurations du système. Nous évaluerons le besoin en bande passante de l'approche et les performances de l'allocation de tâche en termes de rapidité et d'utilité pour la mission en fonction du nombre de plateformes présentes dans le réseau, du nombre de menaces et du nombre de senseurs actifs. De plus, le système multi-agent devrait permettre au système d'adapter l'utilisation des senseurs dans des situations critiques où des plateformes peuvent sortir du réseau en cours de mission. Ces expérimentations, et les futurs résultats associés permettront d'observer les points forts de l'approche ainsi que ses limites.

De l'organisation d'un système multi-agent de cyberdéfense

Julien Soulé^{1,2}
julien.soule@lcis.grenoble-inp.fr

Jean-Paul Jamont¹, Michel Occello¹
{jean-paul.jamont, michel.occello}@lcis.grenoble-inp.fr

Paul Théron³
paul.theron@orange.fr

¹Univ. Grenoble Alpes, Grenoble INP, LCIS, Valence, France

²Thales Land and Air Systems, BL IAS, Rennes, France

³AICA IWG, La Guillermie, France

Résumé

Ce poster présente un travail de thèse s'intéressant aux systèmes de cyberdéfense vus comme des ensembles d'entités autonomes coopérantes et déployables au plus près des points sensibles d'un environnement hôte en réseau.

Mots-clés : cyberdéfense, système multi-agent, organisation

Abstract

This poster presents a PhD work focusing on cyber-defence systems modeled as a set of co-operating and deployable autonomous entities as close as possible to the critical points of a host networked environment.

Keywords : cyber-defence, multi-agent system, organization

1 Contexte et problématique

Le groupe de travail *AICA IWG* [1] développe des travaux sur les agents *AICA* (*Autonomous Intelligent Cyber-defence Agent*). L'agent *AICA* doit pouvoir être déployé sur un système hôte pour détecter, identifier et caractériser des anomalies/attaques, élaborer et piloter l'exécution de contre-mesures et dialoguer avec l'extérieur. L'agent *AICA* peut être vu comme un Système Multi-Agent (SMA) permettant ainsi de prendre en charge l'ouverture, le passage à l'échelle et l'autonomie de la cyberdéfense en déléguant ses différentes fonctions aux agents. Notre problématique consiste à définir l'organisation du SMA qui permettrait d'assurer la cyberdéfense compte tenu des contraintes de l'environnement hôte.

2 Aperçu des travaux

Nous avons conduit une revue des SMAs de cyberdéfense disponibles[3]. La majorité des sys-

tèmes étudiés ont une organisation centralisée et se concentrent sur la détection d'anomalies. Peu de travaux traitent d'organisation et d'objectifs de cyberdéfense différents (recherche de contre-mesures adaptées...) et ils sont peu établis dans la pratique. Cependant, en raison de la subjectivité de la classification et du manque de cohérence entre les systèmes, une modélisation commune des SMAs de cyberdéfense évoluant dans un environnement cyber permet leur comparaison dans un cadre d'évaluation de référence.

Une première proposition de modèle repose sur les *Decentralized Partially Observable Markov Decision Process (Dec-POMDP)* et permet de représenter l'environnement, des agents d'attaque et de défense ainsi que leurs interactions. Nous avons implémenté ce modèle au travers du simulateur *Multi Cyber Agent Simulator* [2] qui permet de modéliser, implémenter et évaluer plusieurs organisations de SMA de cyberdéfense.

3 Conclusion et perspectives

Outre l'amélioration du simulateur, l'utilisation plus approfondie de la décentralisation et de la coopération pourrait contribuer à concevoir des systèmes de cyberdéfense plus adaptatifs et opérationnels dans des contextes réalistes.

Références

- [1] AICA IWG. <https://www.aica-iwg.org/>.
- [2] Multi cyber agent simulator. <https://github.com/julien6/MCAS>. Accessed : 2023-03-07.
- [3] J. Soule et al. De l'organisation des systèmes multi-agents de cyber-defense. In *RESSI*, 2023.

Modélisation des déterminants de l'activité humaine au sein d'une simulation multi-agent

Jean-Baptiste Ly^{a,b}
jean-baptiste.ly@edf.fr

Quentin Reynaud^c
quentin.reynaud.pro@gmail.com

Chloé Le Bail^b
chloe.le-bail@universite-paris-saclay.fr

Mathieu Schumann^a
mathieu.schumann@edf.fr

Vincent Boccara^b
vincent.boccara@universite-paris-saclay.fr

Nicolas Sabouret^b
nicolas.sabouret@universite-paris-saclay.fr

^aEDF R&D, Palaiseau, France

^bLaboratoire Interdisciplinaire des Sciences du Numérique,
LISN, CNRS, UMR 9015, Université Paris-Saclay, France

^cQRCI, Clermont-Ferrand, France

SMACH (Simulation Multi-agent de l'Activité humaine et des Consommations dans l'Habitat) est une plateforme développée par EDF R&D simulant l'activité humaine au sein des foyers avec une approche centrée-agent qui mobilise des concepts de l'ergonomie [2, 3, 4]. Elle utilise des données issues de l'enquête emploi du temps de l'INSEE de 2009-2010 (Time Use Survey, TUS) pour générer l'activité humaine simulée. Cette dernière est couplée à divers systèmes énergétiques afin de générer des courbes de consommation d'énergie. Les agents sont réactifs face aux modifications de l'environnement, et adaptatifs face aux aléas de leur situation; ce qui permet l'émergence de comportements collectifs, et de faire des études de prospective.

Néanmoins, plusieurs limites sont soulevées. Un manque de données sur les variations de l'activité intra-individu dans les TUS provoque une forte limitation du réalisme des simulations au regard de la variabilité attendue des activités simulées au cours du temps. Pour viser un réalisme au niveau individuel, simuler des individus moyens n'a pas de sens [4]. Le manque de réalisme est aussi dû au manque de contexte dans les TUS qui n'indiquent pas ce qui détermine l'activité humaine.

Ainsi, notre objectif est d'élaborer un modèle multi-agent intégrant la notion de déterminant de l'activité humaine, afin de rendre les simulations plus réalistes par rapport aux limites précédemment mentionnées.

Nous proposons un modèle théorique global reposant sur certains concepts issus de l'ergonomie : l'activité est déterminée, finalisée, unique et variable [1]. Il intègre la notion de buts couplée à la fois aux données TUS, et aux déterminants de l'activité humaine.

Cette approche permettrait de simuler les impacts de ces déterminants sur les buts des agents, mais aussi sur les tâches permettant d'atteindre ces buts. Un exemple simple est donné pour le repas, et montre que ce couplage permettrait de générer des variations réalistes de la simulation de l'activité humaine.

En perspectives, nous nous penchons sur les différentes manières de mesurer les variations de l'activité en fonction de ses dimensions, puis nous proposons un recueil de nouvelles données en ergonomie permettant d'alimenter notre modèle théorique.

Références

- [1] François DANIELLOU et Pierre RABARDEL. "Activity-oriented approaches to ergonomics : some traditions and communities". en. In : *Theoretical Issues in Ergonomics Science* 6.5 (sept. 2005), p. 353-357.
- [2] YVON HARADJI. "Simulation multi-agent de l'activité humaine : une concrétisation en ergonomie du programme de recherche technologique « cours d'action »". fr. In : *activites* 18-1 (avr. 2021).
- [3] Quentin REYNAUD et al. "Using Time Use Surveys in Multi Agent based Simulations of Human Activity :". en. In : *Proceedings of the 9th International Conference on Agents and Artificial Intelligence*. Porto, Portugal : SCITEPRESS - Science et Technology Publications, 2017, p. 67-77.
- [4] Quentin REYNAUD et al. "Simulation de l'activité humaine. Une étude sur le réalisme multi-niveau". en. In : *Revue d'intelligence artificielle* 32.2 (avr. 2018), p. 197-221.

Modélisation de comportements humains

Comprendre l'impact des stratégies de prix sur le comportement des consommateurs

J. Vanderlynden^{a,b} P. Mathieu^a R. Warlop^b
jarod.vanderlynden@fifty-five.com philippe.mathieu@univ-lille.fr romain@fifty-five.com

^aLaboratoire CRISAL de Lille,
Université des Sciences et Technologies de Lille, France

^bEntreprise fifty-five,
Paris, France

Résumé

Le marketing est un outil indispensable pour les entreprises qui souhaitent faire connaître un nouveau produit ou fidéliser des consommateurs. Néanmoins, cet outil est influencé par de nombreux facteurs qui rendent sa compréhension complexe et difficilement mesurable dans sa globalité. Nous proposons dans cet article un modèle centré individus qui nous permet de mesurer par exemple les effets des guerres des prix sur la fidélité des consommateurs, l'augmentation des ventes du produit en promotion, la baisse des ventes des concurrents ou les effets sur les consommateurs de la répétition des promotions. Ce modèle multi-agents, à la fois simple et puissant, offre la possibilité de quantifier ces phénomènes et constitue une avancée dans la compréhension et la calibration de nouvelles stratégies marketing.

Mots-clés : Simulation, Comportements, Multi-Agents, Marketing, Dynamique des prix

Abstract

Marketing is a complex tool used by companies to publicize new products or build consumers loyalty. Nevertheless, this tool is influenced by many factors that make its understanding complex and difficult to measure in its entirety. We propose in this article a new agent-based model which allows reproducing the effects of price wars on consumer loyalty, or measure the increase in sales during discounts, the drop in competitors' sales or discount repetition negative effects. This model, both simple and powerful, allows to measure these phenomena and is a step forward in understanding and calibrating new marketing strategies.

Keywords: Simulation, Behaviors, Agent-based, Marketing, Pricing strategy

1 Introduction

Depuis de nombreuses années, les entreprises commerciales mesurent l'importance et les apports d'une campagne promotionnelle, qu'elle soit de type réduction de prix ou de type publicitaire. Leur souhait est évidemment que cette campagne soit la plus efficace possible, et cela, en cernant au mieux les moyens à utiliser et la cible concernée. Malheureusement, ces campagnes ont en général un coût très élevé et peuvent, si elles ne sont pas exécutées correctement, avoir peu d'utilité, voire même avoir une influence négative sur l'image d'une marque ou d'un produit. Comme pour de nombreux phénomènes coûteux et complexes à mettre en œuvre *in vivo* il est préférable d'essayer de mesurer à priori, *in silico* l'impact des campagnes que l'on souhaite réaliser. Par ailleurs, le test *in silico* est depuis quelques années communément admis [15, 1, 16, 9, 13]. Enfin, les effets des campagnes marketing (mix marketing [6]) sont principalement modélisés dans l'industrie par des méthodes statistiques [18, 19] et d'apprentissage automatique [18, 12], bien souvent pour faire de la segmentation clientèle et de l'aide à la décision. Ces approches globalisantes ne permettent ni la mise en place de comportements fins des consommateurs ni la modélisation d'une multitude de différentes campagnes. Par exemple : connaître la différence d'impact entre une promotion de 10% et de 20% ou savoir sur quels consommateurs la différence est forte ou faible. En effets, les approches globalisantes ne permettent de répondre que partiellement à ces questions. Nous soutenons que les systèmes multi-agents (SMA) avec leur approche centrée individus facilitent la conception et la calibration de comportements à un niveau de finesse permettant de mieux comprendre la multitude de facteurs auxquels est confrontée une campagne marketing. Nous montrons par ailleurs qu'une telle approche permet une adaptation plus aisée aux

changements de l'environnement comme l'arrivée ou le changement d'un produit et fournit, de ce fait, des résultats robustes et exploitables.

Dans cet article, nous nous plaçons dans le cadre d'une grande surface de type supermarché avec pour objectif la compréhension des dynamiques de consommateurs à travers leurs réactions, au fil du temps, face aux changements de prix ou de packaging des produits. Pour ce faire, nous proposons une modélisation centrée individus, permettant à travers une simulation de déclencher des campagnes promotionnelle à une date et une durée choisie, et permettant d'en mesurer l'impact sur des populations variées. La grande diversité des comportements clients (inertiels, dépendants du prix ou de la qualité, promophiles, équilibrés...) et des campagnes promotionnelles (baisse du prix, réduction en pourcentage, bons d'achats, lots achetés/offerts...) motive l'utilisation de ce type de modèle.

Dans notre modèle les interactions sont basées sur des critères issus de différentes disciplines comme les sciences sociales pour l'aversion à la perte (Prospect theory) [10], le marketing pour l'inertie influant le choix des marques [3, 7], ou les sciences économiques pour l'évaluation de la qualité [14]. Dans cet article nous proposons un modèle capable de prendre en compte ces différents aspects en nous inspirant par ailleurs de [16, 9, 15] pour la mise en place du choix des produits. Cependant, contrairement à ces articles qui se focalisent sur le comportement des consommateurs, la dynamique des prix est ici le sujet d'étude principal.

Nous montrons dans cet article que notre approche centrée individus, focalisée sur les prix des promotions, permet une meilleure compréhension des comportements que l'on modélise et engendre les faits stylisés connus du domaine comme l'augmentation du volume des ventes et la baisse des ventes concurrentes lors d'une réduction de prix. Ces deux phénomènes sont impératifs à exhiber puisqu'ils sont communs à toute campagne de réduction de prix. Notre approche les met en évidence, mais elle permet de plus d'observer l'impact des réductions de prix sur les phénomènes d'acquisition et de fidélisation selon les profils de consommateurs, ou encore les effets des guerres de prix sur les différents comportements. Finalement, notre modèle multi-agent est auto-adaptatif aux changements de l'environnement, notamment à l'apparition de nouveaux produits, aux changements de prix, ou à la présence d'une nouvelle campagne marketing et ses différents paramètres. Il permet

donc de mesurer efficacement les conséquences de différents scénarios "what if" de test d'hypothèses sur la campagne à réaliser. Sans avoir à réécrire les différents comportements il est donc possible de changer les différents paramètres et d'en calculer les conséquences.

Dans la première partie, nous présentons l'état de l'art du domaine à travers des travaux réalisés dans un cadre similaire au nôtre et nous critiquons le manque de lien entre l'évolution des prix et comportement des consommateurs. Dans la seconde partie, nous présentons un modèle multi-agents de test de campagnes de réductions qui s'appuie sur une fonction d'utilité individuelle que possède chaque agent pour évaluer les produits. La troisième partie décrit plusieurs expériences réalisées avec ce modèle ainsi que les résultats qu'il permet d'obtenir, comme la fidélisation ou les volumes de vente. Enfin, la dernière partie aborde les avantages et les extensions potentielles de ce modèle, notamment la prise en compte de l'influence sociale.

2 État de l'art

2.1 Les SMAs en marketing.

L'étude des stratégies marketing par la simulation n'est pas nouvelle. D'autres travaux s'y sont déjà essayés, souvent à l'aide d'approches équationnelles ou statistiques [2]. Peu d'entre eux se sont appuyés sur une approche centrée individus pourtant nécessaires pour assurer de la différenciation comportementale. On peut néanmoins citer [15, 1, 16, 9, 13] qui utilisent tous une approche à base d'agents.

Les auteurs de [9] modélisent le lancement d'un produit dans une population soumis au phénomène de bouche-à-oreille (WOM pour "Word Of Mouth"). Les possibilités d'interactions sont modélisées via un graphe de type "small world" (Watts–Strogatz) et les interactions sont simples, si un agent adopte le nouveau produit, il tente de convaincre ses voisins de faire de même. Le bouche-à-oreille est un phénomène complexe et les modèles qui l'étudient intègrent des graphes afin de représenter des notions de contacts sociaux, contacts que les agents peuvent influencer. Nous supposons que ces modèles de bouche-à-oreille, et plus globalement d'influence sociale, sont un possible développement futur, mais s'intègrent mal avec ce que nous essayons de montrer. En effet, la superposition des effets de dynamiques des prix et de bouche-à-oreille réduirait l'interprétabilité des résultats, de plus nous

nous intéressons avant tout à comprendre l'impact des prix sur les différents comportements. Les auteurs de [15] proposent des lignes directrices afin de modéliser correctement des aspects marketing dans des SMA. Par ailleurs, les auteurs proposent une approche basée sur l'évaluation des produits suivant une fonction d'utilité. Cela permet de modéliser des agents qui, selon leurs caractéristiques personnelles, évalueront différemment les articles. Ainsi, il devient possible de créer et moduler les caractéristiques des agents, de manière à reproduire des comportements classiquement observés en marketing. Les auteurs de [16] présentent un SMA utilisant des paramètres comportementaux servants à réguler la diffusion ou le reçu de stimuli externe à eux même, ces paramètres sont similaires à ce que nous proposons. Ils montrent que leur modèle est capable de reproduire des faits stylisés sur le choix des marques des consommateurs, notamment l'émergence d'un équilibre entre les parts de marchés et un effet de verrouillage des parts de marché d'une marque dominante ou encore une compétition cyclique entre les marques dominantes. Argumentant que leur outil de simulation permet de reproduire une évolution réaliste d'un marché. L'auteur de [13] applique les quatre éléments marketing ([6]), produit, prix, place de distribution et promotion, dans un modèle centré individus comportemental tiré des simulations sociales. Les agents ont des préférences individuelles et des préférences sociales (qui consomme les produits). Les préférences individuelles sont définies par les caractéristiques de chaque agent et les préférences sociales sont déterminées en regardant la consommation des individus connectés socialement dans un graphe.

On notera cependant que les modèles présentés dans [9, 16] n'intègrent pas explicitement la composante prix, et que [15, 1, 13] intègrent bien le prix, mais s'intéressent avant tout à l'influence sociale sans étudier le lien prix - comportement, point central de notre travail.

D'une manière générale, certaines propriétés sont plus faciles à modéliser à l'aide de SMA et de modèles centrés individus qu'avec des modèles équationnels. C'est le cas notamment de l'influence sociale qui nécessite d'explicitement les liens entre les différents individus, par opposition à la notion de publicité qui peut plus facilement être étudiée avec un modèle équationnel. Cela se reflète dans la littérature.

2.2 Qu'est-ce qui fait un comportement ?

La modélisation d'un processus de comportement d'achat s'appuie sur deux aspects fondamentaux : les influences internes (caractéristiques propres à chaque individu qui influence l'envie d'acheter tel ou tel produit) et les influences externes (la publicité, la promotion, le bouche à oreilles). Dans ce travail on se focalise uniquement sur les influences internes et la promotion.

Il semble naturel pour tous les auteurs d'utiliser le prix et la qualité de chaque produit comme influence interne. [10, 3, 17, 8] proposent d'y ajouter deux critères supplémentaires permettant de rendre les modèles plus réalistes : l'aversion à la perte et l'inertie.

- l'aversion à la perte (prospect theory), s'appuie sur l'idée que perdre 1€ a plus d'impact sur un consommateur que de gagner 1€.
- l'inertie ou la loyauté envers les marques s'appuie sur l'idée qu'un consommateur ne prendra pas forcément le "meilleur" produit disponible, car il est aussi influencé par ses habitudes et sa loyauté envers les marques.

[10, 8] suggèrent d'utiliser un produit de référence pour prendre en compte l'aversion à la perte. Celui-ci peut être propre à chacun grâce à l'utilisation de SMA. L'inertie peut être simplement prise en compte par un processus de renforcement (plus j'en ai acheté, plus j'en achète) [3, 17]. Ces différentes notions peuvent être combinées à l'aide d'une fonction d'utilité qui sera utilisée lors de l'évaluation d'un produit ([15]).

2.3 L'impact des promotions

Les promotions ont des effets indiscutables sur les ventes. [5] décrit les effets connus de ces promotions telles que l'augmentation du volume des ventes, l'impact asymétrique de ces promotions sur les autres produits en fonction de leur qualité (la qualité permet de limiter la perte de vente), ou encore l'impact dégressif d'une promotion au fil des répétitions. L'apparition de ces différents effets nous semblent indispensables pour l'évaluation d'un modèle de campagnes promotionnelles. En conclusion, notre modèle utilise une approche centrée individus comme [15, 1, 16, 9, 13], implémente l'aversion à la perte et l'inertie comme [10, 3, 17, 8] et il permet d'exhiber les propriétés classiques d'impact des promotions décrites par [5].

3 Description du modèle

Dans cet article nous nous focalisons sur un modèle de magasin dans lequel il est possible de simuler différentes promotions sur différents packs de produits. Par exemple une promotion en pourcentage de prix ou une promotion de type *y offert pour l'achat de x*. Le modèle est aussi capable de réagir à des modifications de prix hors du cadre d'une promotion temporaire, ou encore à l'arrivée de nouveaux packs dans le magasin. Dans ce modèle, il n'y a pas de représentation spatiale, les agents sont ici omniscients et connaissent tous les packs et leurs caractéristiques. Nous ne prenons pas en compte le positionnement géographique du magasin et des packs ni l'influence sociale pour se concentrer sur l'influence du prix et des promotions. La notion de pack est à prendre au sens large, un pack est un lot de produit, il dispose donc d'une quantité d'un produit. Pour chaque produit, il existe une quantité unitaire (en gramme, en ml, etc.) pour des raisons de simplicité, nous ne modélisons pas cette quantité unitaire, par contre, nous nous intéressons aux promotions des lots de produits que nous appellerons pack.

Nous présentons dans un premier temps les packs et les agents qui constituent les clients du magasin, dans un second temps l'environnement qui caractérise le magasin et ses packs, dans une troisième partie la dynamique du modèle et finalement, nous présentons le modèle comportemental, en d'autres termes, la manière dont les agents raisonnent pour choisir les packs.

3.1 Les paramètres des packs

Un pack représente un article quelconque dans un supermarché. Cet article peut être vendu en quantité unique, il correspond donc à un article seul, ou en lot. Cette information est représentée par la caractéristique quantité. Soit p le prix, Q_{te} la quantité, Q_a la qualité et D un booléen indiquant la promotion. Un pack est représenté par un quadruplet noté $P(p, Q_{te}, Q_a, D)$. Chaque pack appartient à une catégorie $C_i \in C = C_1, C_2, \dots$

$$\forall P_j \exists C_i | P_j \in C_i, C_i \in C \quad (1)$$

3.2 Les paramètres des clients/agents

Un agent, noté $a(H_i, \lambda_i, P_{ref_i}, (\beta_p, \beta_q, \beta_i, \beta_l))$, représente une entité (une personne, une famille ou autre) qui vient faire ses courses régulièrement dans le magasin. Son comportement doit

être basé sur ses habitudes. Un agent est caractérisé par des paramètres internes qui le différencient de ses semblables ainsi que d'un historique individuel et de références mentales pour chacune des catégories. Soit :

- $H_{i,a}$ une liste de produits pour chaque catégorie correspondant à l'initialisation de l'historique d'achat de l'agent.
- $P_{i,a,t}$ Le produit acheté par l'agent a dans la catégorie C_i au temps t .
- l_h la longueur des historiques d'achats.
- $\lambda_{i,a}$ le besoin pour chaque catégorie C_i .
- $P_{ref,i} \in C_i$ un pack de référence, toujours unitaire $Q_{te} = 1$, à l'instar de [10]
- $\beta_p, \beta_q, \beta_i, \beta_l$ respectivement la sensibilité au prix, la qualité, l'inertie et aux promotions.

3.3 L'environnement

L'environnement représente le magasin. Il contient les agents et les packs. Les agents interagissent avec l'environnement en achetant des packs. On souhaite, avec les paramètres de l'environnement, pouvoir moduler le fonctionnement global du modèle. Par exemple donner plus d'importance à la promotion ou augmenter la puissance de l'aversion à la perte. L'environnement est donc caractérisé par les propriétés suivantes :

- β paramètre de l'aversion à la perte (identique pour le prix et la qualité), $\beta > 1$.
- C limite de quantité d'achat, $C \geq 1$.
- α_{sat} paramètre de saturation.
- G_p, G_q, G_i, G_d les paramètres de régulation de l'impact respectivement du prix, de la qualité, de l'inertie et de la promotion (discount).

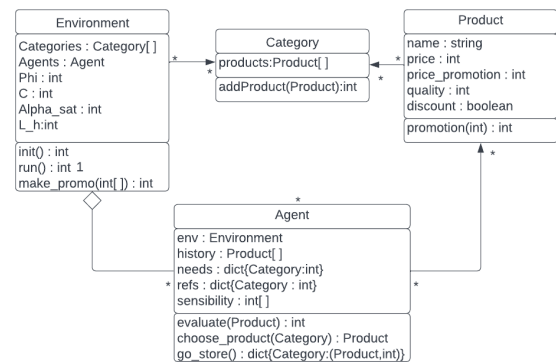


FIGURE 1 – Diagramme UML du modèle.

3.4 Hypothèses

Nous considérons les packs comme des biens de consommation courante, ce qui nous permet d'émettre l'hypothèse que les achats sont fréquents et donc qu'à chaque pas de temps chaque agent s'interroge sur l'achat. On pourrait considérer qu'un pas de temps représente une semaine, et que chaque client vient chaque semaine au magasin faire ses courses. On suppose que le besoin ($\lambda_{i,a}$) peut être calculé grâce à la moyenne de l'historique des quantités achetées, quel que soit le pack de la catégorie $\lambda_{i,a} = \text{mean}(H_{i,a}(qte))$. On exclut les achats de différents packs d'une même catégorie afin de simplifier le choix des agents. Cela revient à exclure l'achat de deux packs similaires, mais de marques différentes. L'objectif des différents agents sera donc de choisir, à chaque pas de temps, au plus un pack, dans chacune des catégories. Cette hypothèse n'interdit pas aux agents d'acheter plusieurs fois le même pack ni de ne rien acheter. On suppose que les agents sont des entités représentant un foyer, une famille... Cette hypothèse permet à un magasin de relier un agent à une carte de fidélité.

3.5 Le choix des packs

L'intérêt de l'agent sur une catégorie. Chaque agent, à chaque pas de temps, pour chaque catégorie, choisit s'il a besoin d'un pack de cette catégorie. L'agent a évalue C_i et calcule $B(C_i, t + 1)$ qui est la probabilité d'avoir besoin d'un pack de cette catégorie au temps $t + 1$. Enfin, soit $N(C_i, t, t - n)$ la quantité de pack acheté par l'agent lors des n dernières étapes.

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

$$B(C_i, t + 1) = \text{sigmoid}\left(\frac{\lambda_{i,a}}{N(C_i, t, t - n)}\right) \quad (3)$$

Intuitivement, cette formule permet à l'agent a d'augmenter la probabilité de s'intéresser à la catégorie C_i s'il a acheté peu de cette dernière sur les derniers pas de temps et vice et versa.

L'évaluation des packs. Lorsqu'un agent s'intéresse à une catégorie, il en évalue tous les packs, leur donne un score et en choisit un seul. Un score élevé signifie que le pack correspond aux attentes de l'agent et a plus de chances d'être choisi. Pour donner un score aux packs, l'agent

utilise ses sensibilités personnelles ainsi que les paramètres des packs.

Nous définissons quatre fonctions d'utilité U_1, U_2, U_3, U_4 , une pour chaque caractéristique évaluée, respectivement : prix, qualité, inertie et promotion. Les équations 4 et 5 comparent le pack en cours d'évaluation, au pack de référence sur le prix et la qualité. C'est dans ces formules que l'aversion à la perte est prise en compte. L'équation 6 prend en compte l'inertie, on la modélisera de la même manière que [3]. Finalement, la dernière fonction calcule l'impact d'une promotion. On pondère ensuite les résultats des quatre fonctions précédentes par les sensibilités de l'agent. La somme obtenue nous indique le score du pack pour cet agent. Plus le score est élevé, plus le pack intéresse l'agent. ¹

$$U_1 = G_p(\beta(p - p_{ref})_+ + (p_{ref} - p)_+) \quad (4)$$

$$U_2 = G_q(\beta(q - q_{ref})_+ + (q_{ref} - q)_+) \quad (5)$$

$$U_3 = G_i(10 \times nb_{bought} - nb_{bought}^2) \quad (6)$$

$$U_4 = G_d \times D \quad (7)$$

$$U = \sum_{k=1}^n U_k * \beta_{a,k} \quad (8)$$

On note que l'impact de la baisse du prix est calculée dans U_1 et non U_4 . On modélise avec U_4 uniquement l'impact de la présence ou non d'une promotion.

La quantité achetée. Le calcul de la quantité achetée est indépendant de la quantité interne à chaque pack. Ce calcul sert à connaître le nombre de packs achetés par l'agent. Si $P(p, Qte, Qa, D)$ est le pack choisi, et que Qte est 1, l'agent va alors grâce à la formule 9 calculer la quantité dont il a besoin, disons 3. L'agent achète alors 3 fois le même pack. $N(C_i, t)$ la quantité achetée au temps t de C_i .

$$\text{Achat}(P, t + 1) = \max(1, \lambda_{i,a} + N)S \quad (9)$$

$$N = \sum_{\tau=0}^T \frac{\lambda_{i,a} - N(C_i, t - \tau)}{T + 1} \quad (10)$$

$$S = \text{Sat}(U(P) - U(P_{ref}, t)) \quad (11)$$

$$\text{Sat}(x) = \frac{C}{1 + e^{-\frac{x}{\alpha_{sat}} + \log(C-1)}} \quad (12)$$

1. $(x)_+$ représente le maximum entre 0 et x .

Le calcul de la quantité est séparé en 2 parties : Une première liée au besoin $\lambda_{i,a}$ correspond à la partie de gauche de l'équation 9 (écart de $\lambda_{i,a}$ au T derniers pas de temps). Une seconde déterminée grâce au score du pack, définissant à quel point l'agent a trouvé un meilleur pack qu'à son habitude, traduit par la fonction de saturation. Celle-ci augmente ou réduit la quantité achetée en fonction du score du pack choisi P par rapport au pack de référence P_{ref} . Si le score de P est plus élevé l'agent achète en plus grande quantité sinon il en achète moins.

Pour résumer, la quantité achetée est fonction de l'écart du besoin $\lambda_{i,a}$ aux T derniers achats multiplié par la fonction de saturation.

Algorithm 1 Raisonnement

```

procedure RUNONESTEP
   $t \leftarrow$  time step  $t$ 
  for all agents  $a \in A$  do
     $a$  go to the store
    for all category  $c_i \in C$  do
       $H_{(i,a,t-l_h)} \leftarrow$  None (delete)
       $p \leftarrow B(c_i, t + 1)$ 
       $x \sim U(0, 1)$ 
      if  $x > p$  then
         $H_{(i,a,t)} \leftarrow (None, 0)$ 
      else
        for every pack  $p \in c_i$  do
           $score(p) \leftarrow U(p, a)$ 
           $l$  append  $(score(p), p)$ 
        end for
         $P_{i,a,t} \leftarrow$  Choose(1)
         $q \leftarrow Qt(P_{i,a,t}, ref)$ 
         $H_{(i,a,t)} \leftarrow (P_{i,a,t}, q)$ 
      end if
    end for
  end for
end procedure

```

3.6 La dynamique du modèle

À chaque pas de temps, tous les agents vont au magasin et déterminent pour chaque catégorie $C_i \in C$ si celle-ci les intéresse ou non suivant une probabilité $B(C_i, t + 1)$. Sur les catégories choisies à ce pas de temps par l'agent, celui-ci évalue tous les packs puis il choisit quel pack acheter pour cette catégorie et en quelle quantité. Ce choix suit le processus de décision interne à chaque agent décrit par l'algorithme 1.

La méthode Choose choisi un pack en utilisant une loi de probabilité proportionnelle au score

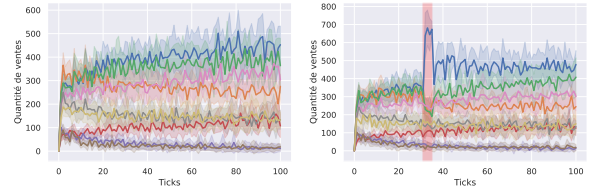


FIGURE 2 – Quantité des ventes pour chacun des 9 packs en fonction du temps dans une simulation. À gauche sans promotion, à droite avec une promotion de 40% entre les pas de temps 30 et 34.

du pack. La méthode Qt utilise les équations 9 à 12 pour calculer la quantité que l'agent a achète.

4 Expérimentations

Nos expériences² sont réalisées avec des paramètres d'environnement fixés pour permettre la comparaison. Les paramètres du modèle : G_p, G_q, G_i, G_d sont égaux à 1. Ce qui signifie que les effets du prix, de la qualité, de l'inertie et de la promotion ont un impact global similaire. Il reste les influences internes de chaque agent qui modifient ces effets comme indiqué dans l'équation 8. Enfin le paramètre β est fixé à 0.5, C à 3 et α_{sat} à 10. Sur une même expérience, les agents et packs ont les mêmes caractéristiques. Les procédures permettant de générer les agents et les packs sont des procédures stochastiques. Les agents sont catégorisés selon leurs sensibilités : c'est ce qui définit leur appartenance à un profil de consommateur. Toutes les expériences sont réalisées 20 fois pour plus de précision. Nous montrons que dans une même situation (agents et packs similaires), deux promotions identiques ont quasiment le même effet.

Le modèle est capable de reproduire des phénomènes promotionnels macroscopiques classiques en marketing comme :

- L'augmentation du volume des ventes sur un pack en promotion ([5])
- La cannibalisation : baisse des ventes des packs en concurrence avec un pack en promotion.
- La diminution de l'impact des promotions répétées sur un même pack.

Mais surtout, le modèle peut reproduire des phénomènes observables uniquement au niveau des agents, impossible à observer avec une approche qui ne serait pas centrée individu, comme :

2. Une feuille Jupyter : <https://github.com/cristal-smac/retail> permet de reproduire les expériences décrites ici.

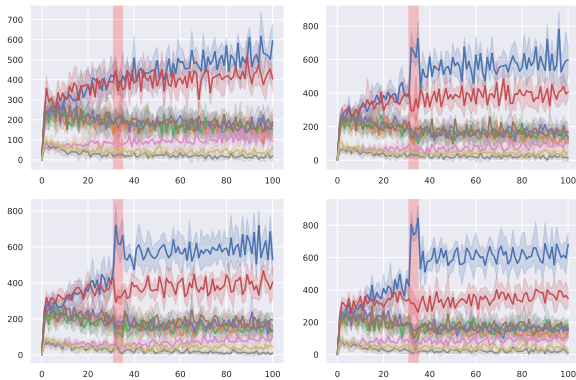


FIGURE 3 – Quantité des ventes pour chacun des 9 packs en fonction du temps dans une simulation. De gauche à droite, une promotion à 5%, 10%, 20% et 60% entre le pas de temps 30 et 34.

- De multiples promotions successives changent le prix de références des agents.
- La guerre des prix est un phénomène avec des impacts macroscopique, mais aussi microscopique en changeant les perceptions que les consommateurs ont de certaines marques. Cette influence a aussi des effets sur la fidélisation des consommateurs.
- L’impact de la promotion sur l’acquisition et la fidélisation de nouveaux consommateurs, notamment selon les différents profils. Par exemple, un consommateur promophile changera régulièrement de pack si ceux-ci sont en promotions.

4.1 L’augmentation du volume des ventes

C’est le phénomène le plus classique, lors d’une promotion il y a une augmentation du volume des ventes du pack en promotion. Notre expérience consiste à comparer le volume des ventes d’un pack, en promotion puis sans promotion.

La figure 2 montre l’évolution du volume des ventes des neuf packs d’une catégorie observée. Sur une réduction de 40% du prix, nous observons en moyenne une augmentation de 50% du volume des ventes sur ce pack.

Ce phénomène est observable peu importe la promotion ou le pack. On peut observer en figure 3 que les résultats diffèrent selon le niveau de promotion, mais génèrent à chaque fois un pic de ventes allant de 40 à 80% d’augmentation. La seule exception est lors d’une promotion à 5%, sur ce jeu de paramètres, le changement n’est pas assez conséquent pour influencer suffisamment les clients. Ces résultats donnent une élasticité

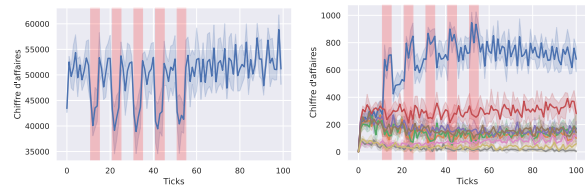


FIGURE 4 – À gauche le chiffre d’affaires, à droite le nombre de ventes. La même promotion sur le même pack est répété plusieurs fois au temps 30 à 34, 40 à 44, ... jusqu’au temps 60 à 64

prix du pack allant de -4 à -1.33. Ces résultats sont du même ordre de grandeur que les élasticités réellement observées par [4, 11].

On observe que les résultats d’une promotion dans notre modèle sont très dépendants du contexte de la promotion. Ce contexte est défini par les paramètres du modèle, les agents et les packs, par exemple :

- La proportion des différents profils d’agents.
- Les paramètres des packs et la similarité entre les packs.
- Le nombre de packs en concurrence (d’une même catégorie).

En effet, si le modèle dispose de 5 profils différents en proportion égale, on observe une augmentation de ventes de 95% lors de la promotion et de 54% après la promotion. Cependant, si le modèle contient 70% d’agents promophiles et 30% d’agents équilibrés, on observe une augmentation des ventes de 173% lors de la promotion et de 81% après la promotion.

Si on applique une promotion sur un pack d’une catégorie ne disposant que de deux packs similaires, on obtient une augmentation des ventes de 40% lors de la promotion et de 27% après la promotion. Si les deux packs sont bien différents, par exemple un pack peu cher de faible qualité et un pack très cher mais de haute qualité, on obtient une augmentation des ventes de 7.5% lors de la promotion et de 12.5% après la promotion si la promotion est sur le pack de faible qualité. Si la promotion est sur le pack de haute qualité, on obtient 31% d’augmentation des ventes lors de la promotion et 17% d’augmentation des ventes après la promotion.

Enfin, si on applique une promotion sur un pack parmi 9 on obtient une augmentation de ventes de 95% lors de la promotion et de 54% après la promotion. si on applique cette même promotion sur le même pack parmi 3 on obtient une

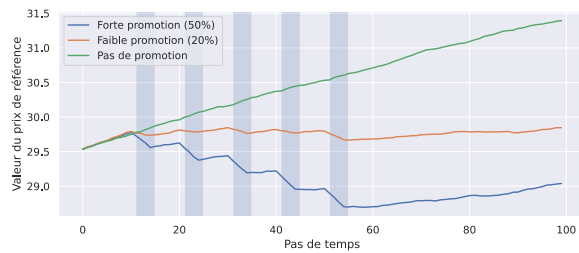


FIGURE 5 – Le prix de référence moyen en fonction du temps. On observe une diminution du prix de référence moyen lors de promotions, ce qui a pour effet d'augmenter l'attractivité des packs les moins chers.

augmentation de ventes de 31.5% lors de la promotion et de 15% après la promotion. Avec ce modèle, on peut ainsi quantifier les effets des promotions et les relier à une situation particulière, un contexte, qui peut être réel.

4.2 Baisse du volume des ventes

La baisse des ventes des packs qui ne sont pas en promotion est un phénomène aussi couramment observé. On dit que le pack en promotion "cannibalise" les ventes des autres packs du même type. On peut observer cet effet sur la figure 2 : la baisse des ventes des packs ne bénéficiant pas de promotions va de 2 à 20 % en moyenne et varie selon les packs et leur similarité avec le pack en promotion. La baisse des ventes est pour au moins 1 pack toujours supérieur à 0%.

4.3 Impact des promotions répétées

Des promotions répétées rapidement diminuent le pic d'augmentation des ventes généré chez les clients lors d'une promotion [5]. Plus les clients sont habitués aux promotions, moins ces clients changent de pack lors d'une promotion. À l'inverse, si les promotions sont exceptionnelles, alors le comportement des clients change. Ils voudront davantage profiter de la promotion, c'est ce qu'on observe en figure 4. Sur chaque simulation, nous obtenons une augmentation du nombre de ventes de 140% sur la première promotion pour atteindre une augmentation de 30% sur la dernière promotion. Enfin, nous avons pu observer un phénomène de diminution du prix du pack de référence des agents, ce qui a pour effet de diminuer leurs fonctions d'utilités. Cette diminution affecte la quantité que les agents achètent et avantage les packs les moins chers.

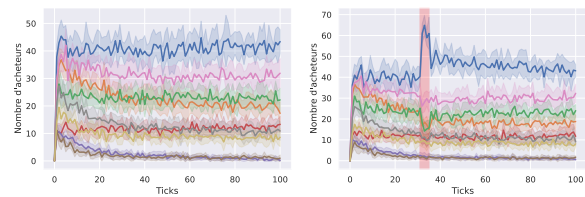


FIGURE 6 – Nombre d'acheteurs pour chaque pack en fonction du temps dans une simulation. À gauche sans promotion, à droite avec une promotion de 40% entre les pas de temps 30 et 34.

4.4 Acquisition de nouveaux clients

Sur la figure 6 nous montrons qu'une promotion entraîne sur le pack en promotion une augmentation du nombre de clients suivie d'une baisse au moment de l'arrêt de la promotion, puis d'une nouvelle augmentation à plus long terme correspondant à une fidélisation. La figure 7 représente le nombre d'agents pour chaque profile ayant acheté le pack P_0 et le pack P_3 . Le pack P_0 est le seul à recevoir une promotion. Ainsi, on observe l'impact de la promotion selon les différents profils d'agents sur P_0 . Nous montrons que lors d'une promotion, il y a une augmentation du nombre d'acheteurs (50% en moyenne) sur les profils de consommateurs promophiles uniquement (intéressés particulièrement par la publicité). Nous en concluons qu'une partie des agents, notamment les plus promophiles, ont acheté le pack P_0 plutôt qu'un autre, parce que le pack était en promotion. On note aussi que les agents sensibles à la qualité achètent peu pour le pack P_0 et inversement ces agents achètent beaucoup pour le pack P_3 car le pack P_3 est plus cher, mais de meilleure qualité. La promotion a très peu d'impact sur cet effet. Ce phénomène s'observe aussi sur les tables 1,2,3 qui montrent, l'évolution selon les profils d'agents, des ventes pendant et après la promotion, comparées aux ventes avant la promotion.

4.5 Impact de la guerre des prix

Afin d'explorer les réactions du modèle face à une guerre des prix, nous réalisons une simulation plus simple dans laquelle il n'y a que deux packs correspondant à la figure 8. Dans cette situation, nous notons une baisse du chiffre d'affaires dès qu'une réduction de prix a lieu, même s'il n'y a une baisse de prix que d'un seul pack. De plus, nous notons que lors d'une guerre des prix, le pack qui baisse le plus son prix récupère la plus grande part de marché, mais les deux

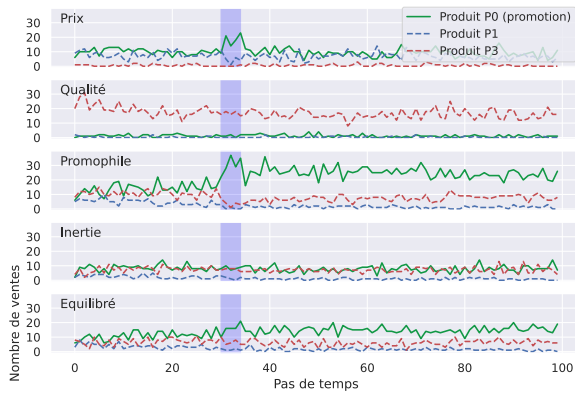


FIGURE 7 – Evolution du nombre d’acheteurs lors d’une promotion de 40% sur P_0 en $30 \leq t \leq 40$. On observe une augmentation durable des ventes sur le pack P_0 chez les promophiles et une augmentation temporaire chez les agents portés sur le prix

packs perdent en chiffre d’affaires.

5 Discussion

Dans cet article, nous montrons dans la partie 4 que l’approche centrée individus proposée dans la partie 3 est capable de reproduire des phénomènes émergents connus en marketing comme l’augmentation du volume des ventes, la "cannibalisation" lié à la concurrence ou encore les changements chez les clients que provoque la répétition rapide de promotions. De plus, l’approche centrée individus permet de montrer des phénomènes qui ne sont observables qu’au niveau des individus comme la fidélisation lors d’une promotion ou les effets des guerres de prix directement sur les consommateurs. Conformément à nos hypothèses, le modèle et nos résultats s’appliquent principalement aux biens de consommation courants. D’autres hypothèses permettraient d’adapter le modèle à d’autres types de biens. Afin d’approfondir le modèle, il est possible d’ajouter un système d’influence sociale similaire à ceux décrits dans la partie 2 qui permettrait aux agents d’échanger et d’interagir entre eux afin de s’influencer. Dans un tel système, une discussion est à prévoir sur les moyens d’influence des agents (système de graphe par exemple) et de leurs effets (est-il pertinent de faire varier des sensibilités personnelles?). De plus, il est possible de ne donner aux agents qu’une connaissance partielle des packs, ainsi les agents devraient découvrir eux même les packs qu’ils ne connaissent pas ou être influencé socialement. Enfin, il serait intéressant d’étudier les

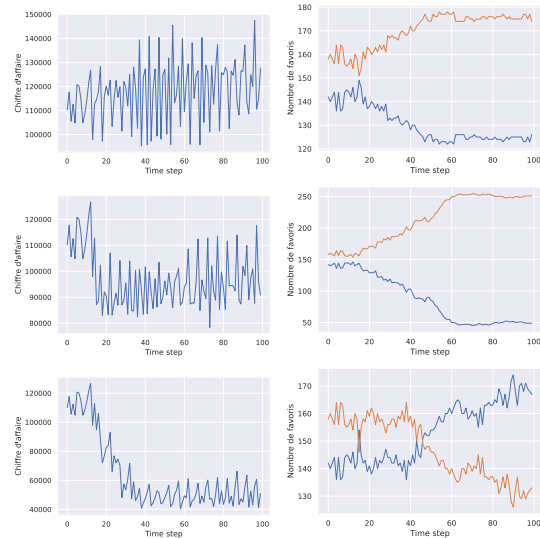


FIGURE 8 – À gauche le chiffre d’affaires, à droite pour chaque pack le nombre d’agents ayant ce produit en "favoris" (le plus acheté par l’agent). De haut en bas, une simulation sans changement de prix, une où seul une marque baisse son prix de 20% à partir de $t = 15$ et une où les 2 packs diminuent leurs prix à certains pas de temps.

Profil d’agent	Évolution pendant la promotion	Évolution après la promotion
Prix	88,10%	-9,8%
Qualité	1,4%	-9,4%
Promophile	132,6%	79,1%
Inertiels	0,4%	-8,7%
Equilibré	57%	35,2%

TABLE 1 – Compare la période de la promotion (respectivement après) avec la période avant la promotion de 40% du pack P_0 .

effets sur les promotions de la similarité et de la concurrence entre produits.

Finalement, le modèle proposé dispose de la possibilité d’intégrer facilement des données réelles (via l’historique) ce qui permettrait d’améliorer le réalisme, mais surtout d’appliquer le modèle à des scénarios concrets. Nous justifions son adaptabilité aux données grâce aux paramètres globaux intégrés au modèle.

Références

[1] Robert L Axtell and J Doyne Farmer. Agent-based modeling in economics and

Profil d'agent	Évolution pendant la promotion	Évolution après la promotion
Prix	-48,32%	-12,25%
Qualité	-76,92%	-23,67%
Promophile	-92,81%	-71,39%
Inertiels	-47,40%	-63,74%
Équilibré	-63,84%	-60,23%

TABLE 2 – Compare la période de la promotion (respectivement après) avec la période avant la promotion de 40% du pack P_1 .

Profil d'agent	Évolution pendant la promotion	Évolution après la promotion
Prix	-8%	-3%
Qualité	-82%	-42%
Promophile	55%	22%
Inertiels	-10%	-22%
Équilibré	18%	6%

TABLE 3 – Même expérience que pour la table 1 mais avec 3 packs dans la catégorie au lieu de 9.

finance : Past, present, and future. *Journal of Economic Literature*, 2022.

- [2] Frank M Bass. A new product growth for model consumer durables. *Management science*, 15(5) :215–227, 1969.
- [3] Kapil Bawa. Modeling inertia and variety seeking tendencies in brand choice behavior. *Marketing science*, 1990.
- [4] Tammo HA Bijmolt, Harald J Van Heerde, and Rik GM Pieters. New empirical generalizations on the determinants of price elasticity. *Journal of marketing research*, 42(2) :141–156, 2005.
- [5] Robert C Blattberg, Richard Briesch, and Edward J Fox. How promotions work. *Marketing science*, 1995.
- [6] Neil H Borden. The concept of the marketing mix. *Journal of advertising research*, 4(2) :2–7, 1964.
- [7] Víctor Cantillo, Juan de Dios Ortuzar, and Huw CWL Williams. Modeling discrete choices in the presence of inertia and serial correlation. *Transportation Science*, 41(2) :195–205, 2007.
- [8] Maxime Cohen, Swati Gupta, Jeremy Kalas, and Georgia Perakis. An efficient algorithm for dynamic pricing using a graphical representation. *Production and Operations Management*, 29(10) :2326–2349, 2020.
- [9] Sebastiano A Delre, Wander Jager, Tammo HA Bijmolt, and Marco A Janssen. Targeting and timing promotional activities : An agent-based model for the takeoff of new products. *Journal of business research*, 60(8) :826–835, 2007.
- [10] Bruce GS Hardie, Eric J Johnson, and Peter S Fader. Modeling loss aversion and reference dependence effects on brand choice. *Marketing science*, 12(4) :378–394, 1993.
- [11] Stephen J Hoch, Byung-Do Kim, Alan L Montgomery, and Peter E Rossi. Determinants of store-level price elasticity. *Journal of marketing Research*, 32(1) :17–29, 1995.
- [12] Phan Duy Hung, Nguyen Duc Ngoc, and Tran Duc Hanh. K-means clustering using ra case study of market segmentation. In *Proceedings of the 2019 5th International Conference on E-Business and Applications*, pages 100–104, 2019.
- [13] Wander Jager. The four p's in social simulation, a perspective on how marketing could benefit from the use of social simulation. *Journal of Business Research*, 60(8) :868–875, 2007.
- [14] Amit Khandelwal. The long and short (of) quality ladders. *The Review of Economic Studies*, 77(4) :1450–1476, 2010.
- [15] Ashkan Negahban and Levent Yilmaz. Agent-based simulation applications in marketing research : an integrated review. *Journal of Simulation*, 8(2) :129–142, 2014.
- [16] Lamjed Ben Said, Thierry Bouron, and Alexis Drogoul. Agent-based interaction analysis of consumer behavior. In *Proceedings of the first international joint conference on AAMAS : part 1*, pages 184–190, 2002.
- [17] PB Seetharaman and Pradeep Chintagunta. A model of inertia and variety-seeking with marketing variables. *International Journal of Research in Marketing*, 15(1) :1–17, 1998.
- [18] Gerard J Tellis. Modeling marketing mix. *Handbook of marketing research*, pages 506–522, 2006.
- [19] Richard Wigren and Filip Cornell. Marketing mix modelling : A comparative study of statistical models, 2019.

Modèle explicatif de la sécession des experts dans les communautés de pratiques

Sébastien Delarre^a

sebastien.delarre@univ-lille.fr

Fabien Eloire^a

fabien.eloire@univ-lille.fr

Antoine Nongaillard^b

antoine.nongaillard@univ-lille.fr

Maxime Morge^b

maxime.morge@univ-lille.fr

^aUniv. Lille, CNRS, UMR 8019 Clersé, F-59000 Lille, France

^bUniv. Lille, CNRS, Centrale Lille, UMR 9189, CRISTAL, F-59000 Lille, France

Résumé

Les sites de questions-réponses permettent aux utilisateurs, qui partagent les mêmes intérêts, d'échanger des connaissances dans le cadre d'un processus d'apprentissage continu. En particulier, la plateforme Stack Overflow soutient des communautés de pratique où : (a) la majorité du contenu est produit par un petit nombre de membres très actifs ; (b) les réponses aux questions les plus récurrentes sont rapidement capitalisées. Nous proposons un modèle centré individus d'une communauté de pratique présentant une dynamique sociale conforme à ces faits stylisés et qui explique le phénomène de sécession des experts.

Mots-clés : Simulation multi-agents, Analyse des réseaux sociaux, Système complexe, Explicabilité

Abstract

Communities of Practice services enable users that share same interests and exchange knowledge in an ongoing learning process. The Stack Overflow platform supports such communities where : (a) the majority of the content are produced by a small number of highly active users ; (b) the answers to the most common questions are quickly capitalized on. We propose an agent-based model of a Community of Practice exhibiting a social dynamics that is compliant with these insights and which explains the phenomenon of secession of experts.

Keywords: Agent-based modeling, Social network analysis, Complex system, Explainability

1 Introduction

Une communauté de pratique - en anglais *Community of Practice* (CoP) - est « un groupe de personnes qui partagent une préoccupation, un ensemble de problèmes ou une passion pour un su-

jet donné, et qui approfondissent leurs connaissances et leur expertise dans ce domaine en interagissant de manière continue » [17]. L'Internet a contribué à la création de communautés de pratique distribuées - en anglais *Distributed Community of Practice* (DCoP) - dont les membres sont géographiquement répartis et utilisent des moyens technologiques pour interagir à distance les uns avec les autres. Par exemple, Stack Overflow¹ n'est pas seulement un simple site de questions-réponses sur la programmation, mais aussi une plateforme de travail collaboratif qui structure des DCoPs.

La formation et la structuration des DCoPs résultent des phénomènes sociaux complexes. Nous nous intéressons ici tout particulièrement au phénomène de sécession des experts. Au-delà du fait que chaque année un tiers des utilisateurs des Stack Overflow deviennent inactifs [12], on observe que, pour la plupart des communautés sur cette plateforme, 80 des 100 réponses les mieux notées sont données au cours de la phase de formation de la communauté (cf. figure 1). Les phénomènes sociaux complexes qui se jouent sur Stack Overflow sont le résultat de l'imbrication de comportements élémentaires microscopiques qui conduit à des dynamiques complexes mésoscopiques (au niveau de la communauté) et macroscopiques (au niveau de la plateforme). Nous soutenons que l'approche centrée individus est appropriée pour l'étude des DCoPs car elle permet de conceptualiser, de calibrer et de simuler ces phénomènes complexes à l'aide de systèmes auto-organisés d'agents coopératifs aux perceptions et à la rationalité limitées, qui sont structurés par l'effet des interactions. Le contrôle de tous les paramètres de cette communauté artificielle permet d'observer *ceteris paribus* les changements induits [7]. Le modèle visé doit être explicatif en

1. <https://stackoverflow.com>

proposant une restitution visuelle de ses propriétés et prescriptif en permettant d'introspecter les choix de conception de la plateforme pour favoriser l'innovation et l'intelligence collective.

Nous proposons ici un modèle centré individus d'une DCoP² qui présente une dynamique sociale où : (a) l'activité est inégalement distribuée parmi les membres ; (b) les réponses aux questions les plus récurrentes sont rapidement capitalisées. Nos résultats expérimentaux montrent que notre modèle est non seulement capable de reproduire ces faits stylisés mais également que la capitalisation explique le phénomène de sécession des experts qui se traduit par un effondrement qualitatif des réponses. Le principal verrou scientifique auquel nous nous sommes confrontés réside dans la calibration à partir de données ouvertes et massives pour la validation du modèle multi-agents de par son caractère multi-niveaux, l'aspect individuel des comportements et les boucles de rétro-action [1, 6, 9, 15]. C'est la raison pour laquelle nous nous sommes conformés au principe de parcimonie. En réduisant le nombre de paramètres et en limitant les mécanismes aux plus simples et aux plus généraux, nous proposons un modèle explicatif fondé sur l'identification des processus de solidarité locale et des comportements sociaux individuels sous-jacents.

Après un aperçu des travaux connexes dans la section 2, nous formalisons notre modèle centré individus, en particulier les comportements de demandeur et d'aidant (cf. section 3). La section 4 décrit notre implémentation de ce modèle. La section 5 présente les résultats obtenus par le modèle à partir de sa calibration sur des données réelles et massives. Enfin, la section 6 résume nos contributions, identifie les limites de notre modèle et présente nos perspectives.

2 Travaux connexes

L'étude et la modélisation des DCoPs n'est pas novateur. Ces dernières ont fait l'objet d'analyses statistiques, d'analyses de réseaux sociaux ou d'analyses temporelles mais de peu de modélisation à base d'agents.

Moutidis et Williams proposent la première grande analyse de l'évolution des communautés sur Stack Overflow [12]. Cette étude statistique de l'apparition, de la perpétuation et de la transformation des DCoPs s'appuie sur des algorithmes de détection de communautés d'utili-

sateurs en fonction de la similitude des étiquettes sur leurs contributions. À l'inverse, dans notre modèle, le réseau social réifie les interactions (questions/réponses) résultantes du comportement des individus. À l'aide de métriques macroscopiques comme le nombre de contributions ou le score total des contributions par période, cette étude observe que chaque année 34 % des utilisateurs deviennent inactifs sans pour autant expliquer ce phénomène.

L'analyse des réseaux sociaux pour les DCoPs vise à automatiser l'identification des figures d'autorité dans les communautés. Conçu pour le web, les algorithmes classiques tels que PageRank et HITS s'avèrent inadaptés aux réseaux sociaux, en particulier les DCoPs dont les topologies sont hétérogènes et se distinguent du web [3]. Bouguessa et *al.* observent que le nombre de contributions identifiées comme la meilleure réponse à une question est la métrique mésoscopique la plus appropriée pour évaluer le niveau d'autorité de l'aidant. L'analyse des réseaux sociaux permet également l'identification de stéréotypes parmi les utilisateurs. Yang et *al.* observent qu'une petite proportion d'utilisateurs est responsable de la majorité des contributions [19]. Ils mesurent également la qualité des réponses à l'aide d'une métrique microscopique, i.e. la contribution experte moyenne. Cela leur permet de distinguer deux stéréotypes de comportement individuel parmi les 10 % des aidants les plus actifs :

- les « moineaux » qui sont majoritaires, très réactifs vis-à-vis des questions simples mais avec des réponses qui ne sont pas nécessairement les plus pertinentes ;
- les « hiboux » qui sont minoritaires, pas nécessairement les plus réactifs mais dont les réponses sont les plus pertinentes et dont les questions sont difficiles.

Yang et *al.* interprète la disparition précoce des « hiboux » et l'arrivée continuelle de nouveaux « moineaux » comme le résultats des incitations à la gamification de plateforme. Toutefois, les résultats d'une récente enquête [8] suggèrent un effet limité des badges et des privilèges de réputation sur la motivation des utilisateurs. Notre modèle montre que ces phénomènes sont le résultat de la capitalisation des réponses aux questions les plus récurrentes.

L'analyse temporelle des DCoPs étudie l'évolution des membres, des contributions et des interactions. Pal et *al.* propose un modèle probabiliste de sélection des questions auxquelles répondre [13, 14]. Afin de maximiser la probabi-

2. Cet article est une version étendue en français de [4].

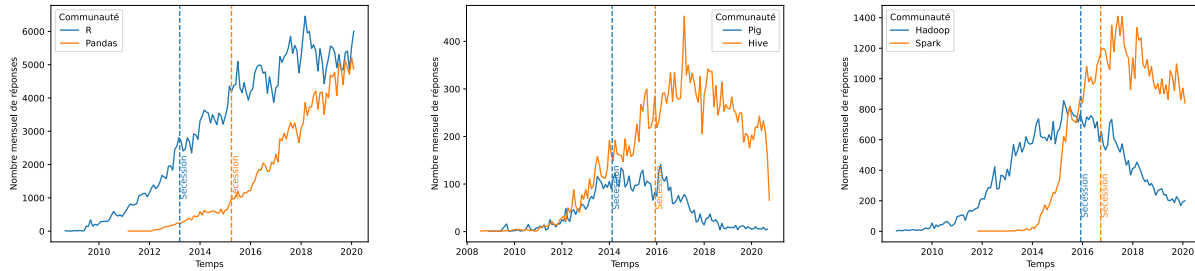


FIGURE 1 – Pour 5 des 6 communautés/thématiques représentées ici en face à face (R/Pandas, Pig/Hive, Hadoop/Spark), on observe que la date de sécession des experts, i.e. la date à laquelle 80 des 100 réponses les mieux notées sont données, intervient dans la phase de formation, i.e. quand le nombre mensuel de réponse croît

lité que leur réponse soit bénéfique pour la communauté, les aidants, qui sont solidaires, considèrent la valeur d’une question comme la valeur globale des réponses à cette question et la valeur d’une réponse comme le bénéfice qu’elle apporte par rapport aux réponses antérieures. C’est sur ce principe que s’appuie le mécanisme de sélection de questions des aidants dans notre modèle. Il est important de noter que cette étude conclut que l’explication du phénomène de sécession des experts est une question ouverte à laquelle notre modèle de capitalisation répond.

Alors que les technologies multi-agents soutenant les DCoPs sont nombreuses et variées [16], les modèles à base d’agents pour les DCoPs sont beaucoup plus rares. Zhang et *al.* adoptent une approche centrée individus pour générer le réseau social [21] et le *CommunityNetSimulator* [20] pour l’analyser. Contrairement à [2], les liens ne sont pas créés uniquement sur la base du principe d’attachement préférentiel selon lequel les nouveaux noeuds se connectent plus facilement à ceux qui ont déjà un grand nombre de connexions comme c’est le cas dans la plupart des réseaux sociaux, mais le modèle considère les liens comme résultants d’interactions (questions/réponses) entre des comportements individuels qui sont à leur tour guidés par les niveaux d’expertise. Afin de peupler la simulation, Zhang et *al.* considèrent, sans calibration, que la distribution de l’expertise est de type loi de puissance. Dans ce modèle, seuls les novices posent des questions et la probabilité de répondre à une question croît de manière exponentielle avec la différence de niveau d’expertise entre le demandeur et l’aidant. Contrairement à ce dernier, notre modèle numérique est calibré à partir des données réelles massives. À la différence de notre modèle, celui

introduit par [21] ne permet pas de reproduire pas le phénomène d’inégalité de l’activité d’une communauté parmi les membres et il n’explique pas le phénomène de sécession des experts.

3 Modèle

Afin de simuler une DCoP, nous modélisons une population constante de N agents. Chaque pas de simulation t correspond à une période bihebdomadaire. Au cours d’une période, chaque agent peut publier un certain nombre de contributions, des questions et/ou des réponses.

Chaque contribution c a un identifiant $id(c)$ – i.e. une clé artificielle distincte pour les questions et les réponses, un horodatage $time(c)$ – i.e. le pas de simulation au cours duquel elle est émise, un auteur $author(c)$ et un score $score(c)$ – i.e. une évaluation numérique de sa fréquence (pour une question) ou de sa pertinence (pour une réponse). De plus, chaque réponse aborde une question, $reply(a)$.

L’idiosyncrasie d’un agent détermine ses actions. Concrètement, le comportement de chaque agent a_i est spécifié par deux paramètres individuels :

- $\overline{nQ(a_i)}$, i.e. le nombre moyen de questions par période ;
- $\overline{nA(a_i)}$, i.e. le nombre moyen de réponses par période.

Le nombre de questions publiées à l’instant t , noté $nQ(a_i, t)$, est issu d’une distribution normale de moyenne $\overline{nQ(a_i)}$ et il est borné par un nombre maximum de questions par pas de simulation, $maxQ$. Le nombre de réponses données à l’instant t , noté $nA(a_i, t)$, est issu d’une distribution normale de moyenne $\overline{nA(a_i)}$ et il est

borné par un nombre maximum de réponses par pas de simulation, $maxA$.

Afin de peupler la simulation, nous considérons, comme le laisse supposer nos données de calibration (cf. section 5.1), que la distribution du nombre bihebdomadaire de réponses d'un agent est de type loi de puissance :

$$\overline{nA(a_i)} = maxA \times rang^{\overline{nA}}(a_i)^{k_A} \text{ avec } k_A < 0 \quad (1)$$

où $rang^{\overline{nA}}(a_i)$ dénote le rang de l'agent a_i par nombre moyen de réponses bihebdomadaires décroissant et le degré de la loi k_A est constant. Comme [19], notre étude de *Stack Overflow* montre que les aidants, qui posent également des questions, sont rares et peu actifs. C'est la raison pour laquelle nous générons $\overline{nQ(a_i)}$ comme $\overline{nA(a_i)}$, et de telle sorte que plus un agent donne de réponses, moins il pose de questions. Formellement, $rang^{\overline{nQ}}(a_i) = N + 1 - rang^{\overline{nA}}(a_i)$ où $rang^{\overline{nQ}}(a_i)$ dénote le rang de l'agent a_i par nombre moyen de questions bihebdomadaires décroissant.

Afin de déterminer le score d'une question, nous considérons, comme le laisse supposer nos observations (cf. section 5.3), une distribution qui est de type loi de puissance :

$$score(q) = maxS_q \times (id(q))^{k_{S_q}} \text{ avec } k_{S_q} < 0 \quad (2)$$

où le score maximal d'une question $maxS_q$ et le degré de la loi k_{S_q} sont constants.

La simulation d'une période t est divisée en deux phases (cf. figure 2). Pendant la phase de question, chaque agent a_i pose $nQ(a_i, t)$ questions. Pendant la phase de réponse, chaque agent réagit à $nA(a_i, t)$ questions posées par les autres agents³. Comme [13, 14], nous supposons que les agents visent à maximiser le bénéfice de leur réponse par rapport aux réponses antérieures. C'est la raison pour laquelle les agents choisissent de répondre aux questions les plus fréquentes. Parce que nous supposons qu'une réponse est d'autant plus pertinente que la question est fréquente, le score de la première réponse est proportionnel au score de cette question (k_{S_a}). Comme nous supposons qu'une réponse vient compléter à la marge les réponses précédentes à la même question, le score décroît de manière exponentielle à chaque nouvelle réponse (e_{S_a}).

3. Les variances sont suffisamment faibles pour maintenir la forme de la distribution de type loi de puissance pour le nombre moyen de messages publiés par période.

4 Simulateur

Notre simulateur ABM4DCoP [11] est fondé sur NetLogo [18] qui est un langage de programmation multi-agents et un environnement de développement intégré pour la conception de simulations à base d'agents. Même si NetLogo est convivial, il permet la conception de modèles de haut niveau avec des comportements complexes.

Comme le montre la figure 3, l'interface de notre simulateur se compose de trois types de composants pour :

- configurer les paramètres de la simulation ;
- visualiser les agents, les interactions et les liens ;
- analyser l'évolution des métriques qui sont calculées et affichées en temps réel pendant la simulation.

Les agents sont disposés en cercle dans le sens des aiguilles d'une montre par ordre croissant de $\overline{nA(a_i)}$. La représentation graphique de la communauté dans la figure 4 nous permet d'observer la distribution spatiale de la productivité des agents (en violet) et de leurs liens (en gris). Un arc représente l'aide apportée par l'agent source à l'agent destinataire. Le degré sortant d'un noeud est le nombre de pairs aidés et le degré entrant est le nombre de conseillers. Au cours de la simulation, les publications s'accumulent sur le segment radial de son auteur : les questions sont en vert et les réponses en rouge.

5 Calibration et simulation

Après une brève description du jeu de données utilisé pour la calibration, nous présentons nos résultats de simulation qui montrent que : (a) notre modèle permet de reproduire l'inégale répartition de l'activité au sein d'une communauté ; (b) la capitalisation des réponses aux questions les plus récurrentes explique l'effondrement qualitatif des réponses.

5.1 Données de calibration

Afin d'explorer notre terrain d'étude *Stack Overflow*, nous avons extrait, transformé et chargé les données brutes produites de 2008 à 2021 (~50 GiO) dans un entrepôt de données [10]. Notre modèle conceptuel multidimensionnel permet d'analyser des indicateurs clés de performance tels que le nombre de publications, la discutabilité des questions ou le temps de réponse. Le modèle logique multidimensionnel est alimenté par 6 communautés en face à

Données : a_i, t : agent/pas de temps
 $H = (Q, A)$: historique des questions/réponses
Résultat : H' : historique
 $nQ(a_i, t) = \mathcal{N}(\mu = \overline{nQ(a_i)}, \sigma^2 = 0.02)$;
si $randomU(0, 1) \leq \{nQ(a_i, t)\}$
alors
 $nQ(a_i, t) += 1$;
pour $j \in [1; nQ(a_i, t)]$ **faire**
 $id(q) = |\{q \in Q\}| + 1$;
 $time(q) = t$;
 $author(q) = a_i$;
 $score(q) = maxS_q \times (id(q))^{kS_q}$;
 $Q \cup = q$;
retourner H

Données : a_i, t : agent/pas de temps
 $H = (Q, A)$: historique de questions/réponses
Résultat : H' : historique
 $nA(a_i, t) = \mathcal{N}(\mu = \overline{nA(a_i)}, \sigma^2 = 0.02)$;
si $randomU(0, 1) \leq \{nA(a_i, t)\}$ **alors**
 $nA(a_i, t) += 1$;
 $Q_O = \langle q_1, \dots, q_n \rangle$;
 /* où Q_O est la liste des
 $q_j \in \{q \in Q \mid author(q) \neq a_i\}$ ordonnée par
 $\frac{kS_a \times score(q_j)}{|\{a \in A \mid reply(a) = q_j\}| + 1}$ décroissant */
pour $j \in [1; nA(a_i, t)]$ **faire**
 /* Pour les questions les plus
 fréquentes Q_O */
 $id(a) = |\{a \in A\}| + 1$;
 $time(a) = t$;
 $author(a) = a_i$;
 $reply(a) = q_j$;
 /* Le score d'un réponse est
 proportionnel au score de la question
 et décroît de manière exponentielle à
 chaque nouvelle réponse */
 $score(a) = \mathcal{N}(\mu = \frac{kS_a \times score(q_j)}{|\{a \in H \mid reply(q_j)\}| + 1}, \sigma^2 = 0.05)$;
 $A \cup = a$;
retourner H

FIGURE 2 – Comportement individuel de demandeur (à gauche) et d’aidant (à droite)

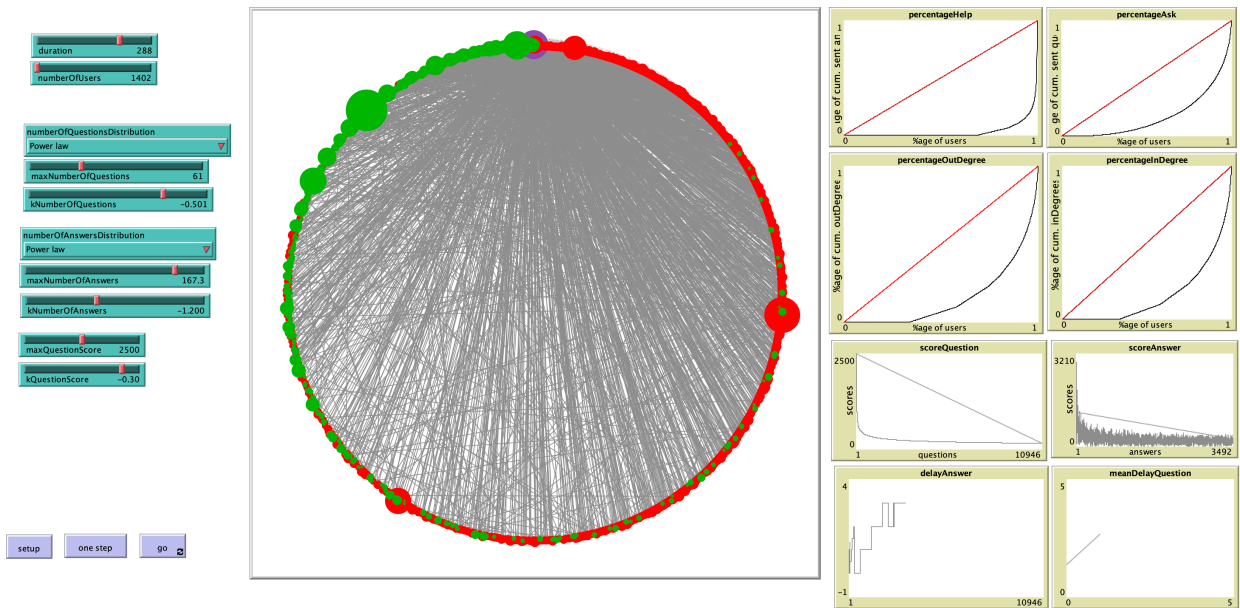


FIGURE 3 – Configuration (à gauche), visualisation des agents disposés en cercle (au centre) et restitution de la simulation

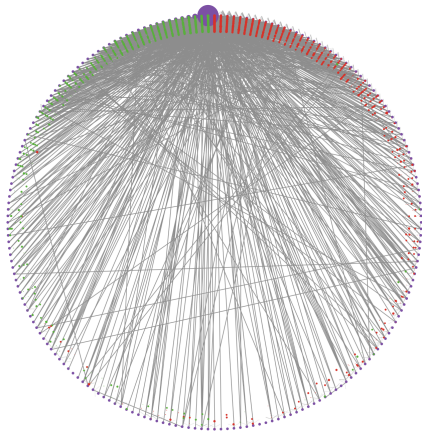


FIGURE 4 – Réseau communautaire quasi biparti

face : Pig/Hive, Hadoop/Spark et R/Pandas. En particulier, ces deux dernières communautés représentent environ 1 300 000 de messages publiés par près de 250 000 membres. Nous avons écarté 2000 publications (0,15 %) impliquant 4000 utilisateurs qui ont supprimé leur compte et pour lesquels nous ne disposons pas de données attributaires. De plus, l'activité après les premiers confinements (mars 2020) est exclue.

Afin de calibrer notre modèle sur la communauté R à partir de nos données réelles, nous considérons que le nombre maximum de questions (resp. réponses) bihebdomadaires est $maxQ = 61$ (resp. $maxA = 167, 3$) et $k_Q = -0, 5$ (resp. $k_A = -1, 2$). Ainsi, notre modèle coïncide avec les données de calibration comme l'illustre la figure 5 qui représente le nombre moyen de réponses par période et par membre.

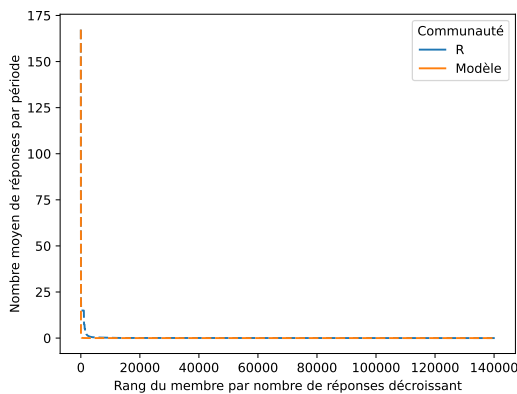


FIGURE 5 – Modèle calibré à partir de données réelles

5.2 Inéquité des contributions

La figure 6 représente la demande et l'aide cumulée des utilisateurs du plus petit au plus grand nombre moyen de questions (resp. réponses) par période, $\overline{nQ}(a_i)$ (resp. $\overline{nA}(a_i)$). Nous avons réimplémenté CommunityNetSimulator [5] avec NetLogo. Selon ce modèle, 20 % des membres les plus actifs sont responsables de plus de 66 % des réponses. Cependant, nous observons que 20 % des membres les plus actifs de la communauté R sont les auteurs de 94 % des réponses sur ce sujet. Selon notre modèle, l'inéquité de la distribution est similaire.

Dans les communautés en ligne, très peu de membres publient la majorité du contenu. Ce phénomène mésoscopique est partagé par tous les DCoPs que nous avons étudiés sur Stack Overflow. L'ordonnanceur pseudo-aléatoire et séquentiel de notre moteur, qui permet aux agents bavards de parler plus d'une fois par période, reproduit à dessein l'inégalité de l'activité événementielle.

5.3 Sécession

Notre modèle suppose ici que le score d'une question, qui est le résultat de votes en sa faveur (*upvote*) ou en sa défaveur (*downvote*), est proportionnelle à sa fréquence. Cette hypothèse est confirmée par la figure 7a qui représente pour chaque question posée dans les communautés R et Pandas, son score ainsi que le trafic web généré par son fil de discussion. On peut noter que, pour ne pas pénaliser les questions les plus récentes, nous considérons le nombre de vues par semaine d'ancienneté.

Dans les DCoPs que nous avons étudiés sur Stack Overflow, nous observons que les questions les mieux notées, c'est-à-dire les plus récurrentes, sont rapidement posées dans la phase de formation de la communauté. La figure 7b, qui représente le score des questions classées par ordre d'apparition, illustre cette observation. De plus, elle représente la distribution du score des questions de notre modèle (en vert) qui, à dessein, est de type loi de puissance.

Notre modèle suppose que le score de la meilleure réponse à une question, qui est le résultat de votes, est proportionnelle au score de la question à laquelle elle répond. Cette hypothèse est confirmée par la figure 7c qui représente pour chaque question à laquelle les communautés R et Pandas ont répondu, son score ainsi que celui

de sa meilleur réponse. C'est la raison pour laquelle notre modèle a été calibré avec $k_{S_a} = 1,5$. De manière similaire, nous avons calibré notre modèle avec $e_{S_a} = 2$.

Le phénomène de sécession des experts correspond à l'effondrement qualitatif rapide des réponses. La figure 7d qui représente le score des réponses classées par ordre d'apparition illustre ce phénomène dans les communautés R et Pandas. On y observe également que notre modèle, avec 1402 agents (1% de la communauté R) simulé pendant 2 ans, est capable de reproduire ce même fait stylisé. Notre modèle explicatif décrit ce phénomène comme le résultat de la capitalisation des réponses aux questions les plus récurrentes. Les questions les plus fréquentes sont rapidement posées dans la phase de formation de la communauté. Pour maximiser le score de leur réponse, le comportement individuel des agents consiste à répondre en premier lieu à ces questions. Dès que ces dernières ont une réponse, la pertinence des conseils décroît rapidement.

6 Discussion

Nous avons proposé ici un modèle explicatif de la sécession des experts dans les communautés de pratiques. Selon notre modèle centré individu, l'effondrement qualitatif précoce des réponses résulte de la capitalisation des réponses aux questions les plus récurrentes dans la communauté. Les membres, qui sont solidaires, évaluent la valeur d'une réponse proportionnelle à la valeur de la question et inversement proportionnelle au nombre de réponses pour cette même question. Leur comportement consiste à donner des réponses perspicaces aux questions les plus récurrentes qui sont rapidement posées puis complètent ces réponses pour finalement donner des réponses étroites aux questions les plus spécifiques. C'est la raison pour laquelle nous pensons que les mécanismes ludiques de gamification de la plateforme Stack Overflow devrait en premier lieu inciter les utilisateurs à poser de bonnes questions.

Bien que calibré à partir des données réelles massives, notre modèle se veut parcimonieux. En réduisant le nombre de paramètres et en limitant les mécanismes aux plus simples et aux plus généraux, nous proposons un modèle explicatif du phénomène de sécession.

Afin de calibrer la population de nos simulations, la population d'agents est homogène, i.e. modélisée par une loi de puissance, et leur expertise est

mesurée par leurs activités *ex post*. Cette modélisation statistique de la distribution des caractéristiques idiosyncrasiques pourrait être raffinée en distinguant les « moineaux » (les individus les plus actifs dont les réponses ne sont pas nécessairement les plus pertinentes) des « hiboux » (les experts du domaine qui ne sont pas nécessairement les plus actifs). Le verrou technologique auquel nous sommes désormais confrontés réside dans le passage à l'échelle du modèle. Jouer 2 ans d'échanges d'une communauté constituée de 1402 agents (1% de la communauté R) nécessite plusieurs dizaines d'heure de simulation.

À plus long terme, nous souhaitons étudier et confronter d'autres phénomènes mésoscopiques tels que l'efficacité avec le temps de réponse et la controverse mesurée par le caractère discutable des questions.

Remerciements. Nous adressons nos remerciements aux relecteurs qui par leurs remarques nous ont permis de clarifier notre modèle.

Références

- [1] Jérémy ALBOUYS-PERROIS, Nicolas SABOURET, Yvon HARADJI, Mathieu SCHUMANN et Christian INARD. « Simulation multi-agent de l'autoconsommation collective en relation avec l'activité des foyers ». In : *Actes des Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. Cépaduès, 2019, p. 129-138.
- [2] Albert-László BARABÁSI et Réka ALBERT. « Emergence of scaling in random networks ». In : *Science* 286.5439 (1999), p. 509-512.
- [3] Mohamed BOUGUessa, Benoit DUMOULIN et Shengrui WANG. « Identifying Authoritative Actors in Question-Answering Forums : The Case of Yahoo! Answers ». In : *Proc. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 2008, p. 866-874.
- [4] Amal CHAOUI, Sébastien DELARRE, Eloire FABIEN, Maxime MORGE et Antoine NONGAILLARD. « Toward an Agent-Based Model of a Community of Practice : Demonstration ». In : *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection*. T. 13616. LNAI. Springer, 2022, p. 467-472.

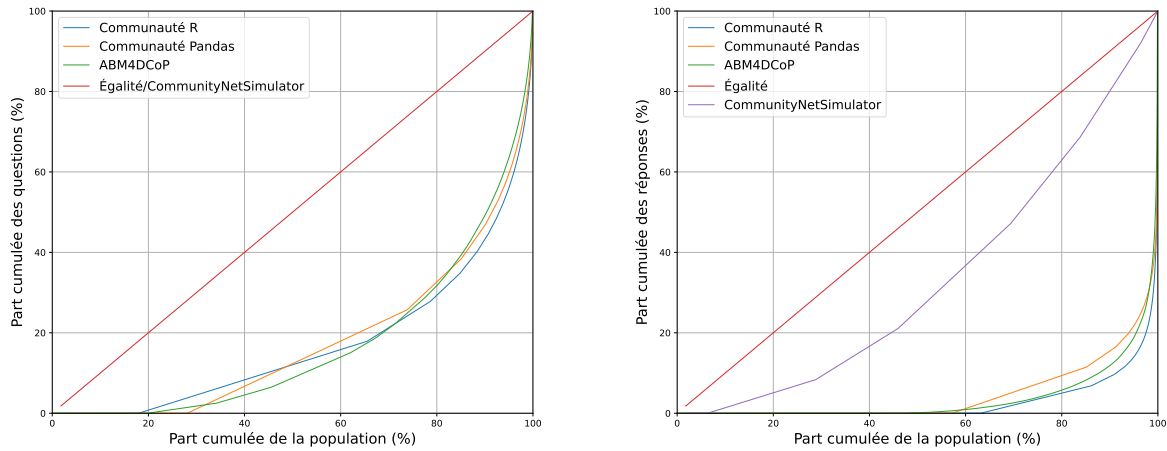
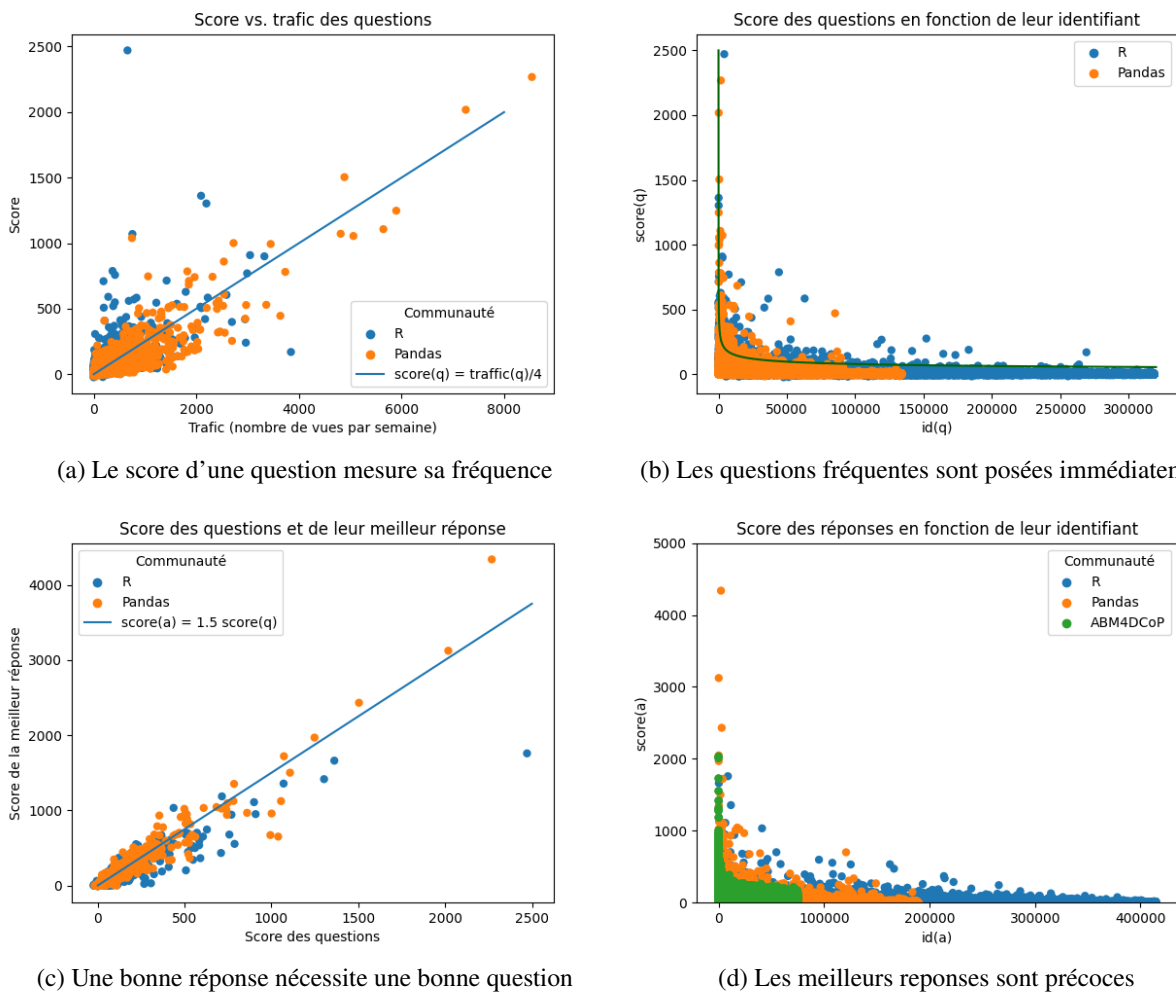


FIGURE 6 – Courbes de Lorenz qui montrent la répartition des questions (à gauche) et des réponses (à droite) au sein des communautés observées, simulées par CommunityNetSimulator et par ABM4DCoP



(a) Le score d'une question mesure sa fréquence

(b) Les questions fréquentes sont posées immédiatement

(c) Une bonne réponse nécessite une bonne question

(d) Les meilleurs reponses sont précoces

FIGURE 7 – Le phénomène de sécession des experts résulte de la capitalisation des réponses aux questions les plus récurrentes

- [5] Amal CHAOUI, Jarod VANDERLYNDEN et Maxime MORGE. *CommunityNetSimulator : Agent-Based Model for Simulating a Community*. <https://gitlab.univ-lille.fr/mocicos/communitynetsimulator>. Online; accessed 2023-03-01.
- [6] Kévin DARTY, Julien SAUNIER et Nicolas SABOURET. « Calibration de simulations multi-agents à l'aide d'une méthode semi-automatique d'analyse du comportement (présentation courte) ». In : *Actes des Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. Cépaduès Éditions, 2015, p. 205-214.
- [7] Nigel GILBERT, éd. *Simulating societies : the computer simulation of social phenomena*. UCL Press, 1994.
- [8] Nicholas HOERNLE, Gregory KEHNE, Ariel D. PROCACCIA et Kobi GAL. « The phantom steering effect in Q&A websites ». In : *Knowledge and Information Systems* 64.2 (2022), p. 475-506.
- [9] Philippe MATHIEU et Sébastien PICAULT. « Calibrer les comportements d'agents à partir de données réelles ». In : *Revue d'Intelligence Artificielle* 28.4 (2014), p. 463-484.
- [10] Maxime MORGE. *SoDyOnStack : An Analysis of Social Dynamics on Stack Overflow*. <https://gitlab.univ-lille.fr/mocicos/sodyonstack>. Online; accessed 2023-02-01.
- [11] Maxime MORGE et Amal CHAOUI. *ABM4DCOP : Agent-Based Modeling for Distributed Community Of Practice*. <https://gitlab.univ-lille.fr/mocicos/abm4dcop>. Online; accessed 2023-03-01.
- [12] Iraklis MOUTIDIS et Hywel T. P. WILLIAMS. « Community Evolution on Stack Overflow ». In : *PLOS ONE* 16.6 (17 juin 2021), e0253010. ISSN : 1932-6203.
- [13] Aditya PAL, Shuo CHANG et Joseph A KONSTAN. « Evolution of Experts in Question Answering Communities ». In : *sixth international AAAI conference on weblogs and social media*. T. 6. 1. 2012, p. 274-281.
- [14] Aditya PAL, F Maxwell HARPER et Joseph A KONSTAN. « Exploring question selection bias to identify experts and potential experts in community question answering ». In : *ACM Transactions on Information Systems (TOIS)* 30.2 (2012), p. 1-28.
- [15] Quentin REYNAUD, François SEMPÉ, Yvon HARADJI et Nicolas SABOURET. « Simuler l'activité humaine avec des approches statistiques basées sur les enquêtes emploi du temps ». In : *Actes des Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*. Cépaduès Éditions, 2017, p. 117-126.
- [16] Gilson Yukio SATO, Hilton José Silva de AZEVEDO et Jean-Paul A BARTHÈS. « Agent and multi-agent applications to support distributed communities of practice : a short review ». In : *Autonomous Agents and Multi-Agent Systems* 25.1 (2012), p. 87-129.
- [17] Etienne WENGER, Richard Arnold McDERMOTT et William SNYDER. *Cultivating communities of practice : A guide to managing knowledge*. Harvard business press, 2002.
- [18] WILENSKY, URI. *NetLogo*. <http://ccl.northwestern.edu/netlogo>. Online; accessed 2023-03-01.
- [19] Jie YANG, Ke TAO, Alessandro BOZZON et Geert-Jan HOUBEN. « Sparrows and owls : Characterisation of expert behaviour in stackoverflow ». In : *Proc. of International Conference on User Modeling, Adaptation, and Personalization*. Springer. 2014, p. 266-277.
- [20] Jun ZHANG, Mark S ACKERMAN et Lada ADAMIC. « Community Net Simulator : Using simulations to study online community networks ». In : *Proc. of International Conference on Communities and Technologies*. Springer, 2007, p. 295-321.
- [21] Jun ZHANG, Mark S ACKERMAN et Lada ADAMIC. « Expertise networks in online communities : structure and algorithms ». In : *Proc. of the International Conference on World Wide Web*. 2007, p. 221-230.

À quoi sert la spécialisation en évolution culturelle de la connaissance ?

Andreas Kalaitzakis^a Jérôme Euzenat^a
andreas.kalaitzakis@inria.fr jerome.euzenat@inria.fr

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

Résumé

Des agents peuvent faire évoluer leurs ontologies en accomplissant conjointement une tâche. Nous considérons un ensemble de tâches dont chaque agent ne considère qu'une partie. Nous supposons que moins un agent considère de tâches, plus la précision de sa meilleure tâche sera élevée. Pour le vérifier, nous simulons différentes populations considérant un nombre de tâches croissant. De manière contre-intuitive, l'hypothèse n'est pas vérifiée. D'une part, lorsque les agents ont une mémoire illimitée, plus un agent considère de tâches, plus il est précis. D'autre part, lorsque les agents ont une mémoire limitée, les objectifs de maximiser la précision de leur meilleures tâches et de s'accorder entre eux sont mutuellement exclusifs. Lorsque les sociétés favorisent la spécialisation, les agents n'améliorent pas leur précision. Cependant, ces agents décideront plus souvent en fonction de leurs meilleures tâches, améliorant ainsi la performance de leur société.

Mots-clés : Évolution culturelle, Spécialisation

Abstract

Agents can evolve their ontologies by jointly performing a task. We consider a set of tasks of which each agent considers only a part. We assume that the fewer tasks an agent considers, the higher the accuracy of its best task. To verify this, we simulate different populations considering an increasing number of tasks. Counter-intuitively, the assumption is not verified. On the one hand, when agents have an unlimited memory, the more tasks an agent considers, the more accurate it is. On the other hand, when agents have limited memory, the goals of maximizing the accuracy of their best tasks and agreeing with each other are mutually exclusive. When societies favor specialization, agents do not improve their accuracy. However, these agents will decide more often based on their best tasks, thus improving the performance of their society.

Keywords: Cultural evolution, specialization

1 Introduction

En communiquant entre eux, des agents font évoluer leurs représentations des connaissances, un phénomène qui se manifeste par un changement de comportement. Ce phénomène est étudié dans le cadre de l'évolution culturelle des connaissances, où les agents acquièrent et font évoluer leurs connaissances, en fonction de leur perception et du retour qu'ils reçoivent des autres agents. Ici, nous modélisons des agents qui représentent leurs connaissances en utilisant des ontologies. Jusqu'à présent, la plupart des travaux sur l'évolution culturelle des connaissances se concentrent sur des agents qui (1) font évoluer leurs connaissances dans des environnements mono-tâche et (2), disposent de ressources illimitées. [15] propose un jeu d'alignement itératif, permettant aux agents de classer progressivement les nombres dans des ensembles identiques. Ainsi, à la fin de l'expérience, les agents partagent la même représentation des connaissances. Dans [2], les agents interagissent sur une seule tâche de décision. Après un nombre fini d'interactions, les agents se mettent d'accord sur toutes les décisions, mais pas nécessairement sur la même base.

Cependant, notre monde n'est pas monolithique et les ressources des agents ne sont pas illimitées. Les agents doivent être capables d'utiliser des ressources limitées afin d'effectuer plusieurs tâches de manière efficace. Pourtant, on sait peu de choses sur la façon dont la réalisation de plusieurs tâches affecte l'évolution culturelle des connaissances. On en sait encore moins sur les agents encouragés à se spécialiser dans des environnements multitâches. Peuvent-ils se spécialiser tout en restant en accord les uns avec les autres ? Les sociétés d'agents bénéficient-elles de la spécialisation des agents ? Nous abordons ces questions en étendant le cadre introduit dans [2]. Dans ce cadre, les agents effectuent la même tâche consistant à prendre une décision abstraite dans un domaine abstrait. Dans notre cadre, les agents sont affectés à une ou plusieurs tâches et

sont donc capables de prendre des décisions dans plusieurs domaines abstraits. Notre hypothèse principale est que les agents qui entreprennent moins de tâches atteignent une précision plus élevée sur leurs meilleures tâches, que les agents qui entreprennent toutes les tâches. Nous supposons qu'en interagissant sur un ensemble limité de tâches, les agents seront plus précis dans certaines tâches que dans d'autres.

Nous organisons notre travail en deux expériences. Dans la première expérience [8], des agents disposant d'une mémoire illimitée entreprennent un ensemble limité de tâches, que nous appellerons par la suite le scope. Dans notre travail, la prise de décisions pour chaque tâche repose sur différentes propriétés. Par exemple, décider de prendre un parapluie peut reposer sur la propriété humidité alors que décider de porter un t-shirt peut reposer sur la propriété température. Dans la deuxième expérience [9], les agents entreprennent les mêmes tâches, tout en faisant face à des limitations de mémoire. À cette fin, nous fixons un nombre maximum de classes à maintenir dans l'ontologie d'un agent. Lorsque les agents atteignent ces limites, ils essaient d'oublier des connaissances qui ne sont pas pertinentes pour leur scope et rendent ainsi l'espace à nouveau disponible. Nous évaluons les ontologies des agents par leur contribution à (1) la promotion de l'accord entre les agents et (2) la prise de décisions correctes pour différentes tâches.

Les entités qui constituent l'environnement, ainsi que la notation qui le décrit, sont présentés dans le §3. La section 4 présente les grandes lignes de l'expérience. La section 5 présente un ensemble d'hypothèses ainsi que le protocole utilisé pour les tester. Les résultats des tests sont présentés dans la section 6.

2 État de l'art

Il a été montré que les jeux référentiels [11], facilitent l'établissement de protocoles de communication entre des agents communicants qui tentent d'obtenir une rétribution partagée. [13] soutient qu'un protocole de communication émerge lorsque les agents tentent de minimiser la complexité computationnelle de l'interprétation sémantique. [7] et [3] présentent différents cadres permettant l'émergence d'un langage partagé. [7] étudie un cadre où deux agents développent un langage dans le but de réussir dans un jeu référentiel. [3] propose un cadre permettant la transmission intra-générationnelle, en se concentrant sur la compositionnalité de la langue émergente.

[6] montre que la transmission culturelle implicite conduit à une plus grande compositionnalité de la langue. Notre travail repose également sur les agents qui essaient de se mettre d'accord entre eux. Cependant, les travaux présentés ici se concentrent sur les caractéristiques du protocole de communication émergé. Bien que des caractéristiques similaires soient également présentes dans notre travail, nous ne nous concentrons pas sur le protocole de communication mais plutôt sur l'impact de l'accord des agents sur la réalisation de plusieurs tâches.

Il a été montré que l'apprentissage multitâche améliore considérablement la classification dans différents domaines, par exemple, la robustesse des adversaires [12], la similarité visuelle interconcept [4], l'apprentissage du phénotype [5]. Les agents ont également été utilisés pour étudier l'impact de l'apprentissage multitâche sur les protocoles de communication émergents. Dans [14], l'apprentissage par renforcement multi-agent coopératif est considéré. La littérature présentée dans ce paragraphe est liée à notre travail car elle considère des agents qui effectuent plusieurs tâches. Cependant, tout changement dans le comportement et les connaissances de ces agents n'est pas le résultat d'une évolution culturelle.

L'évolution culturelle des connaissances a été étudiée dans [2] et [15]. Dans [15], les auteurs cherchent à comprendre comment les concepts sont organisés et comment un comportement collectif peut être établi de manière autonome. Ils montrent qu'il existe un large éventail de conditions sous lesquelles un consensus collectif se produit. Dans [2], une expérience en deux étapes est utilisée où les agents apprennent d'abord un classificateur et interagissent ensuite par paires. Grâce à un mécanisme d'adaptation, il est montré que les agents parviennent à une meilleure connaissance, sans nécessairement aboutir à des ontologies identiques. Nous nous différencions de ceux-ci en introduisant des agents qui font évoluer leurs connaissances par rapport à des tâches multiples.

Enfin, il est à souligner qu'aucun des travaux présentés dans cette section n'impose de limites aux agents.

3 Cadre expérimental

Nous étendons [2] en introduisant des agents qui entreprennent plusieurs tâches. Ici nous fournissons des définitions préliminaires concernant les

agents, les différentes tâches, les objets, ainsi que leur environnement.

3.1 Environnement

Les agents évoluent dans un environnement peuplé d'objets décrits par un ensemble \mathcal{P} de propriétés booléennes. Les objets sont donc décrits par la présence ou l'absence d'une propriété $p \in \mathcal{P}$, dénotée par p et $\neg p$ respectivement. Il n'existe donc que $2^{|\mathcal{P}|}$ types d'objets, qui sont rassemblés dans un ensemble \mathcal{I} .

3.2 Tâches

Le terme tâche fait référence à un travail effectué par un agent. Ici, nous nous concentrerons sur un ensemble de tâches de décision : prendre une décision abstraite dans un domaine abstrait, c'est-à-dire choisir parmi $d_1, d_2 \dots d_n$. Il peut y avoir différentes tâches $t \in \mathcal{T}$ associées à un ensemble différent de décisions possibles \mathcal{D}_t . Dans ce contexte, chaque objet o peut être considéré par rapport à toute tâche $t \in \mathcal{T}$. Une fonction $h^*(o, t) \rightarrow \mathcal{D}_t$ fournit la décision correcte, inconnue des agents, pour un objet o par rapport à une tâche t .

3.3 Agents

Les agents sont des entités autonomes, coexistantes, capables de percevoir et de distinguer des objets en fonction de leurs propriétés. Dans ce contexte, une société d'agents multitâches \mathcal{A} , réalise de multiples tâches. À cette fin, les agents construisent et font évoluer des connaissances sous la forme d'ontologies, privées à chaque agent, exprimées en \mathcal{ALC} [1]. Chaque agent α utilise ses connaissances pour calculer une fonction $h^\alpha(o, t) \rightarrow \mathcal{D}_t$ qui, étant donné un objet o et une tâche t , fournit une décision $h^\alpha(o, t)$. La figure 1 montre un exemple de connaissance multi-tâche construite par un agent α . La partie inférieure représente l'ontologie privée \mathcal{O}^α de l'agent α , lui permettant de classifier les objets de l'environnement. La partie supérieure présente deux ontologies de décision, chacune contenant les décisions valides pour les tâches t_1 et t_2 . Étant donné qu'un agent α apprend au plus une décision pour un objet o et une tâche t , chaque feuille de \mathcal{O}^α ne peut être alignée plus d'une fois avec la même ontologie de décision.

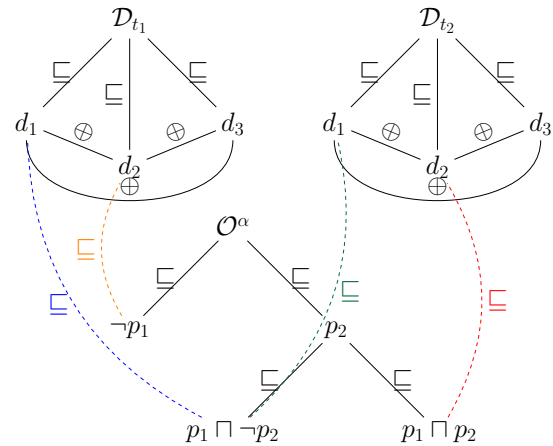


FIGURE (1) Exemple d'une ontologie multitâche pour un agent α . Chaque couleur représente une décision différente. Le symbole \oplus indique que deux décisions sont disjointes.

4 Aperçu de l'expérience

L'expérience est initialisée par une phase d'apprentissage, à la fin de laquelle chaque agent a appris une ontologie privée. Une fois leurs ontologies apprises, les agents passent par un nombre fixe d'interactions. En fonction du résultat d'une interaction, un agent peut adapter son ontologie. La figure 2 illustre le déroulement de l'expérience du point de vue de l'agent α . De plus amples détails sur la façon dont les agents apprennent, définissent leur scope, interagissent, oublient et adaptent leurs ontologies sont présentés dans les sous-sections 4.1, 4.2, 4.3, 4.4 et 4.5 respectivement.

4.1 Apprentissage initial

Nous abordons l'apprentissage multitâche comme un problème d'induction d'une ontologie capable de fournir une décision pour toute tâche $t \in \mathcal{T}$. L'induction d'une telle ontologie peut être vue dans la figure 3. Nous observons qu'il est possible pour un agent d'être capable de classifier un objet mais d'être incapable de prendre une décision concernant une ou plusieurs tâches. Par exemple, étant donné l'ontologie de la figure 1 et un objet décrit par $p_1 \sqcap p_2$, il est impossible de prendre une décision par rapport à la tâche t_1 .

4.2 Définition du scope

Nous implémentons ici des agents qui accomplissent différents sous-ensembles de tâches, en donnant la priorité à certaines tâches par rap-

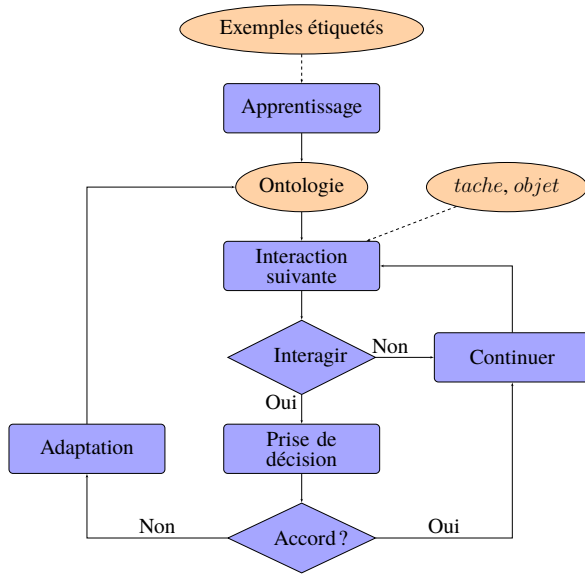


FIGURE (2) Schéma de l'expérience. Les éléments en couleur bisque constituent des entrées pour différentes activités (e.g., les exemples étiquetés). Les éléments en violet constituent soit des activités (rectangles), soit des décisions (diamants).

port à d'autres. À cette fin, nous introduisons la fonction $prior^\alpha(t) \rightarrow \mathcal{N}$, qui renvoie un entier unique attribué à la tâche t par un agent α . Cet entier va de 1 à $|\mathcal{T}|$ et représente la priorité de la tâche t parmi toutes les tâches existantes. Munis de leurs classements de priorité, les agents définissent l'ensemble des tâches qu'ils entreprennent. Afin de décider si une tâche doit être incluse dans leur scope, les agents comparent la priorité de chaque tâche t avec le rang de priorité maximum accepté m . Si $prior^\alpha(t) \leq m$, alors la tâche est dans le scope de l'agent. Si $prior^\alpha(t) > m$, alors la tâche est en dehors du scope de l'agent.

4.3 Interaction

À chaque interaction, deux agents α et β sélectionnés aléatoirement se voient présenter un objet o et une tâche t . Les agents vont alors fournir des décisions basées sur leurs fonctions respectives $h^\alpha(o, t)$ et $h^\beta(o, t)$. Si un des agents n'est pas en mesure de fournir une décision, alors cet agent en choisit une au hasard. Les agents révèlent alors leurs décisions les uns aux autres. Si $h^\alpha(o, t) = h^\beta(o, t)$, les agents sont d'accord et leur interaction se termine avec succès. Dans le cas contraire, on considère qu'il s'agit d'un échec et l'un des deux agents adaptera ses connaissances.

4.4 Limitations et oubli des connaissances

Nos agents sont limités en termes de nombre maximum de classes présentes dans leurs ontologies. Lorsque cette limitation est atteinte, ils essaient d'oublier certaines parties de leurs connaissances afin de libérer de l'espace mémoire en faveur des tâches qu'ils priorisent. Nous supposons que si elle est suffisamment restreinte, une ontologie ne pourra contenir que les propriétés nécessaires pour être précis sur une seule tâche. Par exemple, décidons si l'agent doit prendre un parapluie (d_1) ou non (d_2), en s'appuyant sur la propriété *humidité*. Si l'ontologie de l'agent ne contient pas la propriété *humidité*, alors la probabilité qu'une décision associée soit correcte est de 50%. Nous implémentons le processus d'oubli comme suit. Nous supprimons les nœuds parents qui satisfont aux critères suivants : (a) leurs descendants sont des nœuds feuilles et (b) leurs descendants sont associés à la même décision concernant toutes les tâches dans le scope de l'agent. Le processus est répété de manière récursive, tant que des nœuds parents satisfaisant aux critères (a) et (b) existent.

4.5 Adaptation

Nous étendons le mécanisme d'adaptation présenté dans [2], en introduisant l'adaptation en présence de plusieurs tâches (figure 5). Un agent peut soit remplacer une décision existante, soit diviser une classe en 2 sous-classes. L'agent le fait sur la base d'une propriété qui permet de distinguer l'objet actuel des objets classés par la classe à diviser. Seules les décisions concernant la tâche en cours sont affectées.

Algorithm 1 Adaptation d'ontologie

- 1: α demande à β la définition de la classe classant l'objet o .
- 2: β répond avec \mathcal{C}^β .
- 3: **if** $\mathcal{C}^\alpha \not\sqsubseteq \mathcal{C}^\beta$ **then**
- 4: α demande à β sa décision $h^\beta(o, t)$.
- 5: β répond avec $h^\beta(o, t)$.
- 6: une propriété $p : p \in \mathcal{C}^\beta, p \notin \mathcal{C}^\alpha$ est sélectionnée.
- 7: α divise \mathcal{C}^α en $\mathcal{C}_1^\alpha \equiv \mathcal{C}^\alpha \sqcap \neg p$ et $\mathcal{C}_2^\alpha \equiv \mathcal{C}^\alpha \sqcap p$.
- 8: α associe \mathcal{C}_1^α à toutes les décisions précédemment associées à \mathcal{C}^α .
- 9: **if** Aucune décision concernant t n'est associée à \mathcal{C}^α . **then**
- 10: α associe \mathcal{C}^α à la décision $h^\beta(o, t)$.
- 11: α associe \mathcal{C}_2^α à la décision $h^\beta(o, t)$.
- 12: **else**
- 13: $\mathcal{C}^\alpha = \mathcal{C}^\alpha \sqcap p$
- 14: Associer \mathcal{C}^α à toutes les décisions précédemment associées à \mathcal{C}^β .

À quoi sert la spécialisation en évolution culturelle de la connaissance ?

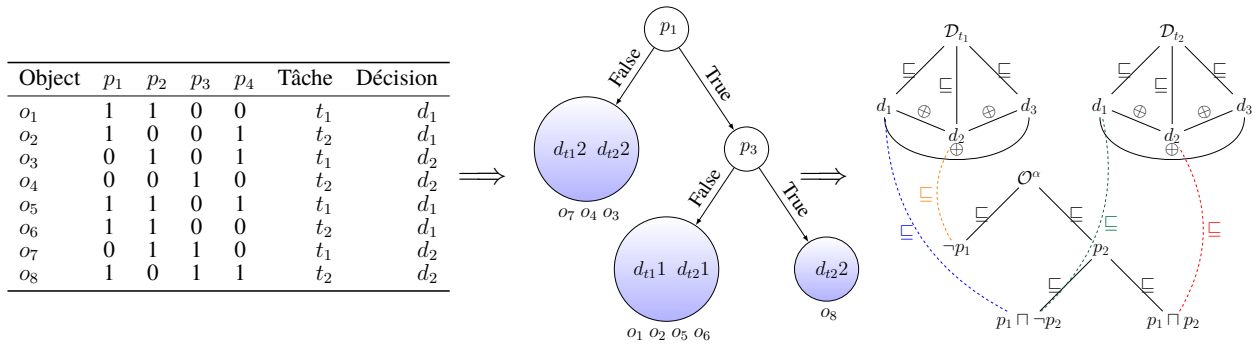


FIGURE (3) Vue d'ensemble de la méthode d'apprentissage proposée. Les agents commencent par induire un arbre de décision basé sur un ensemble d'exemples étiquetés. Cet arbre de décision est ensuite converti en une ontologie.

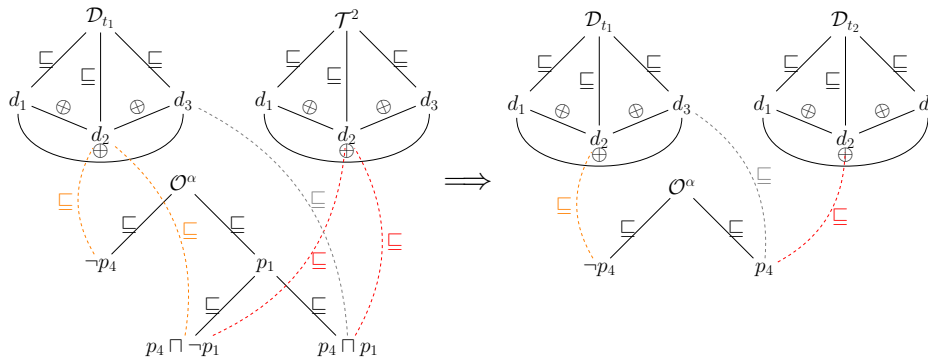


FIGURE (4) Exemple de libération de mémoire par généralisation lors d'une interaction par rapport à la tâche t_2 . Soit la tâche t_2 reposant sur l'ensemble de propriétés \mathcal{P}_{t_2} . La propriété $p_1 \notin \mathcal{P}_{t_2}$, donc p_1 ne permet pas de distinguer différentes décisions pour la tâche t_2 . Dans cet exemple, l'agent a associé la même décision (en rouge), à la fois à $p_4 \sqcap \neg p_1$ et à $p_4 \sqcap p_1$. Ces deux classes peuvent être fusionnées sans aucune perte de connaissance par rapport à t_2 , comme on le voit à droite.

5 Protocole

5.1 Hypothèses

Nous testons les hypothèses suivantes :

- **Hypothèse 1** Les agents convergent sur leurs décisions après un nombre fini d'interactions.
- **Hypothèse 2** Les agents qui entreprennent moins de tâches atteindront une précision plus élevée sur leur meilleurs tâches, que les agents qui entreprennent toutes les tâches.

5.2 Paramètres

Nous réalisons deux expériences. Dans l'expérience 1, les 18 agents ne sont soumis à aucune limitation de mémoire. Dans l'expérience 2, les ontologies des 18 agents sont limitées à un maximum de 4 classes feuilles. Chaque expérience est exécutée sous 3 configurations, correspondant à des agents qui entreprennent 1 à 3 tâches respectivement. Chaque configuration est exécutée 20 fois et la moyenne de ses résultats est calculée.

Une exécution consiste en 80000 interactions, chaque interaction ayant lieu entre deux agents choisis au hasard parmi 18 agents. Leur environnement contient 64 types d'objets différents, chacun étant perceptible à travers 6 propriétés. Les agents sont initialement formés à toutes les tâches $|\mathcal{T}| = \{3\}$, chaque tâche faisant appel à 2 des 6 propriétés. L'induction de l'ontologie initiale est basée sur un échantillon aléatoire de 10 % de tous les exemples étiquetés existants. Pour chaque tâche, il existe 4 décisions différentes. L'évaluation du score entre deux interactions consécutives se base sur les 60 % de tous les exemples étiquetés existants.

5.3 Mesures

Le taux de réussite (*srate*), tel qu'introduit dans [2] évalue l'interopérabilité entre les agents. Il est défini comme la proportion d'interactions réussies, sur toutes les interactions réalisées jusqu'à la $n^{\text{ème}}$ interaction.

Le taux de décisions correctes (*cdrate*) évalue la performance d'une société d'agents. Il est défini comme la proportion de décisions correctes

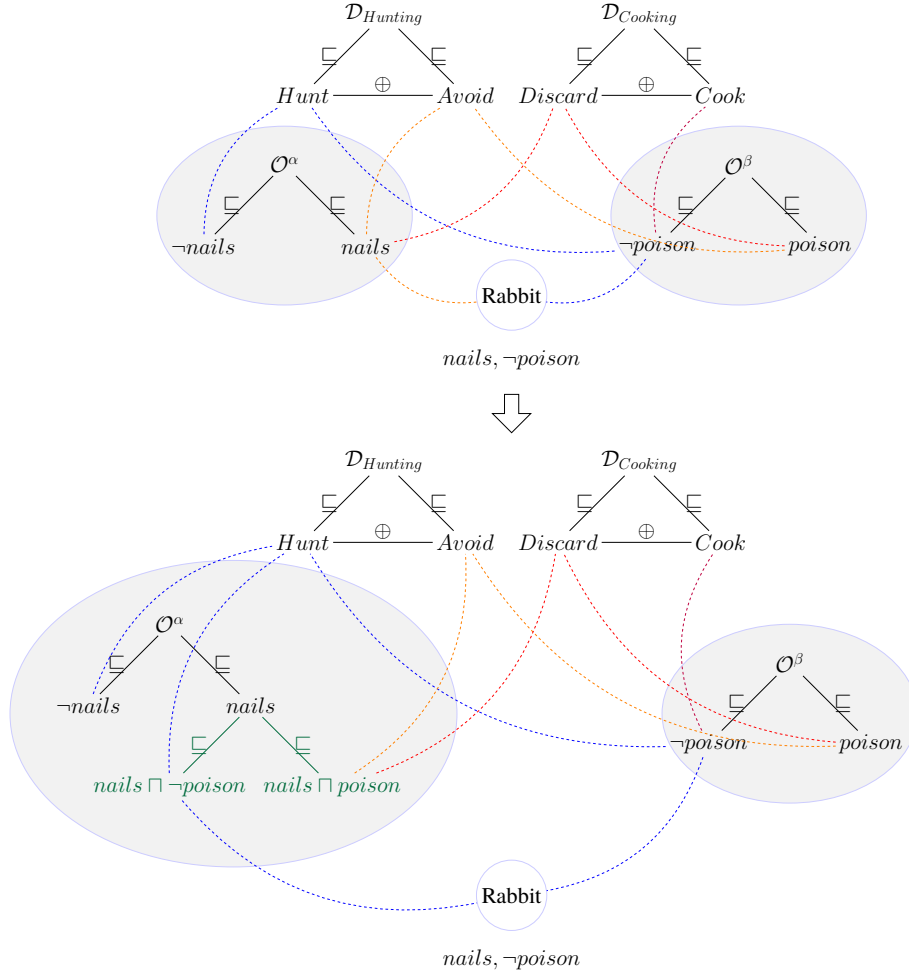


FIGURE (5) Suite à un échec de communication, l'agent α divise la classe *nails* en 2 sous-classes en effectuant une division supplémentaire sur la propriété *poison*. La 1ère sous-classe, $nails \sqcap \neg poison$ sera associée à la même décision que celle de l'agent β (*Hunt*). La 2ème décision sera associée à la décision précédemment associée à la classe *nails* (*Avoid*).

prises par une société d'agents, sur l'ensemble des interactions réalisées jusqu'à la $n^{\text{ème}}$ interaction. Lorsque les agents sont d'accord, cette décision est celle prise par les deux agents. Lorsque les agents sont en désaccord, la décision prise en compte est celle de l'agent ayant le score le plus élevé parmi les deux agents en interaction.

La précision de la tâche (*tacc*) évalue la qualité de la représentation des connaissances d'un agent. Elle adapte la mesure de précision introduite dans [2] à différentes tâches. Elle est définie comme la proportion de types d'objets pour lesquels une décision correcte serait prise par rapport à une tâche t , par un agent α à la n^{e} itération de l'expérience.

$$tacc(\alpha, n, t) = \frac{|\{o \in \mathcal{I} : h_n^\alpha(o, t) = h^*(o, t)\}|}{|\mathcal{I}|}$$

tacc est utilisé pour mesurer la précision moyenne sur la meilleure tâche des agents, ainsi

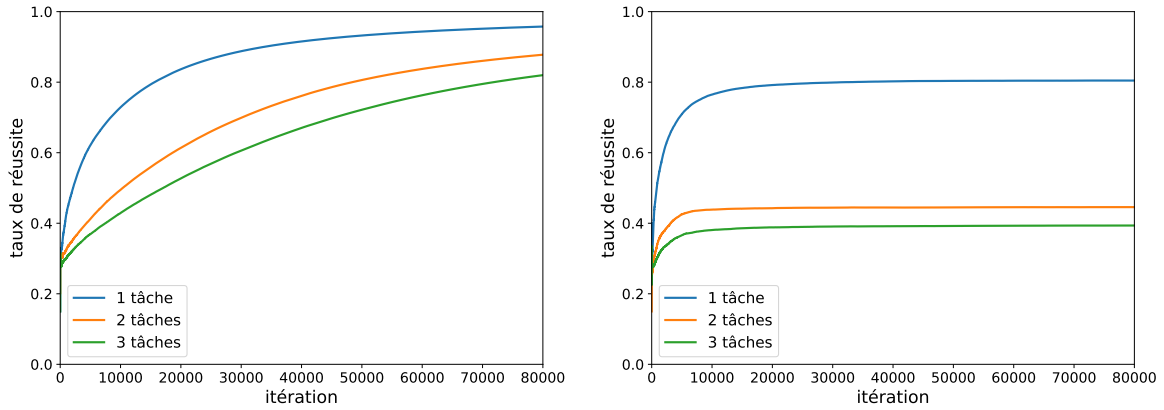
que leur précision moyenne sur toutes les tâches existantes ou entreprises.

6 Résultats et discussion

Nous présentons et discutons ici les résultats obtenus dans le cadre des deux expériences décrites précédemment. Chacune des figures 6, 7 et 9 est divisée en deux sous-figures. La sous-figure (a) correspond aux résultats acquis lors de l'expérience 1, tandis que la sous-figure (b) correspond aux résultats acquis lors de l'expérience 2. Puisque les agents qui ne font face à aucune limitation ne bénéficient pas de la spécialisation, les autres figures concernent uniquement les résultats de l'expérience 2.

Interopérabilité inter-agent. La figure 6 montre l'évolution du taux de réussite moyen pour différentes tailles du scope. La figure 6a

À quoi sert la spécialisation en évolution culturelle de la connaissance ?



(a) Agents avec une mémoire illimitée

(b) Agents dont la mémoire est limitée à 4 classes

FIGURE (6) Évolution du taux de réussite moyen pour différentes tailles du scope des agents.

montre qu'une population d'agents en interaction apprendra à se mettre d'accord sur toutes leurs décisions, indépendamment de la taille de leur scope. Ceci est justifié par le fait que les agents représentés ici ne sont confrontés à aucune limitation. Par conséquent, ils adopteront progressivement les mêmes propriétés dans leurs ontologies, même si leurs classes feuilles ne coïncident pas. Nos observations confirment les résultats qui ont été présentés précédemment dans [2]. De plus, elles indiquent que la taille du scope a un impact sur le taux de réussite obtenu. Plus la taille du scope est grande, plus il faut d'interactions pour se mettre d'accord sur tout, donc plus le taux de réussite moyen à la convergence est faible. La figure 6b montre que le taux de réussite se stabilise, sans toutefois converger à 1. Cela indique que soit les agents continuent à adapter leurs ontologies, soit leurs ontologies finales ne leur permettent pas de s'accorder sur toutes les décisions. Dans la section 4.4, nous avons supposé que si elle est suffisamment restreinte, une ontologie sera capable de contenir uniquement les propriétés requises pour être précis sur une seule tâche. Par conséquent, il faut s'attendre à ce que les agents interagissant sur un nombre de tâches nécessitant plus de mémoire, ne pourront pas se mettre d'accord sur toutes les décisions. Il est montré que cela est vrai même pour les agents qui interagissent sur une seule tâche. Nous ne pouvons donc soutenir l'hypothèse 1 que pour les agents qui ne sont confrontés à aucune limitation. Cependant, nous observons que plus le scope des agents est réduit, plus leur taux de réussite est élevé. En raison de l'algorithme d'induction de l'ontologie initiale, l'ontologie d'un agent qui entreprend la tâche t

peut contenir des propriétés qui n'appartiennent pas à \mathcal{P}_t . Rien ne garantit que les agents parviendront à remplacer ces propriétés par des propriétés appartenant à \mathcal{P}_t . Deux cas distincts peuvent être envisagés. Dans le premier cas, un agent éliminera progressivement toutes les propriétés qui ne sont pas liées à la tâche qu'il entreprend. Cet agent va potentiellement apprendre une ontologie qui lui permettra de prendre des décisions correctes par rapport à tous les types d'objets rencontrés. Dans le second cas, un agent ne parviendra pas à éliminer toutes les propriétés qui ne sont pas liées à la tâche qu'il entreprend. Les agents qui relèvent du deuxième cas remplaceront de manière répétée les propriétés qui appartiennent à \mathcal{P}_t par des propriétés différentes qui appartiennent également à \mathcal{P}_t . Par conséquent, ces agents sont capables de prendre des décisions correctes pour différents sous-ensembles de types d'objets existants à un moment donné.

Précision des ontologies. La figure 7 représente l'évolution de la précision moyenne pour différentes tailles du scope des agents. Dans la section 1, nous supposons que les agents interagissant sur un scope limité de tâches seront plus précis sur certaines tâches que sur d'autres, au détriment de leur précision moyenne. Cependant, nous ne l'observons dans aucune des deux sous-figures de la figure 7. 7a montre que la réalisation de tâches supplémentaires améliore significativement la précision moyenne des agents. 7b montre que la taille du scope des agents a une incidence minimale sur leur précision moyenne. Une fois de plus, cela se justifie par les limitations appliquées. Deux cas peuvent être distingués. Dans le premier cas, un agent devient très

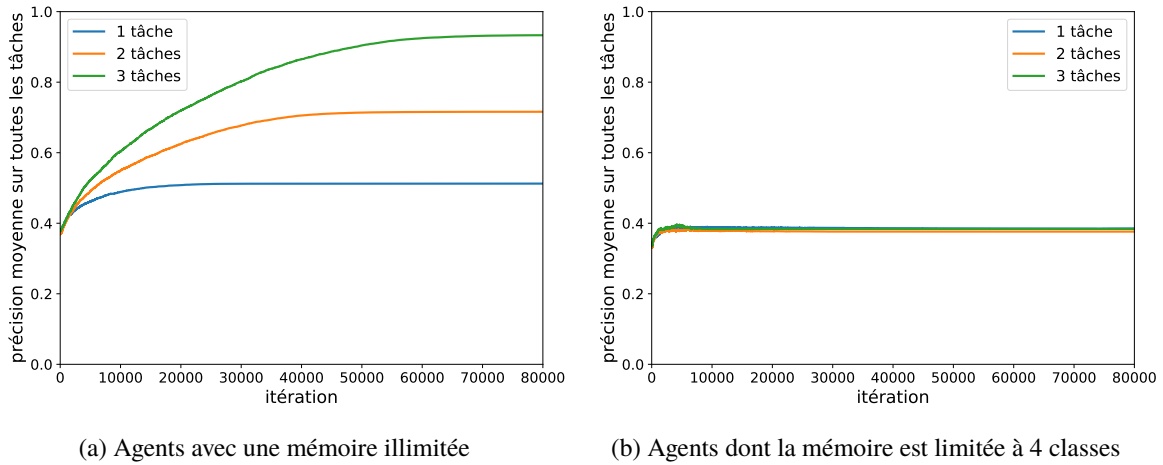


FIGURE (7) Évolution de la précision moyenne pour différentes tailles du scope des agents.

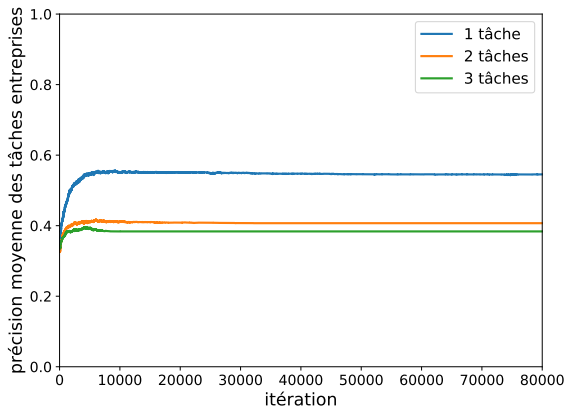


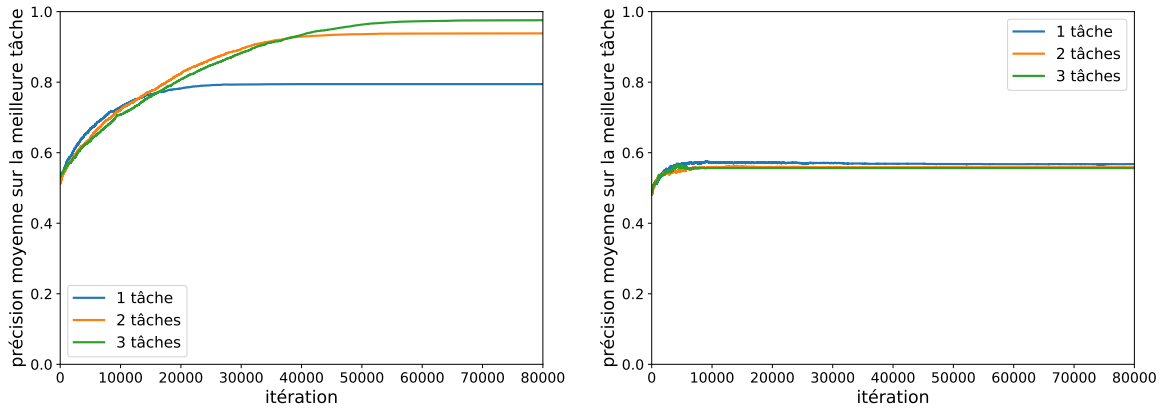
FIGURE (8) Évolution de la précision moyenne des tâches entreprises pour différentes tailles du scope des agents.

précis dans une tâche et nettement moins précis dans les autres. Dans le second cas, un agent apprend une ontologie lui permettant de devenir moyennement précis sur plusieurs tâches. Cela se traduit par des agents qui démontrent des précisions moyennes qui sont statistiquement indiscernables les unes des autres, indépendamment de la taille de leur scope.

La figure 8 montre l'évolution de la précision moyenne sur les tâches que les agents entreprennent pour différentes tailles de leur scope. Elle montre que plus le scope des agents est petit, plus leur précision moyenne sur les tâches effectuées est élevée. Ce résultat est attendu pour la raison suivante. Un agent dans la configuration examinée peut devenir très précis sur une tâche au maximum. Cependant, plus le scope des agents est grand, plus le nombre de tâches sur la

base desquelles la précision moyenne des tâches effectuées est calculée est élevée. Par conséquent, les agents qui s'attaquent à une seule tâche font preuve d'une précision moyenne sur les tâches effectuées supérieure. La figure 9 dépeint l'évolution de la précision sur la meilleure tâche pour différentes tailles du scope des agents. En examinant 9a, deux observations peuvent être tirées. La première observation est que la précision moyenne des agents à mémoire illimitée (figure 7a) est toujours inférieure à la précision moyenne sur la meilleure tâche des mêmes agents montrés ici. Ces agents sont donc capables de se spécialiser en restreignant le scope de leurs tâches. La deuxième observation est que les agents qui s'attaquent à moins de tâches, ont une précision moyenne sur leur meilleure tâche plus faible que les agents qui s'attaquent à toutes les tâches. Plus précisément, plus le scope des agents est restreint, plus cette précision est faible. Ainsi, nous pouvons conclure que la spécialisation des agents disposant d'une mémoire illimitée n'apporte aucun avantage en termes de précision. Dans la configuration examinée, chaque configuration dépend de propriétés différentes. Par conséquent, notre observation n'est pas liée à la transférabilité des connaissances d'une tâche à l'autre, puisque l'apprentissage de la décision par rapport à une tâche n'est pas lié à l'apprentissage de la décision pour une autre tâche. Au contraire, elle est justifiée par le fait que les agents s'attaquant à toutes les tâches construisent des ontologies plus complètes et par conséquent associent les décisions apprises à une classification plus détaillée. La figure 9b montre que les agents à mémoire limitée se spécialisent également. Cependant, cette spécialisation n'est pas

À quoi sert la spécialisation en évolution culturelle de la connaissance ?



(a) Agents avec une mémoire illimitée

(b) Agents dont la mémoire est limitée à 4 classes

FIGURE (9) Précision moyenne sur la meilleure tâche en fonction du scope des agents.

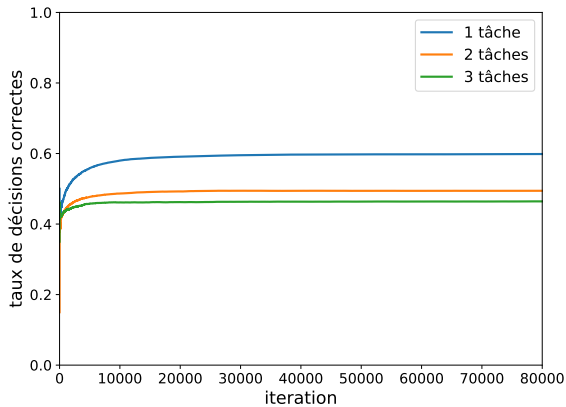


FIGURE (10) Évolution du taux de décision correcte pour différentes tailles du scope des agents.

liée à la taille de leur scope, mais plutôt aux limites de la mémoire. Les agents deviendront très précis au maximum sur une tâche, qu'ils s'attaquent à une ou plusieurs tâches. En d'autres termes, le fait que les agents puissent s'abstenir de toute interaction, sauf celles qui concernent la seule tâche qu'ils entreprennent, ne leur permet pas d'améliorer leur précision sur leur meilleure tâche. Nous avons vérifié que cela est également vrai pour des capacités de mémoire plus élevées (8 et 12 classes respectivement). L'hypothèse 2 est donc soutenue uniquement pour des agents qui ne sont confrontés à aucune limitation.

La figure 10 représente l'évolution du taux moyen de décisions correctes pour différentes tailles du scope des agents. Jusqu'à présent, les résultats montrent que les agents confrontés à des limitations de mémoire (1) spécialisent leurs

connaissances indépendamment de la taille de leur scope et (2) leur précision moyenne ne dépend pas du nombre de tâches qu'ils entreprennent. La figure 10 nous permet d'examiner si les sociétés d'agents bénéficient d'un scope réduit. Les résultats montrent que plus ce scope est réduit, plus le taux de décisions correctes pour une société est élevé. En d'autres termes, si la modification de la taille du scope des agents ne les rend ni moins, ni plus spécialisés, elle permet aux agents de prendre des décisions pour les tâches qu'ils maîtrisent mieux.

Analyse statistique. Notre analyse de variance (*one-way ANOVA*) montre que la taille du scope a un impact statistiquement significatif ($p \leq 0.01$) sur (1) le taux de réussite, (2) la précision moyenne sur les tâches effectuées et (3) le taux moyen de décisions correctes. La taille du scope n'a pas d'impact statistiquement significatif ($p > 0.01$) sur (1) la précision sur leur meilleure tâche et (2), la précision moyenne sur toutes les tâches existantes.

7 Conclusions

Ici nous examinons si les sociétés d'agents bénéficient de la spécialisation. Nous proposons une expérience où les agents font évoluer des ontologies en interagissant sur un ensemble limité de tâches. En exploitant ce cadre, nous montrons que les agents à mémoire limitée spécialiseront leurs connaissances quel que soit le nombre de tâches qu'ils effectuent. Cependant, nos résultats montrent qu'en assignant différentes tâches à différents agents, les sociétés améliorent leur taux de décision correcte. Jusqu'à présent, nous

n'avons examiné que les sociétés homogènes, c'est-à-dire les sociétés où tous les agents effectuent le même nombre de tâches. Ce travail peut être un tremplin vers l'exploration de sociétés d'agents plus complexes. Par exemple, il serait intéressant d'examiner des sociétés d'agents où des agents multitâches coexistent avec des agents qui se concentrent sur des tâches spécifiques.

Remerciements

Ce travail a été partiellement supporté par la chaire MIAI "Knowledge communication and evolution" (ANR-19-P3IA-0003).

Références

- [1] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [2] Yasser Bourahla, Manuel Atencia, and Jérôme Euzenat. Knowledge improvement and diversity under interaction-driven adaptation of learned ontologies. In *Proc. 20th ACM international conference on Autonomous Agents and Multi-Agent Systems (AAMAS), London, United Kingdom*, pages 242–250, 2021.
- [3] Michael Cogswell, Jiasen Lu, Stefan Lee, Devi Parikh, and Dhruv Batra. Emergence of compositional language with deep generational transmission. *CoRR*, abs/1904.09067, 2019. arXiv:1904.09067.
- [4] Jianping Fan, Yuli Gao, and Hangzai Luo. Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation. *IEEE Transactions on Image Processing*, 17(3):407–426, 2008.
- [5] Mohamed Ghalwash, Zijun Yao, Prithwish Chakraporty, James Codella, and Daby Sow. Phenotypical ontology driven framework for multi-task learning. In *Proceedings of the Conference on Health, Inference, and Learning, CHIL '21*, page 183–192, New York, USA, 2021.
- [6] Laura Harding Graesser, Kyunghyun Cho, and Douwe Kiela. Emergent linguistic phenomena in multi-agent communication games. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3700–3710, Hong Kong, China, 2019.
- [7] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequence of symbols. In *5th International Conference on Learning Representations (ICLR 17, workshop track)*, Toulon, France, 2017.
- [8] Andreas Kalaitzakis. 20230110-MTOA experiment description, 2023. URL : <https://sake.re/20230110-MTOA>.
- [9] Andreas Kalaitzakis. 20230120-MTOA experiment description, 2023. URL : <https://sake.re/20230120-MTOA>.
- [10] Lazy lavender. 2020. URL : <https://gitlab.inria.fr/moex/lazylav>.
- [11] David Lewis. Convention : A philosophical study. *Synthese*, 26(1):153–157, 1969.
- [12] Chengzhi Mao, Amogh Gupta, Vikram Nitin, Baishakhi Ray, Shuran Song, Junfeng Yang, and Carl Vondrick. Multitask learning strengthens adversarial robustness. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 158–174, Cham, 2020.
- [13] Luc Steels. What triggers the emergence of grammar? In *AISB'05 : Proceedings of the Second International Symposium on the Emergence and Evolution of Linguistic Communication (EELC'05)*, pages 143–150, Hatfield, United Kingdom, 2005.
- [14] Jonathan Thomas, Raul Santos-Rodriguez, Mihai Anca, and Robert Piechocki. Multilingual agents through multi-headed neural networks. volume 4, Tromsø, Norway, 2023.
- [15] Jun Wang and Les Gasser. Kmutual online ontology alignment. In *Proc. 1st ACM international conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Bologna, Italy*, 2002.

Modèles et outils multi-agents

COBAI : un modèle générique à base d'agents centré sur les contextes et les interactions pour la simulation de comportements

Maëlle Beuret

maelle.beuret@u-bourgogne.fr

Christian Gentil

christian.gentil@u-bourgogne.fr

Irène Foucherot

irene.foucherot@u-bourgogne.fr

Joël Savelli

joel.savelli@u-bourgogne.fr

Laboratoire d'Informatique de Bourgogne, Université de Bourgogne, 21000 Dijon, France

Résumé

Cet article présente un modèle générique à base d'agents pour la simulation de comportements. Le modèle COBAI (Context-Based Agent Interactions) est basé sur un modèle existant dont nous avons conservé les principes fondamentaux : des contextes donnent des comportements aux agents ; les agents peuvent être influencés par plusieurs contextes et choisissent les comportements à adopter en fonction de leurs attributs de personnage. Ce mécanisme permet de contrôler le réalisme à la fois aux niveaux individuel et collectif. Nous proposons un modèle plus complet avec une nouvelle architecture permettant l'exécution de plusieurs comportements simultanés issus d'une combinaison de contextes. Nous introduisons des notions telles que les ressources, outils, modalités et comportements incomplets et nous définissons des groupes d'agents avec distribution de tâches. Nous appliquons le modèle à une simulation de situation de crise développée sur Unity.

Mots-clés : Architecture d'agent, architecture de comportements, simulation multi-agents, interactions d'agents

Abstract

This paper presents a generic agent-based model for human behaviors in simulations. COBAI (Context-Based Agent Interactions) is based on a previous model of which we kept the fundamental principles : contexts give behaviors for agents to adopt. Agents can be influenced by several contexts and choose behaviors to adopt depending on their character attributes. This mechanism lets us control realism both at the individual and collective levels. We propose a more comprehensive model with a new behavior architecture, allowing the execution of several simultaneous behaviors resulting from a combination of contexts. We introduce resources, tools, modalities, and incomplete behaviors. We define

groups of agents with task distribution. We apply the model to a case study of an emergency crisis developed in Unity.

Keywords: Agent architecture, behavior architecture, multi-agent simulation, agent interactions

1 Introduction

Le contexte joue un rôle essentiel dans les comportements humains (ou autres êtres vivants) et les interactions avec l'environnement. Ainsi, il est naturel que le contexte soit devenu un sujet de recherche en intelligence artificielle. Comme les systèmes à base d'agents sont devenus populaires pour la simulation de comportements humains au niveau microscopique, des chercheurs ont exploré l'utilisation de contextes pour influencer les comportements des agents.

Dans ce cadre, Soussi et Savelli ont introduit en 2009 un modèle générique d'animation comportementale basé sur des contextes [15] et des attributs de personnage. Ce modèle permet d'obtenir une variété de comportements cohérents au niveau des individus en utilisant les attributs de personnage (réalisme individuel), tout en contrôlant les comportements collectifs par la force des contextes (réalisme collectif). Ces éléments combinés permettent d'obtenir des comportements globaux réalistes (avec un niveau de granularité dépendant des choix des concepteurs). Ces mécanismes sont prometteuses car elles permettent de représenter une grande variété de situations. Cependant, le modèle présentait certaines limites. Alors que plusieurs contextes pouvaient influencer un agent, un comportement ne pouvait résulter que d'un seul contexte. Cela empêche la prise en compte d'éléments externes, comme la présence d'autres agents, dans un comportement. Des comportements plus élaborés pourraient résulter d'une combinaison

de contextes, notamment la coopération entre agents. Cet article présente un nouveau modèle basé sur les principes fondamentaux des travaux de Soussi et Savelli et traitant ces limites. Nous introduisons une nouvelle architecture de comportements. En utilisant cette architecture, nous présentons une définition de groupes avec distribution de tâches. Dans le cadre d'un partenariat avec le Centre d'Enseignement des Soins d'Urgence de Dijon (CESU21), nous visons à nous assurer que le modèle est adapté à la simulation de situation de crise pour un jeu sérieux visant à entraîner les décideurs. Nous appliquons le modèle conceptuel à un cas d'étude du personnel de la sécurité publique (pompiers, médecins, policiers, infirmiers) et civils (victimes et badauds) dans une situation de crise.

Cet article se présente comme suit. Après avoir introduit le sujet, la deuxième section présente une brève revue des travaux connexes. La troisième section introduit notre modèle COBAI (*Context-Based Agent Interactions*) et détaille chacun de ses éléments. La quatrième section présente une application de COBAI au cas d'étude précédemment mentionné. La cinquième section commente certains concepts utilisés. La dernière section conclut cet article et présente les futurs travaux.

2 Travaux connexes

Un comportement (humain ou animal) est fortement dépendant de circonstances apportées par son environnement. En intelligence artificielle, notamment dans les systèmes à base d'agents, cela a été traduit par certains auteurs par la notion de contexte. Le contexte donne des informations à propos des circonstances d'une action pour qu'elle soit plus pertinente. Dans les systèmes à base d'agents, il est également utilisé pour améliorer l'efficacité de la sélection de comportement, comme affirmé par Turner [16], nécessitant une architecture simple pour sélectionner un comportement pertinent dans une situation donnée. Ces observations ont mené à différentes approches de modélisation de comportements d'agents basés sur les contextes.

Il existe une grande variété de représentations du contexte dans les architectures de systèmes à base d'agents. Par exemple, les schémas contextuels dans le modèle CMB (*Context-Mediated Behaviors*) [16] ou le raisonnement par contexte appliqué aux systèmes à base d'agents [3, 6, 13, 12]. Dans les modèles EASI (*Environment as Active Support of Interaction*) [14] et, plus ré-

cemment utilisé, EASS (*Environment as Active Support for Simulation*) [1, 2], les contextes regroupent les entités externes à l'agent qui interviennent dans des filtres déterminant si un agent peut exécuter une action. Il n'existe ainsi aucune définition universelle du contexte dans ce domaine. L'idée principale est d'inclure des informations contextuelles, c'est-à-dire des informations qui dépendent de variables telles que la localisation, le temps, ou des points d'intérêt dans l'environnement.

Une autre manière plus indirecte de représenter les contextes, mais similaire à notre approche, est l'affordance telle que définie par Gibson [5]. L'affordance est la capacité d'un objet ou d'un environnement à transmettre des actions potentielles qui y sont liées. Bien qu'il fut initialement introduit en 1977, ce concept est toujours utilisé dans des travaux récents [9, 8, 7]. Une vision similaire à la nôtre a été présentée dans [17], où les actions sont fortement dépendantes de la position dans l'espace.

L'idée d'utiliser l'environnement pour stocker les interactions réalisables entre les agents a été introduite par Kubera *et al.* avec le modèle IODA [11, 10].

Notre approche peut être considérée comme une représentation de l'affordance utilisant des entités que nous appelons contextes. Cependant, l'architecture de comportements ainsi que d'autres concepts clefs du modèle tels que les attributs de personnage et les ressources, diffèrent des modèles existants basés sur l'affordance. Les autres modèles basés sur la notion de contexte ont une définition de celle-ci différente de la nôtre, que nous développerons dans la section suivante.

3 Context-Based Agent Interactions

COBAI est basé sur un modèle basé sur les contextes [15], qui fait intervenir trois notions principales : les notions d'agent, de contexte et de règle de comportement. Les agents n'ont pas les moyens d'agir par eux-mêmes, ce sont les contextes qui leur donnent des comportements qu'ils pourront potentiellement adopter. Les agents sélectionnent les comportements les plus appropriés avant de les exécuter. Le rôle de ce modèle était d'obtenir un réalisme au niveau des agents individuels tout en gardant le contrôle des comportements collectifs lorsque cela s'avère nécessaire. Ce contrôle permet d'assurer que les comportements globaux sont cohérents. Cependant, le modèle présentait un pro-

blème majeur que nous avons traité dans COBAI : un comportement ne pouvait résulter de plusieurs contextes simultanés, limitant ainsi la variété et le réalisme des comportements. Il était notamment impossible de faire coopérer des agents. Dans cette section, nous introduisons la nouvelle architecture de comportements.

Nous avons représenté les concepts du modèle et leurs relations dans un diagramme sur la Figure 1. Chaque concept sera détaillé dans cette section.

3.1 Agents

Suivant les mêmes principes que [15], les agents ne peuvent agir sans être soumis à des contextes. Leur rôle est de traiter l'influence et les comportements venant des contextes auxquels ils sont soumis. Chaque agent a des attributs de personnage qui peuvent représenter leurs caractéristiques, par exemple dans le cas de simulation de personnages humains, leurs capacités (comme la force ou l'endurance) et leur personnalité (comme l'exubérance ou encore l'agressivité). Un attribut de personnage a un nom, une tendance et une valeur dans l'intervalle $[0, 100]$, comme décrit dans [15]. Une tendance est la valeur que prend l'attribut lorsqu'aucun contexte ne l'influence. Les attributs de personnage, définis par le concepteur en fonction de l'application, permettent de s'assurer que plusieurs agents dans les mêmes conditions extérieures (soumis aux mêmes contextes) n'adopteront pas toujours les mêmes comportements.

Les contextes peuvent influencer la valeur des attributs de personnage d'un agent (mais pas sa tendance). Par exemple, un contexte entourant une enceinte jouant une musique pourrait augmenter l'enthousiasme d'un agent si celui-ci se trouve sous l'influence du contexte.

Pour des raisons de clarté, nous avons ajouté dans COBAI un espace de travail lié à l'agent. Il s'agit d'une représentation de la situation globale de l'agent : elle contient tous les contextes auxquels il est soumis. L'agent y traite ces contextes et les règles de comportements qui y sont liées. L'espace de travail est détaillé dans la section 3.4.

3.2 Contextes

Dans [15], les contextes sont des éléments fondamentaux qui gèrent toutes les interactions des agents avec leur environnement (y compris les

autres agents). Ils servent de médiateurs entre l'environnement et les agents. Ainsi, toute information pertinente pour les agents (non exhaustivement : les obstacles, les événements, les autres agents, les objets) est associée à un contexte contenant les comportements appropriés. Dans une application, les contextes sont définis par le concepteur et peuvent ensuite être dynamiquement instanciés par les agents au cours de la simulation. Un contexte peut être localisé, comme un objet, ou non, comme une instruction donnée par un autre agent. Prenons par exemple le contexte *Incendie*. Les agents peuvent réagir à la perception du contexte, c'est pourquoi un contexte contient les comportements pertinents pour la situation qu'il représente. Ainsi, il est associé à des règles de comportement. Suivant l'exemple précédent, le contexte *Incendie* contient les comportements suivants : *Fuir*, *AppelerPompiers*, *Éteindre*. Les prémisses de la règle de comportement vont déterminer quel comportement l'agent adopte.

Un contexte possède une force dans l'intervalle $[0, 100]$. Les agents vont privilégier les comportements venant des contextes qui ont la plus grande force. Par exemple, considérons un agent soumis au contexte *Incendie* de force 90 et au contexte *Panneau* (représentant un panneau publicitaire) de force 30. Si l'agent satisfait les prémisses d'une règle de comportement portée par le contexte *Incendie*, il va exécuter ce comportement plutôt qu'un comportement concurrent venant du contexte *Panneau*. En effet, dans une situation réelle, le danger représenté par l'incendie est prioritaire sur l'information fournie par le panneau publicitaire. Le concepteur doit donc choisir la force des contextes de manière pertinente pour éviter les situations incohérentes telles qu'un agent observant un panneau publicitaire pendant un incendie. La force permet également de pondérer l'influence des contextes sur la valeur des attributs de personnage des agents. Par exemple, le contexte *Incendie* pourrait augmenter la valeur de l'attribut *Peur* des agents qui y sont soumis, en fonction des valeurs de la force du contexte et de la tendance de l'attribut pour chaque agent.

Un contexte peut être localisé ou non localisé. Un contexte localisé a une position dans l'espace et une zone d'influence qui représente les points de l'espace auxquels les agents sont sous son influence (sont soumis au contexte). Un contexte non localisé contient une liste révisée dynamiquement des agents sous son influence.

Un contexte localisé peut être associé à un agent.

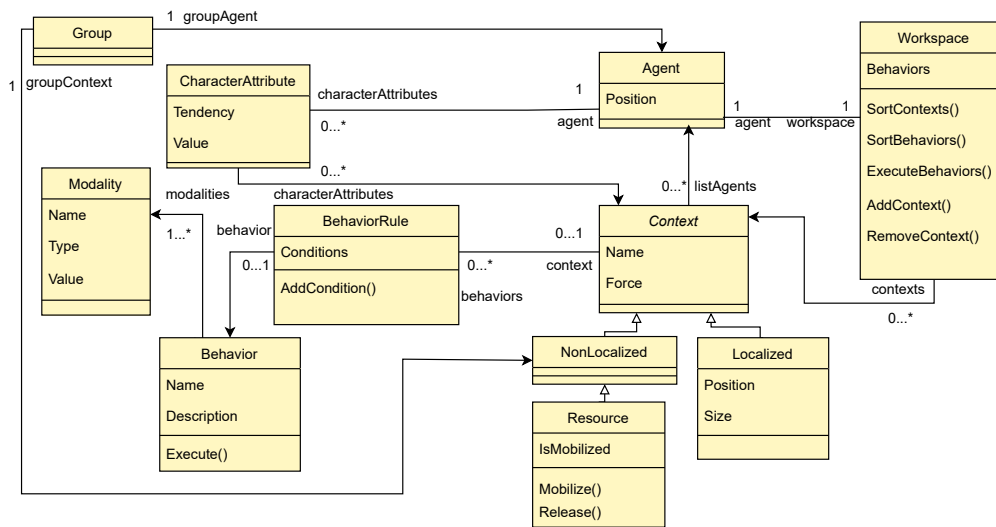


FIGURE 1 – Diagramme conceptuel de COBAI.

Sa position est alors à tout instant celle de l'agent. Un tel contexte peut représenter, par exemple, l'influence d'un agent sur les autres agents. Par exemple, dans [15], les collisions sont gérées par des contextes associés à chaque agent.

Dans COBAI, nous avons introduit un changement mineur dans la définition des contextes : la force d'un contexte peut désormais être une fonction mathématique. Cela permet de faire dépendre la force de la perception qu'un agent a du contexte. Elle peut dépendre par exemple des attributs de personnage de l'agent, ou de la distance à laquelle il se trouve de la source du contexte dans le cas d'un contexte localisé. Par exemple, un contexte peut être perçu comme plus fort lorsqu'un agent est proche de sa source, et plus faible au fur et à mesure qu'il s'éloigne.

3.3 Comportements

Dans [15], un comportement est toute action qu'un agent réalise dans la simulation. Il peut être visible pour l'utilisateur (animation graphique) ou non (modification des données de la simulation, par exemple de la force d'un contexte).

La principale contribution de COBAI est une nouvelle manière de gérer les comportements. Nous avons introduit les nouveaux concepts de règle de comportement incomplète, modalité, ressource et outil. Ces concepts seront détaillés dans cette section.

Le principe général de la nouvelle architecture de comportements est de pouvoir combi-

ner des comportements issus de contextes différents. Cela permet de prendre en compte plusieurs éléments de la situation dans laquelle se trouve l'agent, au lieu de choisir uniquement l'élément le plus pertinent. Plusieurs comportements peuvent être exécutés simultanément. Les conflits sont alors réglés par la gestion des ressources de l'agent.

Comme les comportements dépendent des contextes, ils doivent y être attachés. Ainsi, tout comportement dans la simulation fait partie d'une règle de comportement. Une règle de comportement comporte des prémisses (comme dans une règle logique), un identifiant de comportement, des modalités (incluant le code exécutable du comportement), et au moins un contexte associé. Une modalité est tout élément (donnée, script) nécessaire à l'exécution du comportement. Par exemple, un comportement *SeDéplacer* nécessite un script, une position, et une vitesse. L'association entre les comportements et leurs modalités est réalisée au niveau de la conception par l'identifiant du comportement. Pour exécuter un comportement, les conditions suivantes doivent être respectées. Chaque modalité doit être instanciée, un agent doit être sous l'influence du contexte associé et les prémisses doivent être satisfaites pour cet agent. Une règle de comportement dans laquelle au moins une modalité n'est pas instanciée est une règle de comportement incomplète. Nous utilisons le formalisme suivant pour représenter les règles de comportement :

$[premisses]IdComportement(modalités)$

$\{VariationsComportement()\}$

Dans le cas le plus simple, un contexte contient une règle de comportement complète. Selon ses attributs de personnage, un agent soumis à ce contexte a toutes les modalités pour exécuter le comportement correspondant. Un exemple simple de cette situation est un contexte non localisé contenant la règle de comportement suivante : $\llbracket SeDéplacer(destination = maison)\{Marcher()\}$, où *maison* est une variable contenant la position de la maison de l'agent. Un agent uniquement soumis à ce contexte va simplement marcher jusqu'à sa maison. Cependant, dans de nombreux cas dans COBAI, plusieurs contextes contiennent des règles de comportements incomplètes liées au même comportement. Un agent doit être soumis à plusieurs contextes complémentaires pour exécuter le comportement. Les règles de comportement incomplètes vont alors être combinées dans l'espace de travail de l'agent pour former une règle de comportement complète.

Les contextes non localisés peuvent représenter des ressources. Une ressource est un moyen accessible à l'agent (par exemple, ses bras et ses jambes) pour agir, dont l'accès est concurrent, de manière similaire à Lamarche *et al.* [4]. L'utilisation des ressources résout partiellement la compatibilité entre les comportements. Un contexte représentant une ressource contient des règles de comportements incomplètes nécessitant cette ressource, avec le script correspondant mais sans l'intégralité des modalités. Cela représente les compétences de l'agent, des comportements que l'agent peut adopter si la situation est appropriée. Lorsqu'un agent exécute un comportement *C* utilisant une ressource *R*, *R* est mobilisée, empêchant d'autres comportements nécessitant *R* de s'exécuter. Lorsque *C* est interrompu, *R* est libérée, les autres comportements peuvent donc à nouveau l'utiliser. Reprenons l'exemple précédent d'un agent rentrant chez lui en marchant. Au lieu de rencontrer un seul contexte contenant le comportement avec le script pour marcher et la destination, l'agent dispose d'une ressource *Jambes* avec une règle de comportement incomplète (règle 1) $\llbracket SeDéplacer(destination)\{Marcher()\}$. Lorsqu'il rencontre un contexte contenant une règle de comportement correspondante (règle 2) $\llbracket SeDéplacer(destination = maison)\{\}$, l'agent va combiner les deux règles de comportement dans son espace de travail pour obtenir une nouvelle règle de comportement complète : $\llbracket SeDéplacer(destination =$

*maison)\{Marcher()\}. Maintenant, considérons que les agents ont un attribut de personnage *Sportif*. Dans leur ressource *Jambes*, ils pourraient avoir deux règles de comportement incomplètes : $\llbracket Sportif < 80\rrbracket SeDéplacer(destination)\{Marcher()\}$ et $\llbracket Sportif \geq 80\rrbracket SeDéplacer(destination)\{Courir()\}$. Avec les prémisses que nous avons ajoutées, les agents vont adopter une version différente du comportement *SeDéplacer* en fonction de la valeur de leur attribut de personnage *Sportif* lorsqu'ils rencontrent le contexte complémentaire portant la règle de comportement correspondante $\llbracket SeDéplacer(destination = maison)\{\}$. Certains vont courir, d'autres vont marcher jusqu'à leur maison.*

Nous avons défini un type de ressource spécifique : les outils. Un outil représente un objet (par exemple, un téléphone) qu'un agent peut utiliser pour exécuter un comportement (par exemple, appeler quelqu'un sur le téléphone). Deux différences sont à noter entre les ressources ordinaires et les outils. Les outils sont généralement associés à une entité physique de la simulation et nécessitent de mobiliser d'autres ressources (suivant l'exemple précédent, un bras pour utiliser le téléphone).

Cette nouvelle architecture de comportements permet la représentation de situations plus complexes telles que nous pouvons en rencontrer dans la vie réelle. Un agent peut exécuter plusieurs comportements si ceux-ci sont compatibles, c'est-à-dire s'ils utilisent des ressources différentes. Des comportements peuvent provenir d'interactions indirectes entre plusieurs contextes au lieu de toujours provenir d'un seul.

3.4 Espace de travail de l'agent

L'architecture de comportements que nous avons introduite dans COBAI a rendu le procédé de sélection de comportement plus complexe en ajoutant des opérations (trouver des règles de comportement correspondantes, vérifier que chaque modalité est instanciée et combiner les règles de comportements). Ainsi, il est pertinent de le séparer de l'agent pour la clarté du processus de conception d'application. Nous avons ajouté un espace de travail attaché à chaque agent pour traiter les règles de comportement.

L'espace de travail stocke tous les contextes auxquels l'agent est soumis et les trie par force décroissante. À partir de la liste des règles de comportement associées à ces contextes ainsi

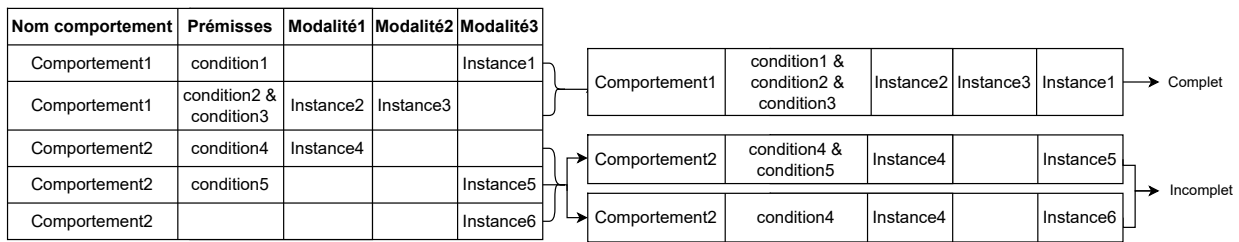


FIGURE 2 – Processus de combinaison de règles de comportement incomplètes permettant d'obtenir des règles de comportement complètes et incomplètes.

triés, il combine des règles de comportement incomplètes afin d'en obtenir des règles complètes qu'il pourra exécuter. Pour cela, il identifie les règles de comportement incomplètes portant les mêmes identifiants et vérifie qu'elles sont unifiées (chaque modalité ne doit être instanciée que dans une seule règle de comportement). Pour unifier les règles de comportement incomplètes, il doit d'abord combiner les prémisses de chaque règle en une unique expression logique. L'agent doit remplir chaque condition de chaque prémisses. Ainsi, les prémisses de chaque règle sont concaténées avec un opérateur logique *ET* entre elles. Il crée une nouvelle règle de comportement contenant toutes les instances de modalités des règles considérées. Après cette étape, certaines règles de comportement peuvent encore être incomplètes et seront alors ignorées pour l'étape suivante du traitement. Ce procédé est illustré par la figure 2.

En parcourant les règles de comportement complètes triées par ordre décroissant de la force de leurs contextes associés, l'agent exécute chaque comportement si les conditions suivantes sont remplies :

- L'agent satisfait les prémisses de la règle de comportement ;
- Les ressources associées à la règle de comportement sont libres (non mobilisées).

Soit l'exemple suivant : un médecin dans le poste médical avancé. En tant que médecin, l'agent possède une compétence donnée par une ressource partagée par tous les agents de type Médecin : $[TypeAgent = Médecin]TraiterVictime(victime, médicament)\{DonnerMédicament()\}$. L'agent a également un outil, une trousse médicale, portant la règle de comportement incomplète suivante : $[TypeAgent = Médecin]TraiterVictime(victime, médicament = TrousseMédicale)\{DonnerMédicament()\}$. Dans le poste médical, plusieurs victimes attendent un traitement, chacune ayant un contexte associé portant

la règle de comportement incomplète correspondante : $[TypeAgent = Médecin]TraiterVictime(victime = cetteVictime, médicament)\{DonnerMédicament()\}$. L'état de santé de la victime pondère la force de chaque contexte. Plus leur état de santé est grave, plus la force devient élevée. L'agent doit choisir une victime à traiter en premier. Il va privilégier le contexte avec la plus grande force. Avec la trousse médicale, la ressource venant du groupe des Médecins et le contexte de la victime, la règle de comportement complète suivante peut être exécutée :

$[TypeAgent = Médecin]TraiterVictime(victime = cetteVictime, médicament = TrousseMédicale)\{DonnerMédicament()\}$. Comme l'agent remplit les prémisses, il peut exécuter le comportement.

3.5 Groupes

Les groupes peuvent représenter une variété de structures impliquant plusieurs agents. Un groupe peut décrire une structure sociale (par exemple, une famille ou un groupe d'amis ou de collègues), un rassemblement circonstanciel (par exemple, une file d'attente ou un événement), ou une structure dédiée à une tâche spécifique. Nous pouvons distinguer deux types principaux de groupes : organisés et non organisés. Dans les groupes organisés, les agents ont des rôles au sein du groupe.

Bien que les groupes soient mentionnés dans [15], il s'agit seulement d'ajouter une propriété aux agents contenant le nom du groupe. Ainsi, cette propriété peut être utilisée dans des prémisses de règles de comportement. Cependant, cela ne permet ni la coopération entre les agents, ni des comportements de groupe, seulement de limiter certains comportements à des catégories d'agents spécifiques.

Par nature, un contexte réunit les agents qui y sont soumis. Il est alors naturel d'utiliser les contextes comme base des groupes dans COBAI.

Un contexte seul peut représenter un groupe circonstanciel non organisé où tous les agents exécutent le même comportement. Cela vaut pour les rassemblements autour d'un point d'intérêt (par exemple, une oeuvre d'art dans un musée) ou une action (file d'attente, spectateurs d'un événement culturel), un événement (une alarme par exemple) ou une tâche ponctuelle (comme chercher des victimes autour d'un accident). Le contexte porte alors les comportements pertinents pour la situation. En suivant les exemples précédents, il s'agirait de : regarder une oeuvre d'art, suivre la file, applaudir lors d'un événement culturel et se déplacer autour de la zone d'accident pour chercher des victimes.

Cependant, un groupe nécessite d'être plus explicite lorsqu'il représente une structure conceptuelle telle qu'une équipe, un groupe organisé (une armée, par exemple, avec des sous-groupes) ou lorsqu'il est plus naturel de donner des comportements à un niveau collectif qu'à des agents individuels.

Dans COBAI, ces groupes explicites sont représentés par un agent et un contexte non localisé qui lui est associé (un agent-groupe et un contexte-groupe). L'agent-groupe permet au groupe d'être influencé par des contextes et traite les comportements pour distribuer des tâches aux agents qui sont rassemblés par le contexte-groupe. Chaque agent dans le groupe est soumis au contexte-groupe. Ce mécanisme permet au groupe de fournir des comportements à ses membres.

Les comportements de groupe peuvent être plus complexes que lorsqu'ils n'impliquent qu'un seul agent. Le cas le plus simple est la distribution égale de tâches à chaque membre du groupe. Cependant, particulièrement dans les groupes organisés, certaines tâches nécessitent d'être divisées en différentes sous-tâches qui doivent être accomplies par des rôles spécifiques dans le groupe. Le groupe doit avoir au moins une ressource associée pour gérer ces deux cas. Elle doit contenir des règles de comportements incomplètes avec les scripts de comportements pour la distribution de tâches aux différents membres du groupe. Nous appelons cette ressource le moteur du groupe.

La Figure 3 montre une représentation de la distribution égale de tâches à tous les membres du groupe sur un exemple de pompiers se déplaçant vers une zone d'accident. De manière plus générale, le moteur du groupe contient une compétence qui correspond à une règle de compor-

tement incomplète fournie par un contexte localisé auquel le groupe est soumis. Le comportement complet obtenu en les combinant clone la règle de comportement issue du contexte localisé dans le contexte-groupe. Les membres du groupe ont accès à une ressource contenant une règle de comportement incomplète correspondante qui peut donc être combinée avec la règle de comportement clonée.

La Figure 4 montre une représentation de la distribution des tâches avec des rôles sur un exemple de division des pompiers en deux groupes ayant des rôles différents. Tout comme pour le cas précédent, le moteur du groupe et un contexte localisé contiennent des règles de comportements incomplètes qui se correspondent. Au lieu de cloner une règle de comportement, le comportement obtenu va créer de nouveaux groupes pour chaque rôle avec le moteur approprié et sélectionner des agents du groupe principal pour les ajouter aux nouveaux groupes.

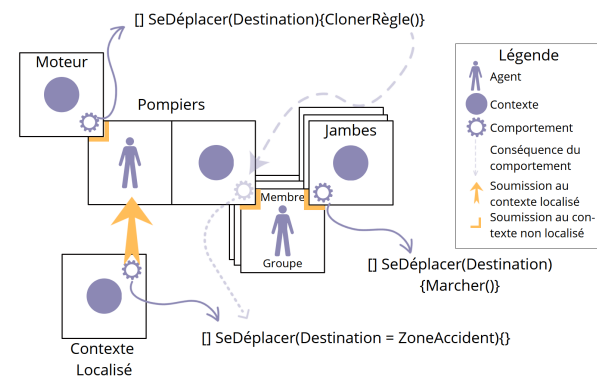


FIGURE 3 – Représentation schématique d'une distribution de tâches uniforme au sein d'un groupe sur l'exemple du déplacement d'un groupe de pompiers vers une zone d'accident.

4 Résultats

Nous appliquons COBAI à une étude de cas de situation de crise, une situation riche et complexe qui présente des interactions variées. En utilisant COBAI, nous représentons des pompiers, des policiers, des médecins, des infirmiers, des victimes et des badauds dans le cas d'une explosion de bâtiment.

Nous implémentons le modèle dans le moteur de jeux Unity. Le moteur de jeu fournit des outils pour créer des environnements tridimensionnels avec la physique et un système de navigation intégrés, qui ne figurent pas dans les points d'intérêt centraux de notre recherche.

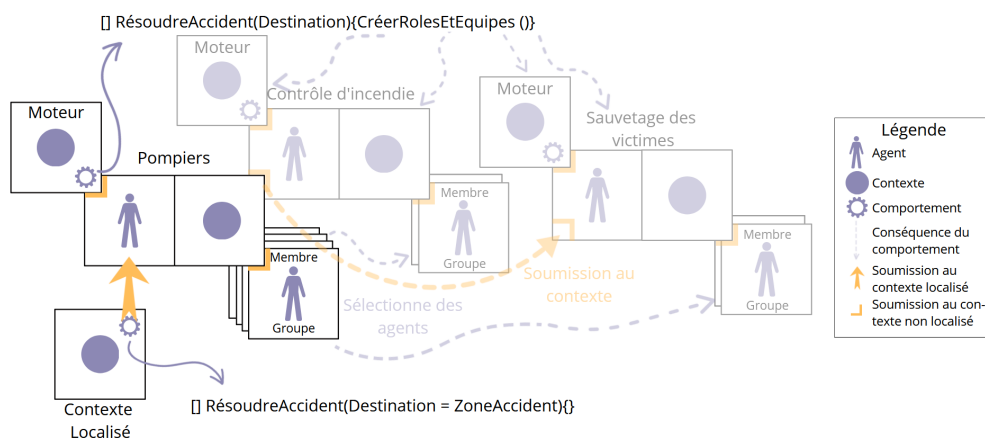


FIGURE 4 – Représentation schématique d’une distribution de tâches complexe au sein d’un groupe avec des rôles, sur l’exemple de pompiers se divisant en deux sous-groupes : contrôle d’incendie et sauvetage des victimes.

La simulation contient une zone d’accident représentée par un contexte localisé, où nous plaçons des victimes dont l’état de santé est variable. Chaque victime a un contexte associé contenant des règles de comportement pour que les pompiers, les infirmiers et les médecins puissent prendre soin d’elle.

Nous donnons à nos agents les ressources suivantes : jambes, bras, tronc et tête. Elles donnent les compétences nécessaires pour que les agents puissent accomplir les tâches. Ces compétences peuvent être génériques (non exhaustivement, marcher, courir) ou spécifiques à la fonction de l’agent (par exemple, donner des médicaments ou arrêter une personne). Outre ces ressources, nous ajoutons divers outils (par exemple des trousse médicales et des brancards) que les agents peuvent utiliser.

En calibrant soigneusement les forces des contextes, nous pouvons créer un séquençage dynamique de comportements. La ressource *Jambes* donne aux agents la règle de comportement incomplète `[] SeDéplacer(destination){Marcher()}`. Les pompiers sont soumis à un contexte non localisé de force faible (30) pour leur permettre de se déplacer vers la zone d’accident. Ce contexte comporte donc la règle de comportement incomplète complémentaire `[] SeDéplacer(destination = ZoneAccident){}`, où *ZoneAccident* contient les coordonnées de la zone où a eu lieu l’accident. La zone d’accident est un contexte localisé avec une force moyenne (60). Comme sa force est plus élevée que celle des contextes non localisés, lorsque les pompiers entrent dans sa

zone d’influence, ils commencent à chercher des victimes aléatoirement dans la zone d’accident. Chaque victime a un contexte associé avec une force élevée (80). Lorsqu’un pompier entre dans la zone d’influence du contexte associé à une victime, comme la force est plus élevée que celle des autres contextes, le pompier va se diriger vers la victime. Lorsqu’il l’atteint, l’un de ses comportements sera de diminuer considérablement la force du contexte associé à la victime pour éviter d’attirer d’autres pompiers. Lorsque le pompier atteint la victime, il la récupère et la ramène au poste médical avancé. La Figure 5 présente une capture d’écran de la situation dans la zone d’accident.



FIGURE 5 – Capture d’écran en mode éditeur de la zone d’accident dans la simulation. La zone d’accident est un contexte désigné par (1), et chaque victime a un contexte associé désigné par (2). Le cercle jaune représente la zone d’influence du contexte, associé à la victime, sélectionné sur la droite.

En utilisant COBAI dans Unity, nous obtenons des contextes dynamiques et des comportements avec séquençage naturel (les comportements s'enchaînent de manière cohérente par rapport à la situation). Les contextes peuvent être créés ou déplacés au cours de la simulation, auxquels les agents vont réagir en fonction de leurs priorités relatives modélisées par leurs forces respectives.

5 Discussions

Définir de multiples contextes complémentaires contenant des règles de comportement incomplètes peut paraître laborieux. Cette technique permet cependant de représenter différentes circonstances de comportements qui participent au réalisme de la simulation. De plus, elles sont simples à mettre en oeuvre pour le concepteur, car elles représentent naturellement des éléments conceptuels issus de l'analyse d'une situation réelle. Par exemple, nous pouvons distinguer un ou plusieurs contextes pour représenter :

- Un besoin qui ordonne d'adopter un comportement sans détailler comment le réaliser. Par exemple, un agent a besoin de se rendre au supermarché. Il peut y aller à pied, en voiture, ou en utilisant d'autres modes de transport. Ce contexte ne fournit aucune information concernant le moyen de transport.
- Des ressources, dont des outils, ajoutent du réalisme, permettant aux agents d'exécuter des comportements similaires différemment en fonction de leurs ressources. En suivant l'exemple précédent, cela fournit le moyen de transport : à pied, en voiture, en vélo ou en bus.
- Des circonstances : pour un comportement donné, elles donnent les informations nécessaires. Il peut s'agir, entre autres, d'une localisation spatiale ou temporelle, d'un objet ou une entité sur laquelle l'agent va appliquer le comportement. Dans l'exemple du supermarché, le contexte pourrait contenir l'emplacement du supermarché.

Concernant la compatibilité des comportements, comme mentionné par Larmarche *et al.* [4], n'autoriser qu'un comportement à la fois pour chaque ressource ne règle pas entièrement le problème. Les concepteurs doivent malgré tout être consciencieux lorsqu'ils associent des comportements à des ressources pour éviter des situations d'incohérence entre comportements simultanés. Cependant, notre solution permet d'exécuter plusieurs comportements simultanés avec un mécanisme simple. Nous envisageons l'ex-

ploration plus approfondie de ce problème dans nos travaux futurs pour améliorer cette solution.

6 Conclusion

Dans cet article, nous avons présenté un modèle générique à base d'agents pour la simulation de comportements. Nous avons utilisé un modèle précédemment développé par Soussi et Savelli, dans lequel des entités appelées contextes imposent des comportements aux agents. Nous avons amélioré ce modèle, en particulier l'architecture des comportements, lorsque les agents sont soumis à plusieurs contextes.

Nous avons intégré de nouveaux concepts au modèle, tels que les ressources, les outils et les règles de comportement incomplètes. Nous avons également défini le fonctionnement des groupes dans le modèle.

Nos prochains travaux seront consacrés à la mise en place d'une méthodologie de conception d'applications avec COBAI. Nous décrirons le procédé de conception associé au modèle et représenterons des situations classiques telles que l'échange d'information entre agents, la coopération entre agents avec des rôles définis pour une tâche commune, la combinaison de ressources et outils pour effectuer une tâche, ou différents types de groupes (par exemple des foules non structurées, des hiérarchies de groupes, ou des groupes dont les membres ont des rôles précis).

Nous visons à évaluer plus précisément les capacités et les limites du modèle au niveau conceptuel (quelles situations peuvent ou non être représentées par COBAI) et au niveau technique (évolution des performances en fonction de l'accroissement du nombre d'agents).

Nous travaillerons sur l'intégration de notre travail dans un jeu sérieux développé par notre partenaire, le CESU21, pour tester le modèle dans une situation concrète.

Remerciements. Le travail présenté dans cet article a été financé par la Région Bourgogne-Franche-Comté. Nous tenons également à remercier le Centre d'Enseignement des Soins d'Urgence (CESU21) et Medicaem pour leur partenariat sur ce projet de recherche.

Références

- [1] Fabien Badeig and Flavien Balbo. Modèle pour l'activation contextuelle : le mo-

- dèle eass. In *Proceedings Journées Francophones pour les Systèmes Multi-Agents (JFSMA'2006)*, pages 49–62, 2006.
- [2] Fabien Badeig, Flavien Balbo, and Mahdi Zargayouna. Dynamically configurable multi-agent simulation for crisis management. In Gordan Jezic, Yun-Heh Jessica Chen-Burger, Mario Kusek, Roman Šperka, Robert J. Howlett, and Lakhmi C. Jain, editors, *Agents and Multi-agent Systems : Technologies and Applications 2019*, pages 343–352, Singapore, 01 2020. Springer Singapore.
- [3] Oana Bucur, Philippe Beaune, and Olivier Boissier. Representing context in an agent architecture for context-based decision making. *CEUR Workshop Proceedings*, 136, 01 2005.
- [4] Lamarche Fabrice and Stéphane Donikian. Automatic orchestration of behaviours through the management of resources and priority levels. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems : Part 3, AAMAS '02*, pages 1309–1316, New York, NY, USA, 07 2002. Association for Computing Machinery.
- [5] James J. Gibson. The theory of affordances. In John Bransford Robert E Shaw, editor, *Perceiving, acting, and knowing : toward an ecological psychology*, pages pp.67–82. Hillsdale, N.J. : Lawrence Erlbaum Associates, 1977.
- [6] Avelino Gonzalez, Brian Stensrud, and Gilbert Barrett. Formalizing context-based reasoning : A modeling paradigm for representing tactical human behavior. *Int. J. Intell. Syst.*, 23 :822–847, 07 2008.
- [7] Sajjad Hassanpour and Amir Rassafi. Agent-based simulation for pedestrian evacuation behaviour using the affordance concept. *KSCE Journal of Civil Engineering*, 25, 01 2021.
- [8] Franziska Klügl and Sabine Timpf. Towards more explicit interaction modelling in agent-based simulation using affordance schemata. In Stefan Edelkamp, Ralf Möller, and Elmar Rueckert, editors, *KI 2021 : Advances in Artificial Intelligence*, pages 324–337, Cham, 2021. Springer International Publishing.
- [9] Franziska Klügl. Using the affordance concept for model design in agent-based simulation. *Annals of Mathematics and Artificial Intelligence*, 78, 09 2016.
- [10] Yoann Kubera, Philippe Mathieu, and Sébastien Picault. IODA : An interaction-oriented approach for Multi-Agent Based Simulations. *Journal of Autonomous Agents and Multi-Agent Systems*, 23(3) :303–343, 2011.
- [11] Yoann Kubera, Philippe Mathieu, and Sébastien Picault. Interaction-oriented agent simulations : From theory to implementation. In Ghallab, Malik, Spyropoulos, Constantine, Fakotakis, Nikos, Avouris, and Nikos, editors, *18th European Conference on Artificial Intelligence (ECAI'08)*, pages 383–387, Patras, Greece, 01 2008. IOS Press.
- [12] Rikke Amilde Løvliid, Solveig Bruvoll, Karsten Brathen, and Avelino Gonzalez. Modeling the behavior of a hierarchy of command agents with context-based reasoning. *The Journal of Defense Modeling and Simulation*, 15(4) :369–381, 2018.
- [13] Francesco Riccio, Emanuele Borzi, Guglielmo Gemignani, and Daniele Nardi. Context-based coordination for a multi-robot soccer team. In Luis Almeida, Jianmin Ji, Gerald Steinbauer, and Sean Luke, editors, *RoboCup 2015 : Robot World Cup XIX*, pages 276–289, Cham, 2015. Springer International Publishing.
- [14] Julien Saunier, Flavien Balbo, and Fabien Badeig. Environment as active support of interaction. In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *International Workshop on Environments for Multi-Agent Systems III*, volume 4389, pages 87–105, Berlin, Heidelberg, 05 2007. Springer Berlin Heidelberg.
- [15] Hakim Soussi, Joël Savelli, and Marc Neveu. A platform for the behavioral animation of crowds. In *Proceedings of the 2009 Summer Computer Simulation Conference*, pages 328–336, Vista, CA, 07 2009. Society for Modeling & Simulation International.
- [16] Roy M. Turner. Context-mediated behavior for intelligent agents. *International Journal of Human-Computer Studies*, 48(3) :307–330, 1998.
- [17] Afoutni Zoubida. *Un modèle multi-agents pour la représentation de l'action située basé sur l'affordance et la stigmergie*. PhD thesis, 09 2015.

Etat de l'art sur la co-simulation robotique et réseau des systèmes multi-robots

Théotime Balaguer^{a,b,c}
theotime.balaguer@insa-lyon.com

Olivier Simonin^a
olivier.simonin@insa-lyon.fr

Isabelle Guérin Lassous^b
isabelle.guerin-lassous@ens-lyon.fr

Isabelle Fantoni^c
isabelle.fantoni@ls2n.fr

^aINSA Lyon, Inria, CITI Lab., France

^bUniversité Lyon 1, ENS Lyon, CNRS, LIP, UMR 5668, France

^cNantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, France

Résumé

La simulation de systèmes multi-robots nécessite l'intégration des composantes robotique et réseau au sein d'un même environnement de simulation. La réutilisation des outils existants dans les communautés robotique et réseau semble être une évidence, mais la fusion de deux simulateurs présente des défis structurels qu'il faut surmonter afin d'obtenir une simulation "réaliste" d'un système multi-robots. Dans cet état de l'art, nous présentons les co-simulateurs qui abordent cette problématique et détaillons les défis auxquels il faut répondre pour la conception d'un co-simulateur performant.

Mots-clés : Systèmes multi-robots, Co-simulation, Réseaux de communication

Abstract

Simulation of multi-robots systems require the integration of both the robotic and network components. Leveraging the existing tools from each community seems evident, but merging two completely different simulators proves challenging. This paper presents the state of the art in co-simulators that tackle this problematic, and provide details on the challenges arising when one wants to create an efficient co-simulator.

Keywords: Multi-robots systems, Co-simulation, Communication networks

1 Introduction

Les systèmes multi-robots ont montré, ces dernières décennies, qu'ils pouvaient surpasser un robot seul pour certaines missions, et répondre à des problématiques impossibles à gérer avec une seule entité [4]. Grâce aux coûts toujours décroissants des composants électroniques, il est de plus en plus facile de déployer une flotte de robots collaboratifs qui achèvent les missions plus

vite, à plus grande échelle et avec plus de robustesse. Les développements de l'intelligence artificielle et de l'algorithmique distribuée participent par ailleurs grandement à la démocratisation des systèmes multi-robots.

Les systèmes de robots connectés cristallisent des problématiques à la frontière entre le domaine de la robotique et celui des réseaux [14]. Pour les étudier, il est essentiel de disposer d'outils de simulation réalistes pouvant simuler les deux aspects de front et permettant de mener des expériences rapides, reproductibles et sûres. Par exemple, la thématique des flottes de drones reflète parfaitement ce besoin de simulation conjointe entre la robotique et les réseaux. D'un côté, les drones ont des contraintes de déplacement qu'il faut respecter (rayon de braquage, accélération, etc.) et de l'autre, les échanges d'information entre les drones peuvent être vitaux pour la flotte (évitement de collisions, vol en formation, etc.) ou pour la mission (partage d'informations, relais réseau, etc.).

Nous verrons que les communautés de chacun des deux domaines – robotique et réseau – disposent déjà d'outils établis et performants [7][16]. Assembler ces outils, c'est-à-dire faire de la *co-simulation*, semble être une solution adaptée. Il est aussi important d'exploiter pleinement ROS (*Robot Operating System*)¹, une suite de logiciels facilitant le développement et l'intégration des systèmes robotiques et qui s'est imposée comme un outil essentiel dans la conception et l'opération de robots.

Dans ce papier, nous présentons et comparons les co-simulateurs existants et discutons des principales problématiques survenant lors de la conception de co-simulateurs. Cet état de l'art peut permettre de choisir le bon outil de co-simulation pour la recherche ou le développe-

1. ROS : <https://www.ros.org/>

ment de systèmes multi-robots communicants.

Organisation de l'article En Section 2, nous présentons les outils de simulation existants indépendamment pour la robotique et les réseaux. La Section 3 expose les co-simulateurs robotique et réseau existants. En Section 4, nous discutons l'architecture type d'un co-simulateur ainsi que les fonctionnalités recherchées. Les Sections 5 et 6 s'intéressent à deux problématiques récurrentes de la co-simulation, à savoir la synchronisation et l'échange d'information entre les simulateurs. Nous présentons nos premiers résultats et concluons en Section 7.

2 Simulateurs spécifiques

Si l'évaluation rigoureuse et exhaustive de simulateurs de robotique et de réseau dépasse largement les limites de cet article, il est nécessaire pour la compréhension de notre discours de présenter ici les simulateurs les plus notables pour chacun de ces domaines. Pour plus de détails, le lecteur peut se référer à [7] pour les simulateurs de robotique (aussi appelés simulateurs de physique) et [16] pour les simulateurs de réseau.

2.1 Simulateurs multi-robots

Le rôle du simulateur de robotique est de simuler les interactions du robot avec son environnement. Dans [7], les auteurs présentent un état de l'art des simulateurs de robotique complet mais sans se focaliser sur les systèmes multi-robots. Selon les auteurs, un simulateur de robotique doit contenir *a minima* :

- Un moteur simulant les phénomènes physiques ;
- Des modèles de friction et de collision ;
- Une interface graphique ;
- Une fonctionnalité d'import de scènes et de *meshes* ;
- Une interface de programmation applicative (API) permettant l'accès à ses propriétés ;
- Une bibliothèque d'effecteurs et de capteurs prêts à l'emploi.

De nombreux simulateurs de physique ont été développés ces dernières années. Certains sont spécialisés dans un sous-domaine de la robotique (robotique aérienne [24], sous-marine [18] [6], corps mous [10], etc.) alors que d'autres permettent de simuler une grande variété de robots. Pour la simulation de systèmes multi-robots, nous nous intéressons particulièrement à la flexibilité et l'efficacité du simulateur. La flexibilité d'un simulateur est son aptitude à s'adap-

ter facilement à de nouvelles situations (différents modèles de robots ou moteurs de physique, par exemple). L'efficacité d'un simulateur est sa capacité à utiliser l'ensemble des capacités de calcul mises à sa disposition.

Devant cette grande diversité de simulateurs, nous nous restreignons ici aux simulateurs multi-robots, actuellement maintenus et qui sont les plus utilisés par la recherche. Par exemple, Airsim [24] est un simulateur réaliste très utilisé mais ne supportant pas les systèmes multi-robots.

ARGoS. ArGoS [21] est un simulateur spécifiquement créé pour la simulation multi-robots. Ce simulateur est basé sur une architecture modulaire offrant à l'utilisateur une grande flexibilité et une bonne efficacité. En effet, différents modules caractérisés par leur précision et leur coût en calcul peuvent être choisis, pour allouer la puissance de calcul là où l'utilisateur considère que c'est nécessaire. ARGoS est capable de simuler de manière efficace des essais de plusieurs milliers de robots grâce à une caractéristique le démarquant des autres simulateurs : le monde 3D simulé peut être découpé en régions. L'utilisateur peut assigner à chaque région un moteur de physique différent (plus ou moins précis, par exemple), et les calculs associés à chaque région peuvent être parallélisés.

Webots. Le développement de Webots a démarré en 1996, faisant de ce simulateur l'un des plus anciens simulateurs multi-robots. Longtemps distribué sous licence propriétaire par Cyberbotics Ltd., Webots est distribué gratuitement depuis 2018, ce qui a accéléré son adoption par la communauté scientifique. En plus de fonctionnalités avancées et d'un haut niveau de réalisme, Webots propose des services annexes tels que l'accès à une version du simulateur sur le web² ou un support utilisateur. Concernant notre problématique robotique/réseau, il est intéressant de remarquer que Webot procure des noeuds *émetteur* et *récepteur* pouvant être utilisés pour modéliser des liaisons radio simples, en paramétrant une portée maximale de communication et un angle d'ouverture du cône d'émission pour les émetteurs infrarouges.

CoppeliaSim. CoppeliaSim [23], anciennement appelé V-REP, est un simulateur de robotique dont l'architecture permet une gestion efficace des systèmes multi-robots. Une étude récente comparant les performances et précision

2. RobotBenchmark : <https://robotbenchmark.net/>

de quatre simulateurs populaires dans la communauté de la robotique [11] place Coppeliasim parmi les meilleurs simulateurs multi-robots (Gazebo, Webots, Coppeliasim et MORSE sont étudiés). Bien que gratuit, Coppeliasim n'est pas open source, un facteur qui peut être limitant pour l'interfaçage avec d'autres simulateurs.

Gazebo. Gazebo (illustré figure 1) est le simulateur de robotique le plus utilisé dans la recherche [7], et ses fonctionnalités le rendent parfaitement adapté à la simulation multi-robots. Gazebo s'est construit autour du projet Player/Stage [13], puis son développement a été confié à l'"Open Source Robotics Foundation" (OSRF) en 2012. Pour améliorer sa flexibilité, le développement de Gazebo a pris un tournant majeur en 2019, avec la publication d'une version entièrement réécrite du simulateur. L'architecture monolithique historique ayant été remplacée par une architecture modulaire, Gazebo est capable de simuler tous types de robots, dans n'importe quel environnement, et en acceptant des possibilités d'extension infinies. La plus grande force de Gazebo réside cependant dans sa communauté nombreuse et active, qui partage de nouveaux modules, corrige des bugs et procure de l'aide en ligne. Le support des industriels à la fondation "Open Source Robotics"³ fournit un avantage non négligeable pour le maintien et le développement de ce simulateur. L'OSRF étant aussi responsable du développement de ROS, on peut facilement affirmer que ROS et Gazebo sont et resteront entièrement compatibles.

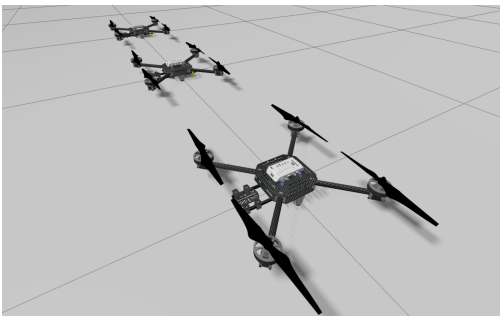


FIGURE 1 – Visualisation de trois robots quadricoptères dans Gazebo

2.2 Simulateurs réseaux

Un simulateur de réseau crée un modèle virtuel d'un réseau d'ordinateurs. Il permet d'expérimenter rapidement des algorithmes, d'évaluer

et de comparer les performances d'architectures et de technologies de communication diverses. L'unité principale de modélisation est souvent le "paquet", ce qui rend les simulateurs de réseau très réalistes pour simuler les protocoles mais moins pour les couches physiques (propagation notamment) et applicatives, qui sont représentées par des modèles plus abstraits.

Dans l'écrasante majorité des cas, les réseaux de robots sont des réseaux sans fil, nous nous intéresserons donc spécifiquement aux simulateurs performants dans la simulation des réseaux sans fil. Nous restreindrons de plus notre étude aux simulateurs open-source, actuellement maintenus et qui sont les plus utilisés pour la recherche. Certains simulateurs ont dû être écartés, comme Netsim⁴ qui est une alternative payante et OPNET⁵ qui est de moins en moins utilisé.

OMNeT++. OMNeT++ [26] est un simulateur à événements discrets modulaire dont la première publication date de 1997. OMNeT++ a la particularité de se présenter comme une *plateforme de simulation*, sur laquelle les équipes de recherche et les industriels peuvent construire leur propre *simulateur de réseau*. OMNeT++ fournit une interface graphique permettant de prototyper plus rapidement des réseaux grâce aux outils de visualisation.

Mininet. Mininet⁶ est un émulateur de réseau léger et facile d'utilisation créé pour la simulation des approches dites *Software Defined Network* (SDN). Son extension Mininet-Wifi [12] permet d'émuler des réseaux sans fil SDN. Mininet-WiFi permet des expérimentations riches et facilement déployables grâce à une distribution par conteneurs Docker. Il se révèle très utile pour l'étude des SDNs et de leur protocole de routage phare, OpenFlow [20].

NS-3. NS-3 [22] remplace son prédécesseur NS-2, le simulateur de réseau qui était déjà le plus utilisé dans les années 2010. Bâti sur l'expérience de ce simulateur largement adopté par la communauté, NS-3 capitalise sur des années de contributions d'une communauté nombreuse et active et reste aujourd'hui le simulateur de réseau le plus utilisé dans la recherche. La grande force de NS-3 réside dans le nombre et la diversité de modules disponibles, qui permettent de simuler de nombreuses technologies sous des topologies variées sans avoir à produire beaucoup de code. Notons cependant que NS-3 manque

3. Intrinsic Acquires OSRC and OSRC-SG : <https://tinyurl.com/5eb7bxfb>

4. Netsim™ : <https://netsim.boson.com/>

5. OPNET : <https://opnetprojects.com/opnet-network-simulator/>

6. Mininet : <http://mininet.org/>

d'une interface graphique intuitive et permettant de simuler rapidement des cas simples.

Enfin, les standards de développement sévères imposés par NS-3 assurent que la documentation est à jour et que chaque module est scrupuleusement testé avant son intégration dans le simulateur. Ces règles permettent aux chercheurs d'accorder une grande confiance aux résultats des études menées dans NS-3.

3 Co-simulateurs existants

Comme nous l'avons vu ci-dessus, les systèmes multi-robots peuvent être simulés dans leur composante robotique d'un côté et réseau de l'autre, mais aucun des simulateurs mentionnés n'intègrent ces deux parties du spectre en un seul logiciel. Dans cette partie nous proposons un tour d'horizon des initiatives qui tentent de résoudre cette problématique.

Les chercheurs se sont tournés vers la co-simulation (l'assemblage d'outils de simulation) pour éviter de reprendre à zéro le développements des simulateurs robotique et réseau.

Le tableau 1 synthétise notre étude en donnant pour chaque co-simulateur : les simulateurs de robotique et de réseau sur lesquels il se base, la technique de synchronisation utilisée ainsi que la technologie de partage d'information entre les simulateurs. Bien qu'intéressants, les simulateurs AVENS [19] et CUSCUS [27] ne seront pas discutés en détail car ils ne sont plus maintenus et n'implémentent pas de synchronisation entre simulateurs de robotique et de réseau.

FlyNetSim. FlyNetSim [3] est un environnement de co-simulation dédié aux essais de drones. Il utilise la pile logicielle Ardupilot⁷ à laquelle il ajoute une simulation du réseau gérée dans NS-3. Un mécanisme de synchronisation bidirectionnelle y est implémenté : un timestamp est ajouté à chaque message et utilisé pour déterminer quel simulateur est "en avance" sur l'autre. On gèle alors l'exécution du simulateur le plus rapide pour attendre le simulateur le plus lent. Les calculs de physique sont effectués dans un mini-simulateur de physique intégré à la version *Software-In-The-Loop* (SITL) d'Ardupilot, mais celui-ci n'est pas aussi avancé que les moteurs de physique utilisés dans des simulateurs comme Gazebo ou ARGoS.

FlyNetSim possède par ailleurs un mode "émulation" ou *Hardware-In-The-Loop* (HITL) dans lequel le programme de l'autopilote n'est plus

exécuté par l'ordinateur mais directement sur le drone. Le réseau et les entrées des capteurs sont quant à eux toujours simulés. Cette capacité permet de passer rapidement de la phase de simulation à la phase d'expérimentation en milieu réel.

CORNET et CORNET 2.0. Introduit en 2020, CORNET [1] est un middleware de co-simulation capable d'interconnecter Gazebo et NS-3 pour le cas des flottes de drones. La deuxième version de ce logiciel, CORNET 2.0 [2], publiée en 2022, capitalise sur les leçons de cette première expérience. NS-3 y est remplacé par Mininet-WiFi et des améliorations majeures sont apportées : CORNET 2.0 n'est plus limité aux robots aériens, est indépendant des simulateurs de physique et de réseau utilisés, ne dépend pas de ROS et profite du déploiement conteneurisé de Mininet, permettant de distribuer efficacement les calculs sur plusieurs machines.

La synchronisation temporelle est gérée par un plugin Gazebo. Cette approche ne peut fournir qu'une synchronisation unidirectionnelle du simulateur de réseau vers le simulateur de physique, ce qui engendre une grande contrainte : il faut que le simulateur de réseau fonctionne à un rythme plus élevé que le simulateur de physique. Malheureusement, cette hypothèse ne se vérifie pas dans le cas de réseaux fortement dynamiques, très denses ou avec des calculs de propagation réalistes [17].

RoboNetSim. RoboNetSim [17] est un environnement de simulation intégrant le simulateur de physique ARGoS et les simulateurs de réseau NS-2 ou NS-3. Créé en 2013, ce co-simulateur dédié à l'étude des grands essaims de robots (plusieurs dizaines) propose une synchronisation bidirectionnelle et un échange de message par socket. Des études de performances avec RoboNetSim ont montré les points suivants :

- NS-3 est plus performant (rapide) que son alter-ego NS-2 ;
- Le simulateur de physique (ici ARGoS) est le facteur limitant quand moins d'un certain nombre de robots communicants sont simulés (75 pour NS-2, 100 pour NS-3) ;
- Le sur-coût en temps de calcul de la co-simulation sont existants mais pas insurmontables. Pour 100 robots, le coût total de calcul est environ deux fois plus grand quand NS-3 est utilisé en parallèle d'ARGoS (ce qui ne veut pas dire que le temps total de la simulation est deux fois plus grand, puisque chaque simulateur s'exécute en parallèle) ;
- La co-simulation ne rajoute pas d'erreur : les résultats obtenus durant une co-simulation

7. Ardupilot : <https://ardupilot.org/>

TABLE 1 – Tableau comparatif des co-simulateurs robotique et réseau existants

Nom	Simulateur Multi-Robot	Simulateur Réseau	Synchronisation	Echanges d'information	Année	Open-Source	Réf.
RoboNetSim	ARGoS	NS-2 / NS-3	Time-stepped (Bidirectionnelle)	Socket (TCP et UDP)	2013	Oui	[17]
FlyNetSim	Ardupilot	NS-3	Time-stepped (Bidirectionnelle)	Message queue (ZMQ)	2018	Non	[3]
CPS-Sim	Matlab Simulink	QualNet OMNeT++	Variable time-stepped (Bidirectionnelle)	Custom (SNSP)	2018	Non	[25]
GzUav	Gazebo	NS-3	Exécution séquentielle	Unix Domain Socket + socket TCP	2019	Oui	[8]
CORNET	Gazebo	NS-3	Variable time-stepped (Unidirectionnelle)	Message queue (ZMQ)	2020	Oui	[1]
ROS-NetSim	Tous	Tous	Sliding Window (Bidirectionnelle)	Unix Domain Socket	2021	Oui	[5]
CORNET 2.0	Gazebo	MiniNet	Variable time-stepped (Unidirectionnelle)	Message queue (ZMQ)	2022	Oui	[2]
SynchroSim	Gazebo	NS-3	Variable time-stepped (Bidirectionnelle)	Non précisé	2022	Non	[9]

sont les mêmes que les résultats fournis par chaque simulateur, indépendamment.

SynchroSim. SynchroSim [9] est un co-simulateur récent qui se focalise sur la simulation de systèmes multi-robot hétérogènes, avec l'objectif principal de simuler l'opération coordonnée et collaborative de robots terrestres et aériens sur le champ de bataille. Il utilise Gazebo et NS-3, ainsi que ROS (1).

La principale originalité de cette proposition réside dans sa méthode de synchronisation. SynchroSim introduit une synchronisation basée sur une fenêtre glissante à taille variable, ce qui règle en partie la problématique récurrente du choix de la taille de la fenêtre. La taille de la fenêtre est ajustée en fonction de la différence de vitesse entre deux agents, mais les auteurs ne justifient pas ce choix et ils n'apportent pas de preuve de justesse de leur simulation. Enfin, ce simulateur n'est pas *open-source*, rendant difficile une évaluation critique.

CPS-Sim. CPS-Sim [25] accorde Matlab Simulink (simulation physique) et au choix QualNet ou OMNeT++ (simulation réseau). Il se démarque des autres co-simulateurs par sa méthode de synchronisation par pas variable, qui permet des calculs plus précis au prix d'un temps de simulation élevé, en particulier quand le nombre d'événements réseau augmente (voir section 5). En revanche, le choix de Matlab/Simulink pour modéliser la physique prive CPS-Sim des avantages d'utilisation de simulateurs robotiques existants, comme une interface visuelle ou un contrôle direct des actionneurs.

Pour certains problèmes, la méthode de synchronisation de CPS-Sim apporte une justesse de ré-

sultats nécessaire. Par exemple, dans le cas de la simulation de protocoles de synchronisation d'horloges dans un réseau de capteurs sans fil, la précision temporelle est à la fois très importante et sensible à la moindre erreur. Cette méthode de synchronisation est donc adaptée.

GzUav. GzUav [8] est un environnement de co-simulation qui modélise une flotte de drones communicants en assemblant trois modules complémentaires : Gazebo gère la simulation physique ; Ardupilot gère la simulation de l'autopilote (SITL) ; NS-3 simule les échanges réseau. Il permet une synchronisation bidirectionnelle et sa charge de travail peut être distribuée facilement sur plusieurs ordinateurs en réseau. L'intérêt de ce simulateur est l'intégration *by design* du code de l'autopilote, ainsi que son mécanisme de synchronisation par exécution séquentielle qui assure sa "justesse" au prix d'un temps de simulation élevé (la simulation physique avance d'un pas, attend que l'autopilote et la simulation réseau aient terminé leurs calculs, puis passe au pas suivant). Pour un réseau inférieur à 100 robots, GzUav montre des performances acceptables. S'il est très spécifique aux flottes de robots volants, ce simulateur se place parmi les co-simulateurs les plus avancés.

ROS-NetSim. L'ambition de ROS-NetSim [5] est de connecter un simulateur de physique avec un simulateur de réseau, peu importe les simulateurs choisis, pour n'importe quel scénario multi-robots, en se chargeant de la synchronisation bidirectionnelle et de l'échange de messages. Ce co-simulateur cherche à rester le plus général possible et utilise donc ROS pour s'adapter à tous types de robots. Sa méthode

de synchronisation est à fenêtre fixe et il utilise la technologie Protobuf⁸ pour l'échange d'information entre les simulateurs, ce qui permet à l'utilisateur de facilement changer le contenu des échanges.

ROS-NetSim propose aussi une méthode d'abstraction des informations du canal radio : le simulateur de physique fournit au simulateur de réseau les informations nécessaires (chemins d'ondes multiples et pertes associées) pour calculer l'affaiblissement dû à la propagation.

Le grand intérêt de ROS-NetSim est sa transparence du point de vue du robot. Un robot ROS ne peut distinguer s'il évolue dans un environnement simulé ou réel. Quand il est dans le monde réel, son trafic réseau sera géré par sa carte réseau, alors que s'il est simulé, ce trafic sera intercepté, passera dans le simulateur de réseau puis sera réinjecté au nœud destinataire. Cette architecture permet des simulations HITL ainsi qu'une transition fluide vers des expérimentations en milieu réel.

4 Architecture et propriétés de co-simulateur

Les co-simulateurs mentionnés ci-dessus ont des architectures proches : un programme sur-mesure initialise les simulateurs robotique et réseau puis met en place des canaux de communication entre ceux-ci. Il s'assure ensuite que les simulateurs fonctionnent de manière synchronisée tout au long de la simulation. Une représentation de l'architecture-type est donnée dans la figure 2.

Les co-simulateurs mentionnés fournissent cependant des fonctionnalités différentes. Nous listons ici les fonctionnalités qui semblent utiles pour la simulation de systèmes multi-robots.

4.1 Hardware In The Loop

Étudier un nouveau protocole ou une nouvelle technologie sur une version simulée d'un robot (SITL) peut mener à des erreurs ou des biais. Effectivement, les limitations du robot réel (puissance de calcul, énergie limitée, chauffe des composants, etc.) doivent être prises en compte dans le développement. Un simulateur permettant le *Hardware In The Loop* (HITL) permet de surpasser ces limitations en exécutant le programme du robot directement sur le robot réel. Seuls les entrées (capteurs) et sorties (actionneurs) sont alors simulés. Ainsi, le plan de dé-

8. Protocol Buffer : <https://protobuf.dev/>

veloppement d'une nouvelle technique multi-robots est le suivant : d'abord, une expérimentation rapide en simulation, puis une validation du logiciel exécuté sur le robot mais avec un environnement simulé, puis une expérimentation en milieu réel qui valide définitivement notre nouvelle technique.

4.2 Contrôle du temps

Il est intéressant d'avoir la possibilité d'enregistrer les expériences dans un format qui permet de rejouer la simulation, de la mettre sur pause ou de la jouer à l'envers. Ce format standard rend l'échange et la reproduction des expériences plus accessibles. La possibilité de rejouer les simulations avec un interface de visualisation est un plus non négligeable pour identifier rapidement les problèmes. Par exemple, pour les simulateurs fonctionnant autour de ROS, les *rosbags* sont des fichiers enregistrant l'ensemble des messages échangés pendant la simulation, permettant de rejouer le scénario exactement à l'identique.

4.3 Intégration ROS

Le *Robot Operating System* (ROS) est omniprésent dans la recherche en robotique aujourd'hui. Il sera donc essentiel que le co-simulateur accueille facilement des robots exécutant ROS. ROS présente une grande bibliothèque de modules (capteurs, algorithmes, ...) pouvant être facilement intégrés et interfacés avec un robot.

4.4 Performances

Enfin, les co-simulateurs seront *in fine* surtout évalués sur leur efficacité. Un co-simulateur rajoutant une surcharge de temps de calcul trop grande ne sera pas utilisé pour l'étude de systèmes multi-robots. Bien sûr, la performance d'un co-simulateur dépend directement des performances des simulateurs qu'il connecte. L'exécution d'une co-simulation ne pourra pas être plus rapide que le temps de calcul du plus lent des simulateurs (robotique ou réseau).

5 Synchronisation temporelle

La synchronisation des horloges entre plusieurs simulateurs est une problématique récurrente de la co-simulation. Les simulateurs de robotique décrivent la dynamique et cinématique des modèles grâce à des équations différentielles, qui doivent être discrétisées pour être implémentées de manière logicielle. Cette discrétisation

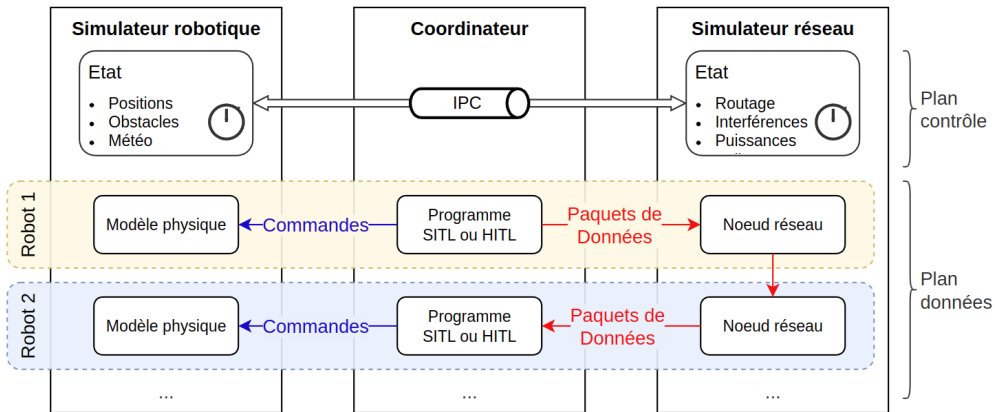


FIGURE 2 – Architecture représentative d'un co-simulateur robotique et réseau

est faite avec un temps d'échantillonnage, qui constitue la granularité de la simulation. Ces simulateurs sont dits *temps discret à pas constant* (Gazebo, ARGoS, Webots, etc.). Les simulateurs de réseau n'ont pas la contrainte d'un temps qui s'écoule régulièrement. Un nœud peut être silencieux pendant un temps long, puis déclencher une suite d'événements en décidant d'émettre un paquet. Dans ces simulateurs, le temps simulé avance d'événement en événement, le temps entre ceux-ci pouvant changer à loisir. Ce sont des simulateurs à *événements discrets* (NS-3, OMNeT++, QualNet, etc.)

Cette différence de nature rend nécessaire l'implémentation de mécanismes de synchronisation qui peuvent être unidirectionnels (un simulateur utilise l'horloge simulée de l'autre simulateur) ou bidirectionnels.

Cette section présente les techniques de synchronisation utilisées dans les co-simulateurs existants. Une représentation graphique de ces techniques est donnée en figure 3.

Time-stepped. C'est l'approche la plus simple : la synchronisation a lieu régulièrement avec une période multiple de la durée du pas du simulateur de physique (sur la figure 3, cette période est de 5 pas). A chaque synchronisation, les simulateurs échangent leurs statuts et variables puis avancent indépendamment jusqu'à la prochaine synchronisation. Cette méthode est simple à implémenter mais présente deux problèmes affectant la justesse des calculs : un délai injustifié est introduit pour le simulateur de robotique et les informations de mobilité fournies au simulateur de réseau sont imprécises.

Le premier effet s'explique facilement : le simulateur de robotique est notifié de la réception d'un message seulement lors de la prochaine synchronisation. Il reçoit toujours les paquets

avec un délai qui dépend de la taille de la fenêtre de synchronisation. Si les paquets arrivent de manière uniforme et aléatoire, le délai moyen introduit est donc de la moitié de la fenêtre de synchronisation (vérifié en pratique dans [17]).

Le deuxième problème est dû à la fréquence des mises à jour des positions dans le simulateur de réseau. En effet, les robots sont considérés immobiles durant toute la fenêtre de simulation, ce qui peut entraîner des imprécisions dans les calculs du simulateur de réseau. Cependant, cet effet ne représente en pratique que de faibles imprécisions [17].

Variable time-stepped. Pour améliorer la précision des calculs, il est possible d'adapter la méthode précédente en brisant la linéarité du simulateur de physique. La procédure est alors la suivante : si le simulateur de réseau possède un événement pendant la prochaine fenêtre du simulateur de physique, ce dernier se met sur pause. Le simulateur de réseau calcule l'événement puis envoie au simulateur de physique les informations de synchronisation et le *timestamp* de l'événement. Le simulateur de physique doit alors avancer jusqu'à ce *timestamp*. Avec cette méthode, le simulateur de physique obtient les informations des activités du réseau sans délai, ce qui représente un avantage dans les applications où le réseau et la robotique sont très intriqués. En revanche l'implémentation de pas variables dans le simulateur de physique se révèle souvent techniquement difficile. Les performances de cette méthode dépendent beaucoup de l'application étudiée : s'il y a peu d'événements réseau, les performances sont bonnes car il y a peu de synchronisation et donc peu d'échanges de messages. En revanche avec un grand nombre d'événements réseau, la surcharge de temps de calcul augmente dramatiquement.

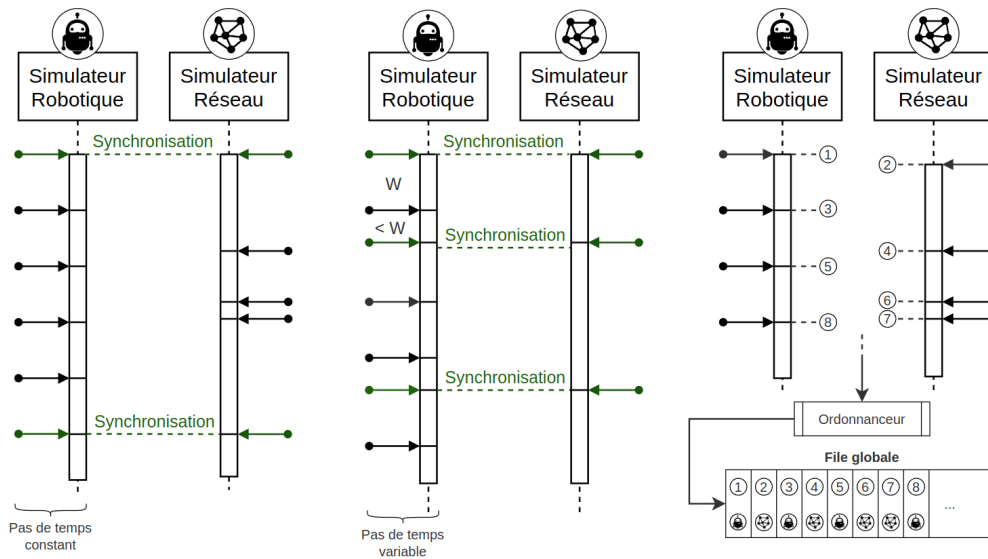


FIGURE 3 – Méthodes de synchronisation de simulateurs de physique et de réseau. de gauche à droite : *time-stepped*, *variable time-stepped*, *global event-driven*

Global Event-driven. Une approche radicalement différente consiste à gérer les deux simulateurs depuis un ordonnanceur unique. Chaque pas du simulateur de physique est alors traité comme un événement et mis dans la même file d'attente que les événements du simulateur de réseau. Les événements sont exécutés de manière séquentielle. L'avantage majeur de cette méthode est qu'elle n'engendre pas de délai supplémentaire ni d'approximations de calcul. En retour, les événements des deux simulateurs n'étant pas calculés simultanément, le coût en temps de calcul est décuplé.

6 Flux d'information entre les simulateurs

6.1 Technologie d'échange

Les simulateurs s'échangent des informations tout au long de la simulation, au minimum pour se synchroniser mais aussi pour collaborer. Ces échanges de messages représentent l'essentiel de la surcharge de calcul générée par la co-simulation, il est donc important de choisir une technologie de communication judicieusement. Le tableau 1 précise les techniques de communication utilisées dans les co-simulateurs étudiés.

Si détailler les avantages et inconvénients de chacune de ces techniques sort du cadre de ce papier, nous discutons ci-après les techniques utilisées dans les co-simulateurs que nous avons étudiés. Le choix de la technique de communication

entre simulateurs sera avant tout déterminé par l'exécution distribuée du co-simulateur : s'il doit pouvoir s'exécuter sur plusieurs ordinateurs en réseau (pour partager la charge de calcul), l'utilisation de *sockets* Internet (UDP ou TCP) sera presque automatique. Dans le cas contraire, les autres techniques de communication inter-processus (*Inter-Process-Communication*, IPC) peuvent s'avérer plus efficaces.

Les *sockets* sont probablement la technique la plus versatile. Ils peuvent être utilisés localement ou en réseau (permettant des calculs distribués), sont indépendants de la plateforme (permettant l'exécution sur différents systèmes d'exploitation) et proposent des communications bloquantes ou non-bloquantes (ce qui rend possible la synchronisation des horloges).

Les files de messages présentent moins de charge de calcul que les *sockets* mais dépendent du système d'exploitation et peuvent être problématiques quand le nombre de message est grand.

La mémoire partagée ne peut pas être utilisée si les calculs doivent être distribués sur plusieurs machines, mais représente la technique la plus efficace en temps de calcul. Cependant, cette technique requiert une organisation rigoureuse des accès en mémoire, complexifiant le développement du co-simulateur.

6.2 Contenu des échanges

Le contenu des échanges entre les simulateurs relève aussi une importance déterminante. Il est crucial de correctement définir les informations

échangées : partager trop d'information générera une surcharge du temps de calcul, mais en partager trop peu limite l'intrication des simulateurs, rendant le co-simulateur inutile. Par exemple dans notre cas, il est primordial que les positions des robots soient cohérentes entre la simulation de physique et la simulation de réseau. Le concepteur du co-simulateur doit donc répondre à deux questions essentielles : quel est le contenu des messages échangés entre les deux simulateurs, et à quelle fréquence ces échanges ont-ils lieu ?

Le problème de l'abstraction du canal radio est un exemple intéressant dans le cas des simulations de systèmes multi-robots. Pour déterminer la réussite de réception d'un paquet entre deux robots, le simulateur de réseau doit calculer un bilan de liaison et comparer la puissance reçue au seuil de réception du robot destinataire. Pour cela, il doit détenir une représentation du canal radio (distance, obstacles, interférences, etc.), qui soit cohérente avec la représentation du monde dans le simulateur de physique. Pour faire passer ces informations du simulateur de physique au simulateur de réseau, [5] propose une abstraction du canal qui cherche à compresser les informations essentielles nécessaires au calcul du bilan de liaison.

La question du niveau de détail nécessaire à la simulation des réseaux sans fil revêt en réalité une grande complexité. [15] étudie cette question en profondeur et montre que c'est la question scientifique qui doit dicter la finesse de la simulation. Dans certains cas, un haut niveau d'abstraction permettra d'accélérer la simulation et d'éviter que les résultats propres à la question posée ne soient pollués par les artefacts d'une simulation trop précise. Dans d'autres cas, une simulation trop abstraite peut empêcher le chercheur de déceler les problèmes que son nouvel algorithme ou nouveau protocole pourrait rencontrer.

7 Conclusion

Dans le cadre des systèmes multi-robots communicants, nous avons présenté la nécessité de simuler simultanément la physique et le contrôle des robots d'un côté, et les flux d'information traversant le réseau de l'autre. Une étude rapide des simulateurs de robotique et de réseau existants a montré qu'il était essentiel de bénéficier des années de développement dans chacun de ces domaines. Les équipes de recherche se sont donc tournées vers la co-simulation et nous en avons présenté les initiatives les plus pertinentes. Ces projets ont chacun leurs avantages, mais aucun

n'a pour l'instant fait consensus dans la communauté multi-robots. En particulier, le maintien de ces plateformes et leur compatibilité avec les outils modernes (ROS, Ubuntu, etc.) est à notre sens une barrière difficile à surmonter.

Travaux en cours Dans le cadre du projet ANR/AID CONCERTO, nous avons débuté le développement d'un environnement de co-simulation robotique et réseau adapté aux flottes de robots aériens. Après une étude approfondie des co-simulateurs existants, nous avons choisi d'utiliser ROS-NetSim, Gazebo et NS-3. Les principes de simulation apportés par ROS-NetSim sont les plus concluants. Sa grande flexibilité et l'intégration à ROS jouent aussi en sa faveur. Cependant, ROS-NetSim n'est qu'une preuve de concept et son intégration réelle avec des simulateurs de robotique et de réseau nécessite le développement de connecteurs appropriés. Nos choix de Gazebo et NS-3 ont été essentiellement motivés par la présence d'une communauté active et nombreuse qui accélérera le développement de notre co-simulateur. Nous avons commencé par la modernisation de ROS-NetSim (passage à ROS 2, changements mineurs) puis nous avons développé un coordinateur permettant de connecter la dernière version de Gazebo au co-simulateur. Les prochaines étapes de nos travaux comprendront donc l'implémentation d'une simulation de réseau sans fil basé sur la technologie Wi-Fi dans NS-3 et l'intégration d'un programme d'auto-pilote (PX4) permettant de réaliser des simulations SITL réalistes. Enfin, la mise en place d'une expérience simple mettant en scène le réseau et le déplacement des drones permettra de valider la pertinence de notre plan de recherche en trois temps : simulation SITL ; simulation HITL ; déploiement en milieu réel.

Remerciements

Ce travail a été financé par l'Agence de l'Innovation de Défense et l'INSA Lyon et dans le cadre du projet CONCERTO (ANR-20-ASTR-0003).

Références

- [1] S. ACHARYA et al. "CORNET : A Co-Simulation Middleware for Robot Networks". In : COMSNETS. 2020.
- [2] S. ACHARYA et al. "CORNET 2.0 : A Co-Simulation Middleware for Robot Networks". In : COMSNETS. 2022.

- [3] S. BAIDYA et al. "FlyNetSim : An Open Source Synchronized UAV Network Simulator based on ns-3 and Ardupilot". In : *Proceedings of the ACM MSWIM*. 2018.
- [4] İlker BEKMEZCI et al. "Flying Ad-Hoc Networks (FANETs) : A survey". In : *Ad Hoc Networks* (1^{er} mai 2013).
- [5] M. CALVO-FULLANA et al. "ROS-NetSim : A Framework for the Integration of Robotic and Network Simulators". In : *IEEE Robotics and Automation Letters* (2021).
- [6] P. CIEŚLAK. "Stonefish : An Advanced Open-Source Simulation Tool Designed for Marine Robotics, With a ROS Interface". In : OCEANS MTS/IEEE. 2019.
- [7] J. COLLINS et al. "A Review of Physics Simulators for Robotic Applications". In : *IEEE Access* (2021).
- [8] F. D'URSO et al. "An integrated framework for the realistic simulation of multi-UAV applications". In : *Computers & Electrical Engineering* (2019).
- [9] E. DEY et al. "SynchroSim : An Integrated Co-simulation Middleware for Heterogeneous Multi-robot System". In : DCOSS. 2022.
- [10] E. J. F. DICKINSON et al. "COMSOL Multiphysics® : Finite element software for electrochemical analysis. A mini-review". In : *Electrochemistry Communications* (2014).
- [11] A. FARLEY et al. "How to pick a mobile robot simulator : A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion". In : *Simulation Modelling Practice and Theory* (2022).
- [12] R. R. FONTES et al. "Mininet-WiFi : Emulating software-defined wireless networks". In : International CNSM. 2015.
- [13] B. P. GERKEY et al. "The Player/Stage Project : Tools for Multi-Robot and Distributed Sensor Systems". In : *Proceedings of the ICAR*. 2003.
- [14] Samira HAYAT et al. "Survey on Unmanned Aerial Vehicle Networks for Civil Applications : A Communications Viewpoint". In : *IEEE Communications Surveys & Tutorials* (2016).
- [15] J. HEIDEMANN et al. "Effects of Detail in Wireless Network Simulation". In : USC/ISI. 2001.
- [16] M. H. KABIR et al. "Detail Comparison of Network Simulators". In : *IJSER* (2014).
- [17] M. KUDELSKI et al. "RoboNetSim : An integrated framework for multi-robot and network simulation". In : *Robotics and Autonomous Systems* (2013).
- [18] M. M. M. MANHÃES et al. "UUV Simulator : A Gazebo-based package for underwater intervention and multi-robot simulation". In : OCEANS MTS/IEEE. 2016.
- [19] E. A. MARCONATO et al. "AVENS - A Novel Flying Ad Hoc Network Simulator with Automatic Code Generation for Unmanned Aircraft System". In : Hawaii International Conference on System Sciences. 2017.
- [20] N. McKEOWN et al. "OpenFlow : enabling innovation in campus networks". In : *ACM SIGCOMM Computer Communication Review* (2008).
- [21] C. PINCIROLI et al. "ARGoS : a modular, parallel, multi-engine simulator for multi-robot systems". In : *Swarm Intelligence* (2012).
- [22] G. F. RILEY et Thomas R. HENDERSON. "The ns-3 Network Simulator". In : *Modeling and Tools for Network Simulation*. Springer, 2010.
- [23] E. ROHMER et al. "V-REP : A versatile and scalable robot simulation framework". In : IEEE/RSJ IROS. 2013.
- [24] S. SHAH et al. "AirSim : High-Fidelity Visual and Physical Simulation for Autonomous Vehicles". In : Springer Proceedings in Advanced Robotics. 2018.
- [25] A. SUZUKI et al. "CPS-Sim : Co-Simulation for Cyber-Physical Systems with Accurate Time Synchronization". In : IFAC NECSYS. 2018.
- [26] A. VARGA et Rudolf HORNIG. "An overview of the OMNeT++ Simulation Environment". In : ICST. 2010.
- [27] N. R. ZEMA et al. "CUSCUS : An integrated simulation architecture for distributed networked control systems". In : IEEE CCNC. 2017.

Gestion des connaissances partagées par des agents à ressources et connectivité limitées : étude, analyse et expérimentation

M. Limame^a

mohamed.limame@etu.sorbonne-universite.fr

J. Henriet^b

julien.henriet@univ-fcomte.fr

C. Lang^b

christophe.lang@univ-fcomte.fr

N. Marilleau^a

nicolas.marilleau@ird.fr

^aInstitut de recherche pour le développement,
Sorbonne Université, France

^bInstitut FEMTO-ST,
université de Franche-Comté, France

Résumé

Un système distribué se compose d'un ensemble de nœuds connectés, communiquant et partageant des données. N'ayant pas accès à une horloge centrale, la cohérence de ces données partagées est l'une des problématiques importantes posées par ce type de système. Des approches permettant d'instaurer une cohérence temporelle de ces données partagées ont été conçues et implémentées par le passé. Cette problématique étant également rencontrée dans les systèmes multi-agents, notre étude présente et analyse les protocoles de gestion de la cohérence temporelle existants afin d'identifier ceux qui sont en adéquation avec les spécificités des systèmes multi-agents à ressources et connectivité limitées et faiblement accointants. Nous proposons dans cet article une classification et une analyse comparative de ces protocoles en vue d'une utilisation au sein de ses systèmes multi-agents. Nous allons aussi effectuer une simulation de la nouvelle approche *smasdev*, une des méthodes de synchronisation étudiées, pour interpréter ensuite les résultats obtenus.

Mots-clés : Gestion de la cohérence, cohérence temporelle, données partagées, synchronisation, systèmes multi-agents, systèmes multi-agents à ressources et connectivité limitées, *smasdev*

Abstract

A distributed system consists of a set of nodes connected, communicating and sharing data. Since it does not have access to a central clock, the consistency of this shared data is one of the important issues posed by this type of system. Approaches to establish temporal consistency of this shared data have been designed and implemented in the past. As this issue is also encountered in multi-agent systems, our study presents

and analyses existing protocols for managing temporal coherence in order to identify those that are in line with the specificities of weakly adjoining multi-agent systems with limited resources and connectivity. We propose in this paper a classification and a comparative analysis of these protocols for use within this category of multi-agent systems. We will also perform a simulation of the new *smasdev* approach, one of the studied synchronisation's methods, and then interpret the results obtained.

Keywords: Coherence management, temporal coherence, shared data, synchronisation, multi-agent systems, multi-agent systems with limited resources and connectivity, *smasdev*

1 Introduction

Au regard de l'importance que représente la notion temps dans les systèmes multi-agents dans l'interprétation des informations recueillies par chacun des agents et dans la détermination du comportement des agents notamment lors des échanges de données, nous nous intéressons dans cet article aux approches utilisées dans les systèmes distribués et dont l'objectif est d'assurer une cohérence temporelle des données partagées. Nous proposons dans cet article une classification et une analyse comparative des protocoles de la littérature relative aux systèmes distribués en vue d'une utilisation au sein d'un système multi-agents (SMA) à ressources et connectivité limitées.

Selon B. Chaib-Draa [2], un SMA est un système distribué composé d'éléments particuliers appelés agents qui partagent des connaissances et communiquent entre eux afin de résoudre un problème qui dépasse la portée d'un seul agent.

De ce fait, la problématique de la gestion de la cohérence temporelle s'impose également aux SMA, notamment dans les domaines d'applications sensibles comme la surveillance. En effet, il est nécessaire que les informations recueillies par chacun des agents pris individuellement soient cohérentes avec celles des autres agents du SMA pour que l'objectif global du système soit atteint. Comme précisé, nous nous focalisons dans cette étude sur les SMA à ressources et connectivité limitées qui se caractérisent par une faible capacité de calcul, une faible capacité de stockage, une énergie limitée et notamment une activité réduite focalisée sur sa survie avec un faible besoin en connectivité.

Selon A. Tanenbaum [28], un système distribué est un ensemble d'ordinateurs autonomes reliés par un réseau informatique qui apparaît à ses utilisateurs comme un seul système cohérent. Ces ordinateurs partagent des informations et des ressources sur une large étendue géographique ce qui a donné lieu dans certaines applications à des problèmes de cohérence temporelle notamment dans les applications temps réel utilisées dans les usines, l'aéronautique, les véhicules spatiaux et les applications militaires et également plus largement dans la gestion des versions et le contrôle des accès concurrents dans les systèmes distribués de base de données. Dans les systèmes distribués plusieurs approches ont été identifiées pour pallier à cette problématique. Il est impératif de comprendre les caractéristiques des agents pour distinguer parmi les approches utilisées dans les systèmes distribués celles qui sont en adéquation avec leurs besoins et contraintes.

Les principales contributions de cet article sont une classification des approches et algorithmes de synchronisation les plus utilisés dans les systèmes distribués et une expérimentation de la nouvelle approche smasdev. Le travail de classification et les résultats d'expérimentation de smasdev aideront ainsi les chercheurs dans l'établissement du choix de la technique de synchronisation la plus en adéquation avec les contraintes de leur propre système multi-agents faiblement accointants à ressources et connectivité limitées.

Dans ce qui suit, nous allons dans un premier temps définir la cohérence pour passer ensuite à la définition de la cohérence temporelle. Nous passons ensuite à la présentation des approches de synchronisation qui seront étudiées dans cet article et qui permettent d'instaurer une cohérence temporelle au sein d'un système multi-agents. Puis nous définissons les critères de clas-

sification avant de présenter la qualification des approches selon ces critères. Nous passons ensuite par une analyse comparative des approches citées. Nous terminerons par une simulation de l'approche smasdev et une interprétation des résultats obtenus.

2 Cohérence : concept et méthodes

2.1 Définitions de la cohérence

Il existe plusieurs définitions de la cohérence dans la littérature comme le confirme M. Olsson [21]. DJW Strümpfer [27] a défini la cohérence comme un moyen de mesure de l'absence de concept d'affectivité négative qui est similaire au névrosisme. En linguistique, selon B. Grosz et CL. Sidner [3] et K. Tracy et RT. Craig [9], un discours (texte ou dialogue) peut être qualifié de cohérent lorsque ses parties "vont ensemble" et structurent ainsi le discours. P. Thagard [29] [30] apporte une vision épistémologique en définissant la cohérence comme étant une théorie cognitive traduisant un niveau de satisfaction des contraintes contenues dans la connaissance individuelle sous ses différentes formes : explicative, analogique, déductive, perceptuelle et conceptuelle.

Dans les systèmes distribués, selon J. Sablatnig *et al.* [13] le concept de cohérence se traduit par la conservation d'une logique notamment temporelle, explicative ou conceptuelle assurée par une algorithmique et des contraintes. Sa gestion s'avère essentielle pour éviter les erreurs de calcul, d'une part, et les anomalies d'exécution des algorithmes d'autre part.

Dans les systèmes multi-agents, selon Qan. Vu [31] la cohérence des informations partagées dans le système est définie par la compatibilité des informations recueillies et stockées par chacun des agents. Contrairement à la cohérence dans les systèmes distribués où celle-ci est souvent globale et associée à l'ensemble du système. La cohérence des données partagées dans les systèmes agents est plutôt émergente et résulte des cohérences individuelles.

La cohérence temporelle dans un système distribué a été introduite par J-P. Thomesse [14] lors des travaux de recherche sur les réseaux de terrain et est relative à la notion de « fenêtre temporelle ». Pour ce dernier, la notion de cohérence temporelle s'applique à un ensemble d'informations et vise à garantir que les informations appartenant à cet ensemble ont été pro-

duites dans une même « fenêtre temporelle ». Selon L. Cauffriez et J. Defrenne [1], deux types de cohérence temporelle peuvent être distingués dans les systèmes répartis par rapport au niveau d'exigence de l'exactitude de la notion temps entre les différents éléments du système multi-agents. Une cohérence temporelle stricte dans laquelle tous les éléments et connaissances détenus par l'ensemble des agents ont exactement le même estampille temps et une cohérence temporelle non stricte dans laquelle des différences dans l'estampillage des éléments et connaissances peuvent exister entre les agents. La cohérence temporelle au niveau global est atteinte lorsque la cohérence temporelle de chaque élément du système multi-agents est respectée.

2.2 Méthodes

Parmi les outils et méthodes pouvant être appliquées pour instaurer une cohérence temporelle, il y a la technique de synchronisation de la notion temps entre agents d'un SMA par l'utilisation d'un protocole de synchronisation, il y a aussi selon J. Ferber [12], la technique de l'**arbitrage** qui conduit à la définition de règles de comportement qui agissent comme des contraintes sur l'ensemble des agents. J. Ferber [12] évoque également l'élaboration d'une **convention de priorité** ou l'établissement d'une **procédure de vote et élection**. Selon S. Bussmann et H.J. Müller [24], il existe aussi la **négociation** qui permet d'atteindre un accord mutuellement accepté entre agents.

Différents protocoles de synchronisation sont basés sur des horloges physiques et logiques. **Une horloge physique** est un processus physique couplé à une méthode de mesure du temps. La plupart des horloges physiques sont basées sur des processus cycliques basés sur un oscillateur et un compteur. **Une horloge logique** quant à elle, est un moyen d'horodatage qui recense les relations chronologiques et causales dans un système distribué pour établir un classement global des événements de différents processus du système.

Algorithmes de synchronisation d'horloges physique.

Le concept de synchronisation se base sur une source d'horloge physique dont la donnée se propage dans un réseau. Nous allons nous intéresser dans notre étude au protocole NTP, à l'algorithme de Cristian et à l'algorithme de Berkeley.

Le **protocole NTP** utilise un système hiérarchique semi-stratifié de sources de temps se basant sur l'UTC (*coordinated universal time*) comme temps de référence. Comme décrit par N. Minar [20], chaque niveau de cette hiérarchie est appelé *strate* et se voit attribuer un numéro. Un serveur synchronisé sur un serveur de strate n s'exécute à la strate $n + 1$.

L'**algorithme de Cristian** selon [4] est un algorithme de synchronisation d'horloges utilisé pour synchroniser l'horloge locale d'un élément du système distribué avec un serveur de temps externe distant. Un élément du système envoie une demande au serveur temps pour recevoir l'heure actualisée.

L'**algorithme de Berkeley** est décrit par R. Gussella et S. Zatti [10]. Contrairement à l'algorithme de Cristian, le serveur de temps est actif. Il interroge périodiquement chaque élément du système distribué pour lui demander son horloge locale. Sur la base des réponses obtenues, il calcule une durée moyenne et demande à tous les éléments du système d'avancer ou ralentir leurs horloges vers la nouvelle horloge calculée.

L'implémentation dans un SMA du protocole NTP nécessite la mise en place d'une connexion inter-agents respectant une architecture NTP pour l'envoi de la donnée temps et requiert une connectivité continue des agents de la strate $n1$ à une source de temps. Que ce soit pour l'implémentation dans un SMA de l'algorithme de Berkeley ou celui de Cristian, il faut sélectionner un agent du SMA comme étant agent serveur et assurer une connectivité continue de cet agent avec l'ensemble des autres agents pour le déroulement de la synchronisation.

Algorithmes de synchronisation d'horloges logiques.

Le concept d'horloge logique a été initialement proposé par L. Lamport en 1978 [15]. Plusieurs auteurs ont proposé l'utilisation des horloges logiques pour détecter la relation de précedence causale entre les événements (F. Mattern *et al.* [19], J. Fidge [7], R. Schwarz et F. Mattern [25]).

L'**horloge de Lamport** selon [15] démontre que la synchronisation ne doit pas forcément être absolue et qu'elle peut être déduite en se basant sur les relations entre événements. L'ordonnement proposé permet d'attribuer une horloge logique ou estampille à tous les événements d'un système distribué.

Avec l'**horloge de Mattern**, selon F. Mattern *et*

al. [19], chaque élément p du système distribué possède un vecteur d'entiers appelé estampille dans lequel chaque composant $estampille[i]$ est l'estimation par p de la valeur de l'horloge de Lamport du processus i .

Avec les **horloges matricielles** présentées par M. Raynal et M. Singhal [23], chaque processus p d'un système distribué de n processus possède une matrice $n \times n$ d'estampilles dans laquelle chaque composant $estampille[i]$ est l'estimation par p de la valeur de l'horloge de Mattern du processus i .

Pour les 3 approches, une implémentation dans un SMA traduit le principe d'échange entre processus par un échange entre agents. Le ou les estampilles doivent être envoyés lors des interactions inter-agents pour assurer une synchronisation.

Synchronisation d'horloges dans les réseaux de capteurs.

Diverses applications utilisant les réseaux de capteurs sans fil notamment dans la surveillance et la fusion de données, nécessitent que tous les nœuds aient des horloges synchronisées. Or, les méthodes de synchronisation appliquées traditionnellement dans l'industrie ne sont pas forcément adaptées pour une utilisation dans les réseaux de capteurs en raison notamment de problèmes liés à la consommation d'énergie. Par exemple, le protocole NTP, bien que largement utilisé pour la synchronisation d'horloge sur Internet, n'est pas adapté à une utilisation dans les réseaux sans fil, car trop gourmand en énergie. De nouvelles approches adaptées à une utilisation dans ce type de réseau ont ainsi vu le jour ces dernières années.

Protocole de synchronisation de diffusion de référence (RBS) : Ce protocole est basé sur un schéma de synchronisation récepteur-récepteur. Comme décrit par J. Elson *et al.* [6], chaque nœud synchronise son horloge locale avec toutes les autres horloges des nœuds se trouvant dans sa portée de transmission.

Le **protocole de synchronisation temps pour les réseaux de capteurs (TPSN)** est basé sur un schéma de synchronisation émetteur-récepteur. L'algorithme décrit par S. Ganeriwal *et al.* [8] fonctionne en deux étapes. Au cours de la première étape, une structure hiérarchique est établie dans le réseau puis vient une seconde étape de synchronisation durant laquelle les nœuds d'un niveau i sont synchronisés avec les nœuds du niveau $i - 1$ pair par pair.

Le **protocole de synchronisation temps par inondation (FTSP)** décrit par M. Maróti *et al.* [18] met en œuvre une synchronisation dans laquelle le nœud racine transmet périodiquement un message de synchronisation unique aux nœuds qui sont à portée de transmission.

La **mesure du retard de synchronisation du temps pour les réseaux de capteurs sans fil (DMTS)** repose sur une synchronisation émetteur-récepteur dans laquelle l'expéditeur et plusieurs récepteurs sont synchroniser en même temps comme présenté par S. Ping [22].

La **technique de synchronisation temporelle consensuel (CCS)** vise à réduire les écarts d'horloges entre les nœuds proches et fait converger l'ensemble des nœuds vers un écart en commun. L'idée principale de cet algorithme selon M. Maggs *et al.* [17] est de compenser sur plusieurs itérations les écarts d'horloge entre les nœuds du système.

Le **protocole de synchronisation gradient (GTSP)** a été conçu par P. Sommer et W. Roger [26]. Dans ce protocole, les nœuds du réseau diffusent périodiquement un *beacon* de synchronisation avec leurs nœuds voisins. L'horloge logique de chaque nœud converge vers une horloge logique commune.

L'**algorithme reachback firefly (RFA)** cette approche se base sur un réseau d'oscillateurs à couplage d'impulsions où chaque nœud est un oscillateur qui émet périodiquement une impulsion auto-générée. En observant les impulsions des autres oscillateurs, un nœud ajuste légèrement la phase de son propre oscillateur.

Les réseaux de capteurs sans fil pouvant être considéré comme étant des SMA, l'implémentation de ces protocoles dans un SMA reprendra les traitements effectués ou subits au niveau des nœuds pour les appliquer sur les agents du SMA.

Synchronisation utilisant les données.

La **synchronisation des SMA basée sur l'évolution des données (SMASDEV)** selon M. Limame *et al.* [16] s'appuie sur le contenu d'un événement et l'évolution de la donnée qui lui est associée pour trier les événements. Cette nouvelle approche a pour objectif d'instaurer une cohérence globale au niveau d'un SMA en permettant à chaque agent de rétablir un ordre chronologique des données qu'il reçoit des autres agents, sur la base de ses connaissances et sans faire appel à une horloge (physique ou logique). Le prin-

cipe consiste à ce que chaque agent du système enregistre en mémoire sa perception personnelle de l'évolution d'une donnée qu'il recueille avec une fréquence déterminée par rapport à son horloge locale. L'agent peut ainsi prévoir l'évolution future de la donnée. Ainsi, l'agent doit disposer d'un modèle d'évolution en adéquation avec la donnée en question. Pour passer d'une perception personnelle vers une perception globale, chaque agent du système doit, dans un premier temps, demander aux autres agents de lui transmettre les données recueillies puis dans un deuxième temps les positionner par rapport à sa perception personnelle. Ainsi l'agent sera en mesure d'identifier le positionnement de l'ensemble des données reçues par rapport à son horloge. En conséquence, les agents du système seront synchrones et en phase par rapport au modèle d'évolution de la donnée dans le temps.

Selon J. Henriët [11], le **Pèlerin optimiste** est un protocole de gestion de la cohérence des données d'une mémoire partagée répartie de type mémoire répliquée à partage d'objets. Il fonctionne sur la base d'un jeton circulant sur un anneau logique unidirectionnel contenant une structure de données avec les mises à jour des données partagées.

3 Classification des méthodes avec des critères d'évaluation

3.1 Définition des critères

Pour une mesure efficace de la pertinence d'une approche de synchronisation par rapport à une autre, les critères de classification choisis doivent être suffisamment caractérisants. Nous distinguons deux familles de critères.

La famille des critères "*détails de la synchronisation*" permet de qualifier le fonctionnement des approches de synchronisation et le degré de synchronisation obtenu.

La famille des critères "*énergie*" permet de qualifier les besoins des approches de synchronisation notamment en mémoire et puissance pour pouvoir s'exécuter. Les critères de cette famille permettent d'estimer l'énergie requise par l'approche de synchronisation.

Famille des critères "*détails de la synchronisation*" :

Le ***principe de synchronisation*** permet de préciser le mécanisme de synchronisation utilisé par l'approche.

Le ***type de cohérence*** permet de traduire le niveau de synchronisation dans le réseau entre les agents. Dans la littérature il existe plusieurs niveaux de synchronisation, nous en considérons les quatre identifiés par L. Cudennec [5] :

- Le **niveau de cohérence atomique** traduit un degré de synchronisation global stricte au sein de l'ensemble des agents du réseau.
- Le **niveau de cohérence séquentielle** traduit la vision de L. Lamport dont la synchronisation se focalise sur l'ordre des événements et non sur le temps d'occurrence de l'événement.
- Le **niveau de cohérence causale** se base sur la relation de causalité entre les événements survenus dans le réseau d'agents.
- Le **niveau de cohérence relâchée** traduit le degré de synchronisation le plus faible par rapport à ceux cités précédemment.

L'***objet de synchronisation*** précise la nature de ou des éléments faisant objet de synchronisation. La cohérence obtenue à travers la synchronisation est une cohérence temporelle si l'élément synchronisé est une donnée "temps".

Le ***type de synchronisation*** qualifie le processus de synchronisation par rapport à son déclenchement, c'est-à-dire s'il est systématique ou non systématique. Le critère de ***tolérance aux pannes*** permet d'indiquer si l'approche de synchronisation supporte la présence d'une ou plusieurs défaillances au niveau de la flotte d'agents.

Famille des critères "*énergie*" :

Le critère ***calcul réalisé*** permet de caractériser les opérations requises pour pouvoir lancer l'approche de synchronisation.

Le ***volume échangé*** permet de caractériser pour chaque approche de synchronisation le volume requis échangé entre deux agents pour assurer une synchronisation.

Le ***nombre de messages requis pour synchroniser une donnée entre 2 agents*** recense le nombre de messages requis échangés par deux agents pour assurer une synchronisation.

Le ***nombre de messages requis pour synchroniser une donnée entre N agents*** recense le nombre de messages requis échangés par un agent avec les autres agents pour assurer la synchronisation d'une donnée. Le nombre de messages échangés permet de quantifier l'énergie consommée pour l'exécution de l'approche de synchronisation au niveau de la flotte d'agents.

3.2 Classification des approches de synchronisation étudiées

Notre classification est présentée dans le tableau 1. Cinq des quinze approches considérées dans cet article supportent par défaut la tolérance aux pannes. Même si le reste des approches ne présente pas cette faculté, elle peut être mise en place lors de leur implémentation. Qu'il s'agisse des approches de synchronisation se basant sur un échange d'horloges ou de données, la topologie du réseau ne constitue pas un frein pour le lancement de l'approche de synchronisation. En effet, dans treize des quinze approches étudiées, les agents doivent pouvoir communiquer les uns avec les autres indépendamment de la topologie. Toutes les approches étudiées se basant sur les horloges physiques ou logiques assurent une synchronisation entre deux éléments à travers un envoi d'un à trois messages. Dans les approches utilisées dans les réseaux de capteurs sans fil, une synchronisation entre deux nœuds peut être réalisée avec un ou deux messages. Nous constatons que pour les approches RFA et DMTS, le nombre de messages envoyés pour la synchronisation globale d'un système multi-agents ne varie pas en augmentant ou en diminuant le nombre d'agents. En effet, l'émetteur envoie un seul message en mode diffusion vers l'ensemble des nœuds du réseau constitué d'agents.

Toutes les approches de synchronisation à l'exception de celles qui se basent sur les données nécessitent un espace mémoire relativement faible pour instaurer une cohérence temporelle. Le besoin en stockage pour les approches se basant sur les données dépend de la taille de l'objet de synchronisation et du résultat de la fonction utilisée pour la synchronisation. Parmi les approches étudiées, seul le *Pèlerin optimiste* permet d'assurer une cohérence atomique. Neuf des quinze approches étudiées, indépendamment de leurs domaines d'application, permettent d'effectuer une synchronisation de données en plus de la synchronisation de la notion temps (estampille, horloge,...). L'ensemble des approches de synchronisation étudiées se basant sur des échanges d'horloges physiques et logiques, correspondent à des approches à déclenchement systématique à l'occurrence d'événements. Par ailleurs, dans les réseaux de capteur sans fil, parmi les sept approches considérées, les approches RBS, TPSN et DMTS ne mettent pas en œuvre un déclenchement systématique.

4 Analyse et interprétations des résultats de classification

Cette classification constitue une aide à l'identification de la technique de synchronisation la plus adaptée à des contraintes d'application de SMA. Dans le cas d'agents ayant des ressources limitées, leur capacité de calcul peut être elle aussi limitée, l'espace mémoire réduit et les échanges de messages peuvent être restreints. D'une manière générale, plus les besoins de l'approche de synchronisation sont réduits, plus la durée d'application de l'approche est importante. Ce détail peut s'avérer crucial notamment dans les cas d'application sur des trajets et parcours comme lors de missions de cartographie menées par plusieurs agents. Cette interprétation est affinée dans la partie suivante ressource par ressource.

Concernant l'espace mémoire, l'approche de synchronisation se basant sur les Horloges matricielles se distingue par un besoin en espace mémoire important. En effet, le stockage de l'ensemble des matrices associées à chaque événement (local ou reçu) est coûteux. De même, à un degré moins important, la synchronisation utilisant les Horloges de Mattern a un besoin en espace mémoire relativement coûteux. L'utilisation de ces approches peut avoir un impact négatif sur la robustesse du système. A contrario, les approches DMTS, NTP, TPSN et l'algorithme de Cristian se caractérisent par un besoin relativement faible en espace de mémoire. Le besoin consiste essentiellement à stocker au niveau de l'agent récepteur l'horloge ou l'estampille temps à synchroniser. L'Horloge de Lamport, l'Algorithme de Berkeley, RBS, FTSP, CCS et le protocole du Pèlerin optimiste requièrent une faible capacité de calcul réduite à la puissance nécessaire au calcul d'une moyenne de temps, de l'offset ou de l'écriture sur un jeton.

Les algorithmes de Cristian et de Berkeley sont deux approches adaptées à des SMA dans lesquelles il est possible de gérer le temps de manière centralisée.

Sur le plan de la consommation d'énergie, c'est l'approche de synchronisation CCS qui nécessite le plus grand nombre de messages car en effet, l'envoi de messages à l'ensemble des agents du réseau se fait sur plusieurs itérations jusqu'à la suppression des écarts. Ceci peut s'avérer un facteur direct d'épuisement de l'énergie de l'agent. De plus, cette synchronisation est systématique la rendant d'autant plus coûteuse en énergie. De

TABLE 1 – " Classification des approches de synchronisation "

libellé	Famille des critères "détails de la synchronisation"					Famille des critères "énergie"			
	principe de synch.	type de cohérence	objet de synch.	Syncho systématique	tolérance aux pannes	Calcul fait/éché	Volume échangé	nbr msg synch 2 agents	nbr msg synch n agents
Algorithmes de synchronisation d'horloges physiques									
Protocole NTP	partage d'une estampille	R	estampille	X	X	op soustraction + op division	une estampille	2	n-m+1 n niveau récepteur , m niveau émetteur
Algo de Cristian	à travers un serveur temps	S	horloge	X		op addition + op division	Round-trip + horloge serveur temps	2	2n
Algo Berkeley	à travers un serveur temps	R	Offset	X	X	op addition + op division	serveur : n horloges	3	3n
Algorithmes de synchronisation d'horloges logiques									
Horloges Lamport	partage d'une estampille	C	donnée + estampille	X		op comparaison + op incrémentation	estampille locale et synch	1	n-1
Horloges de Mattern	partage d'un vec. estampilles	S	vecteur d'estampilles	X		n-1 op comparaison + op incrémentation estampille	vecteur d'estampilles locales et synch	1	n-1
Horloges matricielles	partage d'une mat. estampilles	S	matrice d'estampilles	X		n-1 op comparaison + op incrémentation estampille	matrices d'estampilles locales et synch	1	n-1
Synchronisation d'horloges dans les réseaux de capteurs									
RBS	partage d'horloge	R	émetteur : tt donnée ; récepteur : horloge			op addition + op division + calcul offset	donnée partagée + k horloges(k nbr voisins)	2	1+(n-1)* k k : nbr noeux voisins aux (n-1) noeux ayant reçu l'horloge
TPSN	partage d'une estampille	R	paquet de synch.			mise en arborescence + calcul nouvelle horloge	une estampille	2	n-m+1 avec n niveau récepteur et m niveau émetteur
FTSP	partage d'horloge	R	ID + estampille racine + donnée	X	X	calcul offset	estampille + donnée partagée	2	1+k : 1msg émetteur + 1msg récepteurs vers voisinage
DMTS	partage d'une estampille	R	estampille racine			calcul nouvelle horloge	une estampille	1	de l'émetteur vers récepteurs
CCS	partage d'une estampille	R	estampille et taux d'écart	X	X	calcul taux d'écart	estampille + taux d'écart	2 * nbr itérations	2*(n-1) * nbr itérations
GTSP	partage d'horloge	R	balise de synch et fréq d'horloge	X	X	mise à jour fréq d'horloge + calcul nouvelle horloge	horloge + fréquence d'horloge	2	2(n-1)
RFA	partage d'un état	S	vecteur d'état	X		calcul vecteur d'état et de la fonction objectif	packet synchronisation vecteur d'état	1	de l'émetteur vers récepteurs
Synchronisation utilisant les données									
Smasdev	assimilation de données	C	donnée ayant 1 modèle d'évol			manipulation donnée + simulation modèle d'évolution	tuplet de données partagées	1	n-1
Pélerin optimiste	Réservation ressources et diffusion via jeton	A	Données partagées			manipulation jeton	données partagées	6	3n+1

même, les protocoles NTP, FTSP, l'algorithme de Cristian, l'algorithme de Berkeley étant exécutés d'une façon régulière et périodique, ils risquent de consommer beaucoup d'énergie si la fréquence de lancement est importante. Quant aux approches qui se lancent à l'occurrence d'un événement ou l'une de leurs évolutions (RFA ou Horloges de Mattern par exemple), la consommation d'énergie augmente de manière modérée lors de chaque lancement ce qui est en adéquation avec une utilisation dans un système multi-agents à ressources et connectivité limitées. La consommation d'énergie pour les approches Kalman et Smasdev pour lesquelles la synchronisation n'est pas systématique, reste relativement faible et augmente proportionnellement avec la complexité du modèle d'évolution de la donnée partagée et la densité de répartition du réseau d'agents.

Bien que les sept approches étudiées et utilisées dans les réseaux de capteurs sans fil puissent être appliquées dans les SMA, il serait souhaitable de limiter au maximum la fréquence de leur exécution dans un système dans lequel les agents ont des ressources limitées.

En se basant sur l'approche de Lamport, la comparaison des valeurs des horloges ne permet pas de déduire une relation de causalité entre deux événements a et b . En effet, si $H(a) < H(b)$ ne signifie pas forcément que $a \Rightarrow b$. De plus, en cas d'affluence, des messages peuvent ne pas être reçus. Au regard de ces limites, nous proposons de l'écarter de la liste des approches retenues.

Le choix de l'approche de synchronisation peut aussi être déterminé en fonction de la nature de la donnée partagée. Par exemple, si la donnée partagée porte uniquement sur la notion de temps (estampille, matrice temps, ..), le choix peut être l'une des approches TPSN, DMTS, CCS ou GTSP conçues pour instaurer une synchronisation temporelle et en adéquation avec le type de système multi-agents étudié. Par contre si la donnée partagée ne porte pas sur la notion temps, nous proposons l'utilisation des approches RFA, Smasdev, RBS ou FTSP qui sont en adéquation avec le systèmes multi-agents étudié et qui sont triées selon l'ordre de degré de cohérence du plus strict vers le moins strict. Nous écartons de cette liste le Pèlerin optimiste nécessitant que tous les agents puissent communiquer continuellement et donc peu compatible avec un SMA constitué d'agents faiblement accointants.

Quoiqu'il en soit, les protocoles nécessitant le moins de ressources sont à privilégier pour

les SMA constitués d'agents à ressources et connectivités limitées. Au-delà de ces considérations, l'approche de synchronisation à choisir dépend fortement du contexte de synchronisation et du niveau de synchronisation recherché. Par exemple, pour répondre à une mission de supervision des données de température sur une zone à risques volcanique, l'approche smasdev semble convenir parfaitement. Nous allons dans ce qui suit effectuer une simulation sous la plateforme Gama de l'approche smasdev afin de vérifier les comportements obtenus.

5 smasdev : simulation et interprétation des résultats

5.1 Détails de simulation

Pour répondre à la mission de supervision des données de température sur une zone à risques volcaniques, nous allons tout d'abord implémenter un modèle agent doté d'une fonctionnalité de synchronisation basée sur smasdev. Pour que la simulation soit proche des conditions réelles d'exécution, nous avons ajouté les paramètres suivants :

Le **taux de connectivité** permet de prendre en considération les problèmes de connectivité qui peuvent survenir durant une mission. Elle est entre 0 et 1.

La **déviations d'horloge** permet de prévoir les déviations en millisecondes que peut avoir une horloge notamment à cause des conditions climatiques extrêmes.

La **période de prise de mesures** permet de renseigner l'intervalle de temps en millisecondes entre deux prises de températures et ceci pour l'ensemble des agents durant leur mission.

En plus de ces paramètres, nous allons aussi faire varier le nombre d'agents mandatés pour la mission afin d'analyser l'impact de la taille de la flotte sur les résultats de synchronisation.

Pour tester le comportement de l'approche smasdev, nous allons utiliser 5 instances de simulations sous la plateforme gama dans sa version 1.8. Le SMA étudié étant faiblement connecté, le paramètre taux de connectivité sera positionné à 10%. La mission se déroulant sur une zone volcanique à haute température, le paramètre déviation d'horloge sera positionné à 550ms (0.55s). Le dernier paramètre fixant l'intervalle de prise de mesure sera positionné à 50ms. Nous varions le nombre d'agents à chaque simulation.

Les données utilisées sont représentées dans le tableau ci-après :

	sim 1	sim 2	sim 3	sim 4	sim 5
nbr agents	3	5	10	15	20
Autres paramètres pour toutes les simulations :					
Déviaton horloge					0.55s
Période prise de mesure					50ms
Taux de connectivité					10%

Nous utilisons, lors de la simulation, un échantillon de 190 données de température qui seront réparties aléatoirement entre les agents de la flotte en respectant l'intervalle de prise de mesure. Pour des raisons de simplification nous allons considérer que le modèle d'évolution de ces données suit une fonction de non régression linéaire de forme $y : ax+b$.

Lorsqu'un agent est connecté à un autre, il identifie, parmi les données de températures de l'autre agent, celles qui ne se trouvent pas dans sa base de données et lance ensuite l'approche de synchronisation smasdev pour les estampiller selon son horloge local. Le nombre de fois où un agent se trouve connecté à un autre dépend du paramètre "taux de connectivité".

La simulation prend fin au bout de 1500 pas d'horloge.

5.2 Interprétation des résultats de simulation

Une donnée correctement synchronisée par un agent est une donnée dont le positionnement par rapport à ses autres données recueillies est en adéquation avec le modèle d'évolution de la donnée.

Bien que le taux de connectivité soit faible, nous avons obtenu, lors des 5 simulations, un pourcentage de bon positionnement à 100% pour chaque agent. Le pourcentage est défini par le ratio du nombre de données correctement positionnées sur le nombre total de données synchronisées par l'agent. Smasdev est bien adapté à notre SMA étudié par rapport à la contrainte de connectivité.

Après avoir testé smasdev à une petite échelle, nous souhaitons valider l'approche smasdev avec un plan d'expérimentation plus riche et plus conséquent à travers une simulation en masse. Ceci peut être réalisé avec l'utilisation de la bibliothèque gamar.

6 Conclusion

Cet article présente une étude de différentes solutions envisageables à la problématique de la synchronisation temporelle des données partagées entre agents faiblement accointants et à ressources limitées.

La démarche consiste à partir des solutions existantes dans l'univers des systèmes distribués pour identifier celles qui sont en adéquation avec les contraintes du système multi-agents étudié notamment celle de la limitation des ressources.

Nous avons effectué une simulation de l'approche smasdev qui fait partie des approches étudiées pour vérifier sa compatibilité avec notre SMA étudié du point de vue de la connectivité.

Au regard de cette étude et des résultats de simulations, nous envisageons maintenant de valider que l'approche smasdev est complètement compatible avec notre SMA étudié en tenant compte de toutes les contraintes notamment celles liées à la consommation d'énergie. Pour cela nous allons effectuer une simulation à grande échelle tenant en compte de l'ensemble des paramètres pouvant influencer le comportement de smasdev.

Références

- [1] Laurent Cauffriez and Jean Defrenne. Viabilité de l'information et réseau à diffusion. *Diagnostic et sûreté de fonctionnement*, Vol, 5(2) :219–247, 1995.
- [2] Brahim Chaib-draa. Agents et systèmes multi-agents. 1999.
- [3] Robert T Craig and Karen Tracy. *Conversational coherence : Form, structure, and strategy*, volume 2. SAGE Publications, Incorporated, 1983.
- [4] Flaviu Cristian. Probabilistic clock synchronization. *Distributed computing*, 3 :146–158, 1989.
- [5] Loïc Cudennec. Modèles et protocoles de cohérence des données en environnement volatil. 2005.
- [6] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI) :147–163, 2002.
- [7] Colin J Fidge. Timestamps in message-passing systems that preserve the partial ordering. 1987.

- [8] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 138–149, 2003.
- [9] Barbara Grosz and Candace L Sidner. Attention, intentions, and the structure of discourse. *Computational linguistics*, 1986.
- [10] Riccardo Gusella and Stefano Zatti. The accuracy of the clock synchronization achieved by tempo in berkeley unix 4.3 bsd. *IEEE transactions on Software Engineering*, 15(7) :847–853, 1989.
- [11] Julien Henriët. *Evaluation, optimisation et validation de protocoles distribués de gestion de la concurrence pour les interactions coopératives*. PhD thesis, 2005.
- [12] FERBER Jacques. Les systèmes multi-agents : vers une intelligence collective. *InterEditions, Paris*, 322, 1995.
- [13] Sablatnig Jan, Grottké Sven, Kopke Andreas, Chen Jiehua, Seiler Ruedi, and Wolisz Adam. Consistency in distributed systems. 2007.
- [14] Thomesse Jean-Pierre. Le réseau de terrain fip. (3) :287–321, 1993.
- [15] Leslie Lamport. Time, clocks and the ordering of events in a distributed system. *Communications of the ACM*, 1978.
- [16] Mohamed Limame, Julien Henriët, Christophe Lang, and Nicolas Marilleau. Synchronisation d’horloge dans un système multi-agents. In *APIA*, 2019.
- [17] Michael Kevin Maggs, Steven G O’keefe, and David Victor Thiel. Consensus clock synchronization for wireless sensor networks. *IEEE sensors Journal*, 12(6) :2269–2277, 2012.
- [18] Miklós Maróti, Branislav Kusy, Gyula Simon, and Akos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 39–49, 2004.
- [19] Friedemann Mattern et al. *Virtual time and global states of distributed systems*. Univ., Department of Computer Science, 1988.
- [20] Minar Nelson. A survey of the ntp network, 1999.
- [21] Martin Olsson, Kjell Hansson, Ann-Marie Lundblad, and Marianne Cederblad. Sense of coherence : definition and explanation. *International Journal of Social Welfare*, 15(3) :219–229, 2006.
- [22] Su Ping. Delay measurement time synchronization for wireless sensor networks. *Intel Research Berkeley Lab*, 6 :1–10, 2003.
- [23] Michel Raynal and Mukesh Singhal. Logical time : Capturing causality in distributed systems. *Computer*, 29(2) :49–56, 1996.
- [24] J. Müller S. Bussmann. A negotiation framework for cooperating agents. *Proceedings of CKBS-SIG (CKBS’92)*, 1992.
- [25] Reinhard Schwarz and Friedemann Mattern. Detecting causal relationships in distributed computations : In search of the holy grail. *Distributed computing*, 7, 1994.
- [26] Philipp Sommer and Roger Wattenhofer. Gradient clock synchronization in wireless sensor networks. In *2009 International Conference on Information Processing in Sensor Networks*, pages 37–48. IEEE, 2009.
- [27] DJW Strümpfer, JF Gouws, and MR Viviers. Antonovsky’s sense of coherence scale related to negative and positive affectivity. *European Journal of Personality*, 12(6) :457–480, 1998.
- [28] Andrew Tanenbaum. *systèmes centralisés systèmes distribués*. 2ème édition Dunod, ISBN-10 : 210004554, 1998.
- [29] Paul Thagard. *Coherence in thought and action*. MIT press, 2000.
- [30] Paul Thagard. *Hot thought : Mechanisms and applications of emotional cognition*. MIT press, 2008.
- [31] Quang-Anh Nguyen Vu. *Cohérence et robustesse dans un système multiagent perturbé : application à un système décentralisé de collecte d’information distribué*. PhD thesis, Université Claude Bernard-Lyon I, 2012.

Outils et démonstrations logiciels

ESCAPE: Simulation à base d'agents pour l'évacuation de populations lors des situations d'urgence

Olivier Gillet^a Eric Daudé^a Arnaud Saval^b Clément Caron^a
olivier.gillet@univ-rouen.fr eric.daude@cnr.fr arnaud.saval@univ-rouen.fr caron.clement@gmail.com

Sébastien Rey-Coyrehourcq^a Patrick Taillandier^c Pierrick Tranouez^b
sebastien.rey-coyrehourcq@univ-rouen.fr patrick.taillandier@inrae.fr pierrick.tranouez@univ-rouen.fr

^aCNRS, Normandie Université, UMR 6266 IDEES

^bUniversité de Rouen Normandie, EA LITIS

^cUniversité de Toulouse, INRAE, UR MIAT

Résumé

L'évacuation des populations face aux risques naturels ou technologiques est un défi complexe qui requiert une planification et des exercices de préparation pour identifier les éventuels verrous et dangers. La simulation informatique, comme les modèles multiagents, est une alternative aux exercices grandeur nature pour tester divers scénarios tout en évitant les contraintes associées à ces exercices. Le projet ESCAPE a pour objectif de développer un corpus d'outils d'aide à la modélisation et à la simulation d'évacuation d'un territoire. L'analyse des scénarios simulés permet alors de comprendre la complexité d'une évacuation et sa mise en œuvre pratique. Cet article présente l'environnement ESCAPE, de la construction de l'environnement de simulation à la production d'indicateurs statistiques.

Mots-clés : évacuation, simulation à base d'agents, gestion de crise, risques naturels

Abstract

Mass evacuation from natural and technological hazards is a complex challenge that requires planning. Exercises must be used to prepare crisis managers and population, and they can also highlight the possible obstacles and dangers of an evacuation. Computer simulation, and in particular agent-based models, is then used as an alternative to these exercises because of its capacity to reproduce situations close to real systems and its ability to test a wide range of scenarios, while avoiding the constraints associated with full-scale exercises. The ESCAPE project aims to develop a set of tools to help model and simulate the evacuation of a territory. The analysis of the simulated scenarios allows to understand the complexity of such a strategy and its op-

rationnalization. From simulation environment construction to the production of statistical indicators, this article presents the entire ESCAPE environment.

Keywords: evacuation, agent based simulation, crisis management, natural hazards

1 Introduction

Les aléas naturels et technologiques peuvent être à l'origine des crises qui constituent des menaces potentielles pour les biens, l'environnement et les personnes qui y sont exposées. L'expression de ces aléas est très variable en intensité, en fréquence et en probabilité d'occurrence. Elle peut être isolée, combinée ou en séquence [7]. Or, la vulnérabilité des territoires dépend à la fois des enjeux exposés et de la capacité à anticiper la nature et l'importance des dangers qui les menacent. Les risques se définissent comme une combinaison localisée d'un aléa et d'enjeux exposés. La réduction des risques porte à la fois sur une meilleure connaissance des aléas (surveillance, prédiction) et sur la réduction des vulnérabilités des enjeux exposés (mesures de mitigation, culture du risque)[16]. Selon la localisation de l'aléa et sa nature, l'évacuation massive de la population peut être une mesure phare de la gestion de crise. Cette stratégie peut s'avérer efficace pour soustraire toute ou une partie de la population d'une menace réelle mais peut entraîner des conséquences catastrophiques en cas de manque de préparation [3]. La préparation concerne à la fois les acteurs de la gestion de crise et la population. Les premiers doivent identifier les zones potentiellement exposées, connaître le nombre de personnes exposées ainsi que leurs vulnérabilités, préparer des centres d'accueil,

des moyens de transports collectifs et identifier des itinéraires d'évacuation. Les seconds doivent être conscients des dangers, connaître les consignes de sécurité et les systèmes d'alerte, être prêts à suivre les recommandations et à faire les choix qui s'imposent en contexte de crise.

Dans nombre de territoires soumis aux aléas naturels, la préparation des autorités et des populations s'est acquise au fil des crises passées. Une catastrophe majeure a dans bien des cas été l'élément déclencheur d'une prise de conscience de la nécessaire préparation à la stratégie de l'évacuation. C'est le cas du super cyclone qui a touché le Bangladesh en 1991 et causé la mort de plus de 140 000 personnes [10]. Cette tragédie a conduit l'Inde à inscrire dans la loi la planification et la gestion de crise. Le pays conduit chaque année, notamment pendant la période des cyclones, des évacuations massives de plusieurs milliers voire millions de personnes en limitant ainsi le nombre de décès [11]. Mais de nombreux territoires exposés à la menace d'un aléa n'ont heureusement pas encore connu de catastrophe, et la préparation à l'évacuation massive n'est pas encore une doctrine de gestion de crise pour les autorités. Et même lorsque celle-ci est inscrite dans des plans, l'opérationnalisation de ceux-ci en l'absence de crises ou d'exercices laisse de nombreuses zones d'ombre sur son efficacité et sur les réactions des acteurs en cas de déclenchement. C'est dans ce contexte que la modélisation et la simulation spatiale peuvent être un outil intéressant pour explorer les dynamiques possibles d'un tel événement.

Daudé et al., (2019) [8] propose une ontologie du système ESCAPE. Ce cadre conceptuel s'articule autour de quatre modules principaux : (1) L'environnement ; (2) L'aléa ; (3) Processus de prise de décision des agents ; (4) La gestion des situations d'urgence. Cet article veut présenter l'opérationnalité du projet ESCAPE et de la généralité des outils développés [8]. Cette suite d'outils a pour objectif d'identifier des verrous thématiques dans un territoire dont la population pourrait être amenée à évacuer. Une première série d'outils mobilise les méthodes des systèmes d'information géographiques pour décrire l'environnement de simulation. L'ensemble de ces données est compilé et utilisé dans un modèle de simulation qui permet de spécifier les dynamiques du système. Un ensemble de modèles permet alors de simuler les dynamiques spatiales (modèles de déplacement individuels et collectifs) et temporelles (injection d'événements). ESCAPE permet ainsi d'explorer de

nombreuses thématiques à partir de différents scénarios, qu'il s'agisse de l'évolution des conditions de trafic, de la capacité de réponse des populations ou encore des itinéraires d'évacuation. Selon différents plans d'expériences prédéfinis et stratégies d'exploration de modèle, de nombreuses sorties standardisées sont enfin traitées dans une dernière suite d'outils utilisant de nouveau les méthodes des systèmes d'information géographique et de statistique spatiale. Une application web de visualisation de données hébergées par l'infrastructure de recherche Humanités Numériques ¹ permet d'explorer les résultats des simulations de manière intuitive afin de poursuivre les investigations et/ou de discuter avec les opérateurs de gestion de crise sur les connaissances produites par le système.

En continuité avec la partie introductive, la section 1 présente comment le système ESCAPE se positionne vis-à-vis des travaux existant dans le domaine de la simulation informatique des évacuations massives. La Section 2.1 présente en détails le système ESCAPE de la génération des inputs à l'analyse des sorties statistiques des simulations. Enfin, la Section 3.4 propose deux cas d'application (l'un sur le risque d'inondation à Saumur, l'autre sur le risque volcanique en Guadeloupe) pour illustrer l'outil ESCAPE.

2 Contexte

2.1 Modéliser les évacuations massives

La modélisation informatique des évacuations massive de la population est un sujet de recherche complexe. La modélisation nécessite une approche intégrée qui doit permettre de coupler des données spatiales hétérogènes. Elles incluent des données de population, les réseaux de transport ou encore les circuits de sirènes mobiles. Elle doit également intégrer les scénarios probables d'évolution de l'aléa et être en capacité de reproduire par la simulation la variété des comportements individuels et collectifs qui pourraient être observés dans des situations analogues. Cette chaîne de traitement doit fournir des outils d'exploration spécifiques à l'évacuation, telle que la localisation optimale de centres de refuge ou des itinéraires de collecte des personnes non autonomes.

Les modèles spatialisés de simulation à base d'agents ont investi le champ de la recherche

1. Site web de l'infrastructure de recherche Humanités Numériques <https://www.huma-num.fr/>

sur les évacuations massives de populations [5]. Ceux-ci sont dans la plupart des cas décomposés en deux sous-systèmes qui interagissent, celui des humains qui décrit les enjeux et celui du système physique qui décrit l'aléa. Le premier est composé d'individus qui prennent des décisions en fonction de descripteurs individuels plus ou moins hétérogènes au sein de la population et de normes collectives qui représentent les règles de l'institution à laquelle ils appartiennent. Le second sous-système représente le comportement du système physique modélisé, en lien avec les enjeux environnementaux qui peuvent être affectés. Le système complet repose alors sur les interactions à de multiples échelles spatiales, temporelles et organisationnelles, avec des comportements régis par des informations imparfaites et des logiques qui peuvent être contradictoires entre les acteurs. La modélisation des déplacements est ainsi au cœur de ces modèles et repose pour la plupart sur des développements ad hoc. Même s'il existe des outils de modélisation de trafic reconnus (MATSim, TRANSIMS, SUMO, SimMobility) qui permettent de modéliser le trafic urbain au jour le jour, la plupart d'entre eux présentent des limites lorsqu'ils sont utilisés pour étudier les perturbations induites par des événements extrêmes [6]. Pour dépasser cette contrainte, des extensions ont été proposées afin d'adapter ces cadres à l'étude de l'évacuation de masse : MATSim fournit une extension dédiée à l'étude de l'évacuation [14]. D'autres travaux utilisent ce modèle [12, 25] pour simuler l'évacuation en cas de tsunami ou d'ouragan avec une simplification de la modélisation.

Puisqu'il est extrêmement difficile de réaliser des exercices grandeur nature pour préparer les autorités et la population, ESCAPE propose un système de modélisation et d'exploration de simulation d'évacuation générique et relativement simple à prendre en main. ESCAPE tente également de répondre à certains verrous liés à la simulation informatique des évacuations massives tels que la non-intégration des modules spécifiques pour modéliser une organisation responsable de gestion de crise. La validation de ces simulations est bien souvent difficile [24]. Les raisons sont multiples : données manquantes, incertitudes épistémiques, etc, Cependant il est indispensable de calibrer et de valider les modèles développés. Pour faire face à cette difficulté, certains sont calibrés via des données terrains (e.g calibration des vitesses de déplacements via des enregistrements GPS, définition des agendas synthétiques via des Enquêtes Ménages - Déplacements, définition des comporte-

ments réflexes via des enquêtes terrains). Puis les modèles sont soumis à des experts pour évaluer la cohérence des concepts et des sorties par rapport à la question de modélisation posée.

3 ESCAPE : un outil générique et complet

L'ontologie conceptualisée dans le cadre du projet ESCAPE décompose le système en quatre composantes : l'environnement, les aléas, le processus de prise de décision des individus et la gestion de crise par les autorités [8]. Le système ainsi envisagé est donc intégratif ; il envisage la combinaison et les interactions entre des composants de différentes natures allant des théories du comportement humain en passant par les dynamiques géophysiques des aléas, tout en mobilisant un cadre méthodologique informatique robuste afin de proposer un outil fiable et opérationnel. En se fondant sur des cas d'application réels, le modèle a été élaboré selon une approche analytique en coordination et concertation avec divers acteurs et experts. La chaîne de traitements nécessaires à l'évaluation d'un tel processus requiert une attention particulière dans la construction de l'environnement de simulation ainsi que la définition de scénarios d'aléas et d'évacuation afin de proposer une modèle réaliste, robuste et calibré.

3.1 Construction de l'environnement de simulation via R-Shiny

À l'heure de l'open data, il est relativement aisé de construire un environnement réaliste via des données géographiques libres d'accès. Le système a été spécialement conçu pour utiliser les données issues d'OSM. Malgré les limites bien connues de ces données, c'est la plus grande base de données spatialisées libres du monde. Il est donc possible à partir de ces données de modéliser un environnement dans lequel vont se mouvoir les agents - piétons et automobilistes. Les outils ESCAPE permettent de modéliser, à partir d'un fichier OSM XML, un territoire et ses infrastructures telles que le réseau routier ou les réseaux de transports en commun si la donnée est renseignée. Tous les éléments de l'environnement sont "agentifiés". Les réseaux sont décomposés en nœuds et tronçons où chacun a ses méthodes et attributs [21]. Il en va de même pour les bâtiments qui possèdent un certain nombre d'attributs comme le nombre d'individus ou de ménages s'il s'agit d'un habitat collectif. La défi-

inition de cet attribut suppose donc de générer et de ventiler une population synthétique sur l'ensemble de la zone d'étude. Cela fait référence à la génération d'une population synthétique qui simule les caractéristiques d'une population réelle lorsque les données réelles sont insuffisantes ou inaccessibles [17]. Les méthodes de génération de population synthétique sont nombreuses (avec ou sans échantillon, déterministe ou stochastique) et bien souvent fonction des données disponibles et du niveau de détails que l'on souhaite intégrer dans le système [5, 9]. Dans le système, chaque agent "humain" est doté d'attributs socio-démographiques, appartient à un ménage et possède un lieu de résidence. La génération d'une population synthétique peut être réalisée à l'aide de diverses techniques, mais l'algorithme le plus utilisé est l'IPF (Iterative Proportional Fitting) [22, 5, ?, 9]. Cette méthode de reconstruction avec échantillon a été utilisée pour l'un des cas d'application, la Guadeloupe. Les recensements de la population en France fournissent, via la base de données "Individus localisés au canton-ou-ville" [?], des données très précises sur les caractéristiques éco-socio-démographiques d'un échantillon représentatif de la population. D'autres données telles que les données carroyées à 200m de l'INSEE, les Enquêtes Ménages - Déplacements (EMD) ou les enquêtes de fréquentation peuvent être mobilisées dans les mêmes buts de générer une population ou d'apporter des précisions sur les mobilités quotidiennes des individus et ménages. Ces dernières incluent non seulement des informations sur les résidents permanents, mais aussi sur les individus qui entrent régulièrement (migration pendulaire) ou de manière ponctuelle, comme les touristes, dans les zones exposées. La population est ensuite ventilée sur le territoire selon les caractéristiques du bâti (fonction, superficie, nombre d'étages) [20].

La conception d'un environnement de simulation passe également par l'intégration d'une dynamique spatio-temporelle réaliste des aléas. Les sorties des modèles numériques modélisant la géodynamique d'aléas naturels comme technologiques constituent les entrées du système ESCAPE. Statique ou dynamique, l'agent "aléa" peut impacter les autres agents du système et modifier l'environnement de simulation. Il convient donc de définir au préalable les règles d'exposition à l'aléa des autres agents. Par exemple, l'exposition à un écoulement pyroclastique lors d'une crise éruptive, aussi brève soit-elle, est fatale alors que les conséquences directes des retombées de cendres volcaniques sont moindres

[2]. Concernant l'aléa inondation, la hauteur et la vitesse d'eau peuvent également avoir un impact plus ou moins important sur les capacités de déplacement des agents. L'intégration des aléas et ses interactions avec les autres agents n'est pas automatique. La grande diversité d'aléas et des gradients d'intensité compliquent le développement d'un outil générique. Il est nécessaire de programmer en GAML (GAMA Modeling Language) le processus de diffusion et d'interaction de l'aléa. Le GAML est le langage "haut niveau" orienté "objet-agent" de la plateforme de simulation GAMA. Il a été développé pour être utilisé par des non-informaticiens [23]. Il s'agit ici de définir un agent "aléa" et ses interactions avec les autres agents.

En plus de caractériser le territoire, les enjeux exposés, l'aléa et ses dynamiques possibles, il est possible de caractériser les outils de gestion de crise lorsqu'ils existent. Il s'agit par exemple de localiser des zones de refuges, des itinéraires d'évacuation ou bien encore des circuits de transport collectif pour évacuer les personnes non-autonomes. Une application Web a été développée en R via le package « shiny »[4] pour générer, sur les bases d'un savoir expert, de nombreuses réponses d'urgence conformes au formalisme du système ESCAPE. La figure 1 présente le visualiseur dynamique et interactif de l'application Web. Cette interface permet, aux experts ou non, d'éditer et de spatialiser diverses mesures telles que les points de rassemblement ou les itinéraires d'évacuation. L'application génère ensuite une archive contenant une liste de shapefiles (EPSG : :4326) pouvant être utilisés dans GAMA par l'extension logicielle ESCAPE.

3.2 Une extension logicielle de GAMA

Le système ESCAPE est une série d'outils qui permet de simuler l'évacuation de la population d'un territoire à des échelles temporelles et géographiques fines. Une attention particulière a été portée sur la modélisation des déplacements. Si de nombreux modèles dédiés à simuler des évacuations existent, peu sont capables de prendre en compte la réalité d'une évacuation massive de la population qui impliquerait des déplacements multi-modaux (à pied, en deux roues, en voiture privée, par les transports publics). Ainsi, si ESCAPE a été développé via la plateforme de simulation à base d'agents open-source GAMA [23, 5], il intègre en plus des extensions permettant de prendre en compte de façon fine ces différents types de mobilités [21]. La plateforme permet également d'exporter les données brutes

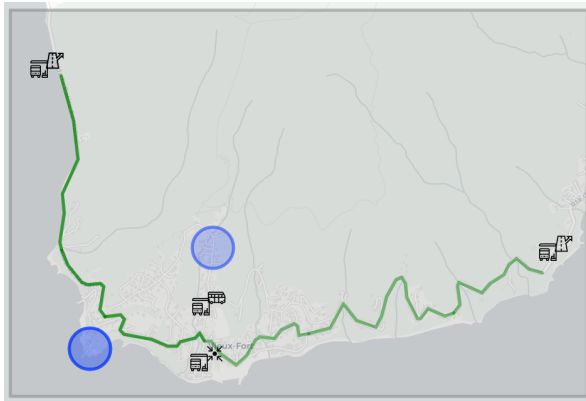


FIGURE 1 – Viewer dynamique et interactif de l'application R-Shiny. Les pictogrammes indiquent la localisation des points de rassemblement et des points de fuite. Le linéaire vert correspond aux itinéraires d'évacuation. Les cercles bleus indiquent l'emplacement des sirènes. Il s'agit d'un exemple entièrement fictif sur la commune de Vieux-Fort, Guadeloupe.

issues des simulations mais aussi des résumés statistiques. Le système se veut précis et exhaustif dans la récolte d'information sur les agents et leurs actions au cours des simulations.

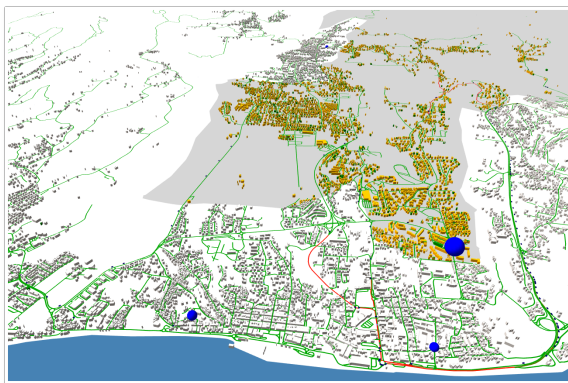


FIGURE 2 – Interface graphique de la plateforme de simulation GAMA. La figure présente l'évacuation polyphasée du sud de la région de Basse-Terre. Les axes routiers embouteillés sont en rouge, les sphères bleues représentent les points de rassemblement et la zone grisée est la zone à évacuer.

L'environnement de développement GAMA offre une interface de visualisation graphique des simulations en 2D ou 3D. La figure 2 présente une situation d'évacuation polyphasée des communes de Basse-Terre et Saint-Claude en Guadeloupe. Cette stratégie consiste à définir un

séquençage des départs afin de faciliter l'évacuation des populations et l'action des intervenants. La plateforme propose également un mode "headless". Cela permet d'exécuter une ou plusieurs instances de GAMA sans interface utilisateur, de sorte que les modèles et les expériences peuvent être lancés sur une grille ou un cluster de calcul. Ces simulations concernent généralement des dizaines de milliers d'agents et exigent des ressources de calcul importantes. L'accès aux infrastructures de calculs intensifs est requis pour l'exécution de plans d'expérience conséquents.

3.3 Exploration de scénarios "how-to" et "what-if" via OpenMole

Simuler l'évacuation de plusieurs dizaines voir des centaines de milliers d'agents qui interagissent entre eux, avec l'environnement et avec un aléa lui-même dynamique est un véritable défi en matière de temps de calcul. Ce défi se couple avec la complexité des scénarios explorés et des plans d'expérience programmés. ESCAPE permet de définir et d'explorer des scénarios "How-to" et "What-if" via le logiciel OpenMOLE (Open MOdeL Experiment) sur des environnements de calculs intensifs [19]. Dans ESCAPE, un scénario "what-if" consiste à introduire des événements perturbateurs pour voir comment ils affectent le déroulement d'une simulation (e.g. absence d'alerte ou route bloquée). En revanche, un scénario "how-to" a pour objectif de minimiser certaines variables telles que le temps total d'évacuation ou le nombre d'individus exposés aux aléas. Le cas d'application sur le risque d'inondation présenté dans la Section 3.4 ci-après est un exemple de scénario "what-if" dans lequel le temps de réaction des autorités est évalué. L'objectif, à travers ce scénario, est d'évaluer les conséquences sur les populations d'une réaction tardive des autorités ?

3.4 Une diversité d'indicateurs pour analyser un processus complexe via R-Shiny

Une simulation produit entre 1 et 2 Go de données, en fonction de la durée de la simulation et de l'intervalle d'enregistrement des données. Le choix d'exporter un volume important de données brutes a été fait pour étudier la robustesse ou la sensibilité du modèle aux variations de nombreux indicateurs. Au total, une trentaine d'indicateurs sont exportés afin d'étudier ce processus complexe. Une seconde application R-shiny [4] a été développée pour explorer

ces données sous la forme de résultats statistiques et cartographiques. Tout cela requiert un travail d'agrégation de données préalable pour produire des résumés statistiques pour chaque combinaison de paramètres. Le modèle est stochastique à plusieurs égards. Certains attributs des ménages, selon les scénarios, sont générés de manière aléatoire. Le choix d'évacuer, l'itinéraire d'évacuation ou encore le temps de réponse individuelle sont des potentielles sources de stochasticité qui doivent être prises en compte dans l'analyse. Le couplage d'OpenMOLE et des moyens de calcul intensifs haute-performance du CRIANN, le Centre Régional Informatique et d'Applications Numériques de Normandie, permet la parallélisation à large échelle afin d'explorer la combinatoire des paramètres et de saisir la stochasticité des modèles. Cela représente parfois plusieurs milliers d'exécutions dont certaines peuvent mettre plusieurs heures à s'exécuter. Les résultats statistiques agrégés des simulations correspondent à une moyenne de 30 répliquations au minimum d'une combinaison de paramètres pour capturer toute la stochasticité du système. L'archive contient des informations très précises sur, par exemple, l'exposition des individus aux aléas ou sur l'évolution de l'état de l'évacuation. Dans ESCAPE, il est possible de paramétrer tous les aspects, que ce soit l'intervalle de temps d'enregistrement ou la liste des indicateurs à exporter. Par défaut, le système sauvegarde la position de chaque agent "véhicules" toutes les secondes ainsi que diverses informations telles que le nombre de personnes en sécurité ou le nombre de véhicules bloqués toutes les minutes. Le parallélisme des traitements sous R permet l'exploration massive de plusieurs téraoctets de sorties statistiques (shapefiles et autres fichiers "text/csv"). Des listes d'indicateurs au format ".RData" sont alors créées pour faciliter l'exploration et la visualisation dans l'interface web (figure 3). Elle offre un aperçu général d'un plan d'expérience via des indicateurs mais permet également d'explorer, via des sliders, chaque combinaison de paramètres du plan. L'application offre, par exemple, la possibilité de visualiser l'état du trafic routier au cours de la simulation (nombre de véhicules sur les routes, temps d'embouteillage des tronçons routiers) ou d'identifier les "hotspots" d'exposition des individus aux aléas (nombre d'individus exposés, localisation des "victimes").

Nous présentons dans la section qui suit deux exemples d'applications développées dans le cadre de collaborations avec des opérateurs de gestion de crise, préfecture et communauté de

communes notamment.

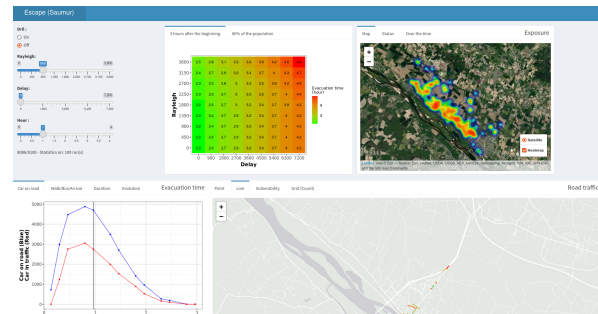


FIGURE 3 – Application web « R-shiny » d'analyse et d'exploration des sorties statistiques des modèles. L'application est composée de quatre grands blocs dans lesquels sont présentés des indicateurs globaux sur l'ensemble du plan d'expérience ainsi que des informations sur chaque combinaison de paramètres du plan. Les divers blocs présentent des informations relatives à l'exposition des individus (e.g. heatmap des "victimes"), l'évolution de l'évacuation (e.g. nombre de personnes en sécurité) et l'état du trafic routier (e.g. temps d'embouteillage).

4 Études de cas - Application du système ESCAPE

Nous illustrons le système ESCAPE par l'étude de deux cas de gestion de crise : l'évacuation massive de la ville de Saumur et ses environs (France) suite à une rupture de digues ; et l'évacuation spontanée et réactive du sud de la région de Basse-Terre durant une crise éruptive de La Soufrière de Guadeloupe (France).

4.1 Inondation et rupture de digues à Saumur

La ville de Saumur est située dans le sud-est du département du Maine-et-Loire, dans la région des Pays de la Loire. La commune compte plus de 26 000 habitants intra-muros et environ 30 000 habitants dans son agglomération. C'est le lieu de la confluence du Thouet avec la Loire. Pour se protéger, la ville s'est dotée d'ouvrages de protection, mais en cas de crue extrême, ces ouvrages pourraient ne pas suffire. La ville de Saumur a donc élaboré un dispositif ORSEC d'évacuation en cas de crue de la Loire. Ce dispositif se décline en trois phases et niveaux d'actions en fonction de l'intensité de la crue de la Loire : (1) La phase de vigilance renforcée inclut des actions préparatoires à une évacuation et des

actions d'information ; (2) La phase de préoccupations sérieuses avec les premières évacuations des personnes vulnérables ; (3) La phase de péril imminent qui prévoit l'évacuation totale de la population.

Le modèle est testé pour un scénario d'évacuation et d'aléa dans lequel l'alerte est au niveau de vigilance renforcée et où la rupture de digue est programmée pour 7 heures du matin, lorsque la plupart des gens sont chez eux provoquant l'évacuation réactive et totale des zones de danger.

La ville de Saumur et la population synthétiques ont été recréés numériquement à partir des données OSM² et de l'INSEE³ ainsi que les points-clés des outils de la gestion de crise du Plan Communal de Sauvergarde de la ville de Saumur. Plus précisément, le système intègre la localisation de sirènes fixes et mobiles pour alerter la population en cas de danger. Les abris et les centres d'accueil sont également localisés et définis dans le système. Pour les populations du centre-ville, un découpage en 4 zones est réalisé avec un abri associé à chacune d'entre elles pour permettre aux personnes non autonomes de se rendre à pied en lieu sûr. Les lieux de collecte décrits dans les PCS permettent aux personnes non-autonomes d'être transportées par des bus depuis des points de collecte vers des centres d'accueil (1 seul de référencé dans le PCS) ou hors de la zone de danger. Le scénario d'aléa correspond au scénario n°6 établi et détaillé dans le cadre de l'étude de danger de la digue réalisé par un bureau d'études pour la DREAL Centre-Val-de-Loire en 2014. ESCAPE offre la possibilité d'intégrer divers comportements 'réflexes' face aux aléas, tels que le refus d'évacuer ou la panique. Ici, le temps de réponse des individus est modélisé par la loi de Rayleigh [18, 1] et l'ensemble des individus exposés évacuent dès qu'ils ont reçu l'alerte (sirènes mobiles et fixes).

Le plan d'expérience élaboré a un double objectif. D'une part, il vise à évaluer les conséquences d'une alerte précoce comme tardive des autorités sur les évacuations, et d'autre part, il intègre une réaction différenciée des populations à l'alerte afin d'analyser les conséquences des comportements individuels sur la réussite de l'évacuation. Ces mouvements de populations vont également engendrer le départ spontané des individus localisés en dehors de la zone de danger. La figure 4 montre un résultat intéressant de cette exploration. On observe que le déclenchement différé

de l'alerte (entre 0 et 180 minutes après la rupture de digue) à des conséquences importantes sur l'évacuation pour ce scénario d'aléa. En effet, alors que 4 heures sont en moyenne nécessaire pour évacuer la population si la réaction des autorités est immédiate, une alerte tardive entraîne l'exposition en moyenne de plus d'un millier d'individus à l'aléa. L'estimation d'un temps moyen d'évacuation permet ainsi de donner une durée de référence pour étalonner les prises de décisions durant la crise.

4.2 Éruption phréatique de La Soufrière de Guadeloupe

L'activité du stratovolcan de La Soufrière de Guadeloupe a lentement augmenté depuis 1992 jusqu'à atteindre un seuil critique en 2017. Depuis, et compte tenu de l'augmentation de l'activité du système hydrothermal, l'observatoire volcanologique et sismique de Guadeloupe exerce une vigilance accrue. Une future intensification de l'activité du volcan n'est pas exclue. Et bien souvent, la seule protection possible contre les aléas volcaniques violents est de s'échapper de ces zones de danger. Il est donc important de planifier la réponse d'urgence à une future crise éruptive et d'identifier les problèmes ou les dangers potentiels d'une évacuation de masse de la région du sud de la Basse-Terre où se concentrent enjeux stratégiques et structurels [15].

Le modèle intègre seulement la phase phréatique de la dernière éruption magmatique majeure de la Soufrière en 1530 [13]. Le but de ce scénario est de mesurer les effets d'une forte sédimentation de cendres sur les mobilités des individus, via un plan d'expérience qui fera varier à la fois les réponses individuelles (accepter d'évacuer ou non, délais de la prise de décision) et celles des autorités. Les véhicules se trouvant donc dans la zone de danger, où l'épaisseur des dépôts de cendres est supérieure à 100 mm, sont considérés comme bloqués et ne peuvent plus être utilisés pour fuir. L'environnement de simulation représente une zone d'étude de 319 km² soit 7 communes de la Communauté d'agglomération Grand Sud Caraïbe. En ce qui concerne la population, 58478 agents synthétiques ont été créés et répartis dans plus 28000 ménages à partir du recensement de population de 2017. Les ménages véhiculés comme les piétons se déplacent sur plus de 3200 kilomètres de linéaire routier pour s'échapper des zones de danger.

La Soufrière, sans signe précurseur, entre en éruption tôt le matin à 8h55 pendant les vacances

2. *OpenStreetMap* <https://www.openstreetmap.org/>

3. *Données carroyées – Carreau de 200m* <https://www.insee.fr/>

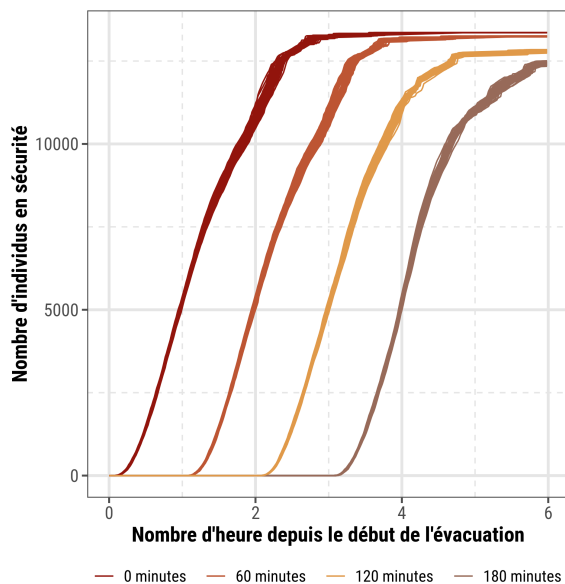


FIGURE 4 – La carte illustre l’extension spatiale de la crue de la Loire 6 heures après l’ouverture de la brèche. Les zones rouges correspondent aux zones à évacuer, 4 zones au total. Pour chaque zone, il y a un ou plusieurs centres d’accueil dont certains sont situés à l’intérieur de la zone de danger, évacuation verticale de la population. La figure présente l’évolution du nombre de personnes en sécurité au cours de l’évacuation et selon quatre modalités du temps d’alerte des autorités. Ces temps de réaction correspondent aux divers temps de l’alerte. Dans ce plan d’expériences, la population est alertée immédiatement ou quelques heures après l’ouverture de la brèche.

scolaires comme lors de l’éruption phréatique de 1976. Cela signifie que l’ensemble des agents sont chez eux et évacuent spontanément. Les premiers mouvements de population sont fonction de la loi de probabilité Rayleigh dont le paramètre σ varie entre 15 et 90 minutes. En considérant cette configuration de base, nous avons pu estimer que le temps d’évacuation moyen de ménages véhiculés est de 3 heures et 20 minutes.

Le système enregistre la position de chaque agent toutes les secondes. Il est donc possible, en post-traitements, de caractériser l’état des véhicules au cours de la simulation. La classification des tronçons routiers en trois états (fluide, modéré, dense) est basée sur la vitesse moyenne de déplacement, le nombre et la durée de présence des agents sur le tronçon, et tient compte des caractéristiques du tronçon comme sa longueur et son nombre de voies. La figure 5 présente l’état du trafic routier 47 minutes après une évacuation partielle et spontanée de certaines communes exposées au risque volcanique. Le départ massif et rapide provoque une forte congestion du réseau routier, notamment sur les communes de Basse-Terre et Saint-Claude, ralentissant l’évacuation.

Ces deux exemples d’applications sont développés dans le cadre de collaborations avec des opérateurs de gestion des risques et des crises, la communauté du Val d’Authion d’une part et la Préfecture de Guadeloupe d’autre part. Ces collaborations ont permis de collecter de nombreuses données et informations pour calibrer les modèles. Les simulations nous conduisent maintenant à discuter des premiers résultats afin d’évaluer la pertinence de ces modèles. Si les temps globaux d’évacuation d’un territoire sont des informations qui intéressent au premier plan les acteurs de la gestion de crise, nous avons pu remarquer que la construction du modèle était également un levier pour motiver les acteurs de la gestion des risques. C’est notamment le cas lorsqu’une commune, parmi trois ou quatre, n’a pas encore élaboré son plan de gestion de l’évacuation.

5 Conclusion

La série d’outils développée offre la possibilité de saisir la dynamique complexe des évacuations massives en proposant une variété de scénarios d’aléas et d’évacuation à grande échelle. Plutôt que de proposer une prédiction précise des événements, l’objectif est de proposer un panorama de résultats potentiels, à travers une chaîne de traitements complète et une vaste exploration

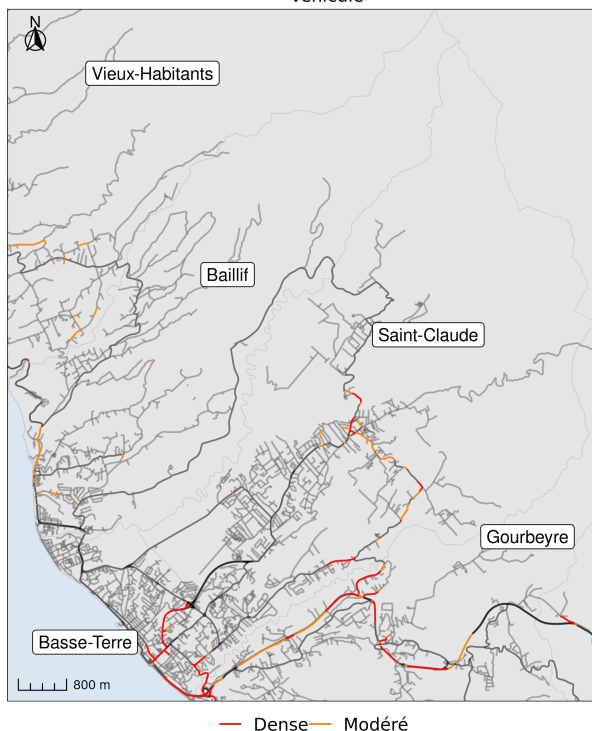
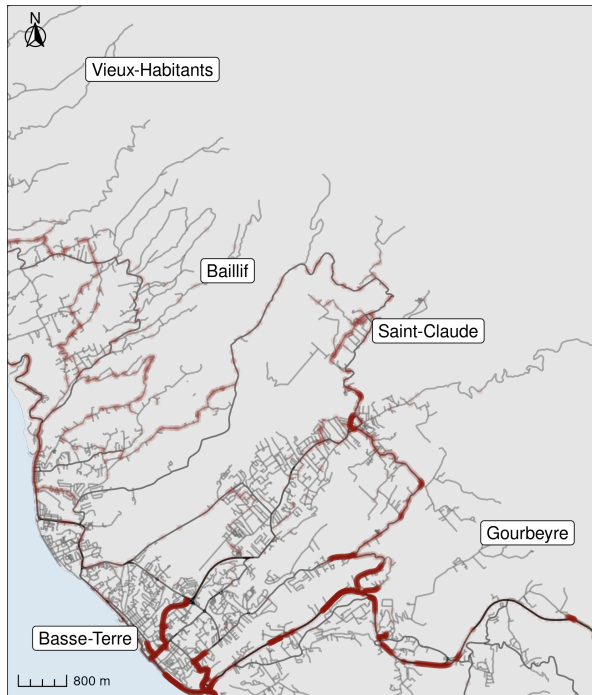


FIGURE 5 – La carte présente l'état du trafic routier lors d'une évacuation partielle et spontanée de certaines communes de Basse-Terre en réponse aux premières manifestations visibles d'une éruption (45 minutes après le début de l'éruption).

des scénarios, afin de fournir aux responsables de la gestion de crise des informations pour cerner la complexité d'un tel processus et de fournir une aide à la réflexion dans le cadre de sa mise en oeuvre. Il convient toutefois de noter certaines limites du système comme la non-intégration de comportement individuel déviant pouvant générer des incidents néfastes à l'évacuation. Des progrès ont déjà été réalisés en ce sens, il est actuellement possible d'intégrer dans ESCAPE des comportements réflexes complexes tels que rejoindre les membres de sa famille. Le processus d'intégration de ces comportements est toujours en cours et les résultats seront publiés prochainement sur un autre cas d'application et sur pour un autre risque, la métropole de Rouen et le risque industriel.

Remerciements. Ce travail est financé par le projet ANR ESCAPE, subvention ANR-16-CE39-0011-01 de l'Agence Nationale de la Recherche française.

Références

- [1] A. Beloglazov, M. Almashor, E. Abebe, J. Richter, and K. C .B. Steer. Simulation of wildfire evacuation with dynamic factors and model composition. *Simulation Modelling Practice and Theory*, 60 :144–159, 2016.
- [2] R. J. Blong. Volcanic Hazards Risk Assessment. In *Monitoring and Mitigation of Volcano Hazards*, pages 675–698. Springer Berlin Heidelberg, 1996.
- [3] CEPRI. L'évacuation Massive Des Populations : Les Territoires Face à l'inondation. Technical report, Centre Européen de Prévention du Risque d'Inondation., 2014.
- [4] W. Chang, J. Cheng, J. J. Allaire, Y. Xie, and J. McPherson. shiny : Web Application Framework for R, 2017.
- [5] K. Chapuis, P. Minh-Duc, A. Brugière, J. D. Zucker, A. Drogoul, P. Tranouez, E. Daude, and P. Taillandier. Exploring multi-modal evacuation strategies for a landlocked population using large-scale agent-based simulations. *International Journal of Geographical Information Science*, 36(9) :1741–1783, 2022.
- [6] G. Czura, P. Taillandier, P. Tranouez, and E. Daude. MOSAIC : City-level agent-based traffic simulation adapted to emergency situations. In *Springer Proceedings in Complexity*, pages 265–274. Springer, 2015.

- [7] N. R. Dalezios. Environmental Hazards Methodologies for Risk Assessment and Management. *Water Intelligence Online*, 16 :9781780407135, 2017.
- [8] E. Daude, K. Chapuis, P. Taillandier, C. Caron, A. Drogoul, B. Gaudou, and S. Rey-Coyrehourcq. ESCAPE : Exploring by Simulation Cities Awareness on Population Evacuation. *ISCRAM, International Conference on Information Systems for Crisis Response and Management*, 2019-May(May) :76–93, 2019.
- [9] B. Fabrice Yaméogo, P. Gastineau, P. Hankach, and P. O. Vandanjon. Comparing Methods for Generating a Two-Layered Synthetic Population. *Transportation Research Record*, 2675(1) :136–147, jan 2020.
- [10] C. E. Haque and D. Blair. Vulnerability to Tropical Cyclones : Evidence from the April 1991 Cyclone in Coastal Bangladesh. *Disasters*, 16(3) :217–229, sep 1992.
- [11] U. Haque, M. Hashizume, K. N. Kolivras, H. J. Overgaard, B. Das, and T. Yamamoto. Reduced death rates from cyclones in Bangladesh : What more needs to be done? *Bulletin of the World Health Organization*, 90(2) :150–156, feb 2012.
- [12] J. Kim, S. Lee, and S. Lee. An evacuation route choice model based on multi-agent simulation in order to prepare Tsunami disasters. *Transportmetrica B*, 5(4) :390–406, 2017.
- [13] J. C. Komorowski, Y. Legendre, B. Caron, and G. Boudon. Reconstruction and analysis of sub-plinian tephra dispersal during the 1530 A.D. Soufrière (Guadeloupe) eruption : Implications for scenario definition and hazards assessment. *Journal of Volcanology and Geothermal Research*, 178(3) :491–515, 2008.
- [14] G. Lämmel, H. Klüpfel, and K. Nagel. The MATSim Network Flow Model for Traffic Simulation Adapted to Large-Scale Emergency Egress and an Application to the Evacuation of the Indonesian City of Padang in Case of a Tsunami Warning. In *Pedestrian Behavior*, pages 245–265. Emerald Group Publishing Limited, nov 2009.
- [15] F. Leone, J. C. Komorowski, M. Gherardi-Leone, and G. Lalubie. Integrating spatial accessibility in the design of volcano evacuation plans in the French West Indies (Guadeloupe and Martinique). *Journal of Applied Volcanology*, 8(1) :8, dec 2019.
- [16] F. Leone, N. Meschinet de Richemond, and F. Vinet. *Aleas naturels et gestion des risques*. Presses universitaires de France, Paris, France, 2010.
- [17] R. Lovelace, M. Dumont, R. Ellison, and M. Založnik. *Spatial Microsimulation with R*. Chapman and Hall/CRC, sep 2017.
- [18] E. Mas, F. Imamura, and S. Koshimura. Modeling the Decision of Evacuation from Tsunami, based on Human Risk Perception. *Annual Meeting of the Tohoku Branch Technology Research Conference, Japan Society of Civil Engineers*, pages 1–2, 2011.
- [19] R. Reuillon, M. Leclaire, and S. Rey-Coyrehourcq. OpenMOLE, a workflow engine specifically tailored for the distributed exploration of simulation models. *Future Generation Computer Systems*, 29(8) :1981–1990, oct 2013.
- [20] F. X. Robin and V. Demeules. Méthodes d’estimations de population : comparaisons et seuils de validité. 2005.
- [21] Arnaud Saval, Duc Pham Minh, Kevin Chapuis, Pierrick Tranouez, Clément Caron, Éric Daudé, and Patrick Taillandier. Dealing with mixed and non-normative traffic. An agent-based simulation with the GAMA platform. *PLOS ONE*, 18(3) :e0281658, mar 2023.
- [22] F. F. Stephan. Iterative method of adjusting frequency tables when expected margins are known. *Annals of Mathematical Statistics*, 13 (2) :166–178, 1942.
- [23] P. Taillandier, B. Gaudou, A. Grignard, Q-N. Huynh, N. Marilleau, P. Caillou, D. Philippon, and A. Drogoul. Building, composing and experimenting complex spatial models with the GAMA platform. *GeoInformatica*, 23(2) :299–322, 2019.
- [24] J. Yu, C. Zhang, J. Wen, W. Li, R. Liu, and H. Xu. Integrating multi-agent evacuation simulation and multi-criteria evaluation for spatial allocation of urban emergency shelters. *International Journal of Geographical Information Science*, 32(9) :1884–1910, 2018.
- [25] Y. Zhu, K. Xie, K. Ozbay, and H. Yang. Hurricane Evacuation Modeling Using Behavior Models and Scenario-Driven Agent-based Simulations. In *Procedia Computer Science*, volume 130, pages 836–843, 2018.

Escape-SG : un jeu sérieux pour mieux préparer les évacuations de masse

Arnaud Saval^a Mathieu Bourgeois^b
arnaud.saval@univ-rouen.fr mathieu.bourgeois@insa-rouen.fr

Éric Daudé^b Pierrick Tranouez^a
eric.daude@cncrs.fr pierrick.tranouez@univ-rouen.fr

^aUniversité de Rouen Normandie, EA LITIS

^bCNRS, Normandie Université, UMR 6266 IDEES

Résumé

Le jeu sérieux Escape-SG a pour objectif de proposer des outils de simulation lors de la formation à la gestion de crise pour des situations d'évacuation massives de zones urbaines. Pour cela, il est nécessaire de proposer un ensemble de simulations crédibles des situations étudiées et de les intégrer dans une interface permettant de reproduire les interactions réelles d'une cellule de gestion de crise. Le jeu sérieux Escape-SG propose ainsi de coupler une simulation de trafic urbain s'appuyant sur les travaux du projet ESCAPE avec une interface offrant une visualisation en direct des événements simulés tout en proposant des actions d'intervention mimant les actions possibles dans un cas réel de gestion de crise.

Mots-clés : Jeu sérieux, simulation à base d'agents, évacuation

Abstract

The objective of the Escape-SG serious game is to propose simulation tools for crisis management training in the context of mass evacuation situations in urban areas. To do so, it is necessary to propose a set of credible simulations of the studied situations and to integrate them in an interface allowing to reproduce the real interactions of a crisis management cell. The serious game Escape-SG thus proposes to couple an urban traffic simulation based on the ESCAPE project with an interface offering a live visualisation of the simulated events while proposing intervention actions mimicking those possible in a real crisis management case.

Keywords: Serious Game, Agent Based Simulation, Evacuation

1 Introduction

Les territoires soumis à d'importants risques, qu'ils soient naturels ou technologiques, ont la nécessité de se préparer pour pouvoir y faire face efficacement en temps voulu. Cette préparation prend la forme de séances d'entraînements et de formations grandeur réelle pour les différentes personnes composant les cellules de crises où des scénarios pré-définis sont rejoués pour étudier les comportements et prises de décisions de chaque participant. Ces séances d'entraînement prennent la forme de jeux sérieux [6] où chaque participant incarne un rôle et où l'entièreté d'un scénario est déroulé, avec la possibilité de rejouer certaines scènes précises.

Ces situations de crise impliquant des zones urbaines de grande taille s'accompagnent, lors de cas réels, d'une gestion de la population subissant l'aléa. La réponse comportementale de cette population n'est pas nécessairement évidente à prédire et peut néanmoins influencer de façon importante sur les décisions prises par la cellule de crise. L'utilisation de simulations à base d'agents permet d'intégrer ces dynamiques sociales sans pour autant réaliser un exercice grandeur réelle impliquant des milliers de personnes [8].

Le projet Escape-SG [5] a pour objectif de proposer des outils de jeux sérieux avec interface de visualisation et d'interaction en temps réel pour la réalisation d'entraînements à des situations d'évacuation de population lors d'aléas naturels ou technologiques. Escape-SG s'appuie sur le programme de recherche ESCAPE [4] qui utilise différents systèmes d'information géographique avec des modélisations multi-échelle à base d'agents et des méthodes d'exploration de simulations. Le but est alors d'inclure ces systèmes de simulations d'évacuation dans un jeu sérieux qui puisse être au centre d'une session d'entraînement d'une cellule de crise.

La Section 2 propose une revue de la littérature sur les systèmes d'étude de gestion de crise, des systèmes de simulation jusqu'aux jeux sérieux. La Section 3 présente plus en détails les hypothèses sélectionnées pour le modèle de simulation de trafic routier et la Section 4 discute des actions de transformation d'une simulation de trafic routier en situation de crise vers un jeu sérieux. Enfin, la Section 5 propose une conclusion et évoque les pistes d'amélioration envisagées pour la poursuite du système.

2 État de l'art

L'aide à la formation de cellules de gestion de crise se fait par le biais de différents outils. La simulation numérique, notamment à base d'agents lorsqu'il convient de simuler des acteurs humains, permet de reproduire une situation d'entraînement à moindre coût [8]. D'autre part, la réalisation de ces formations sous l'angle du jeu sérieux apporte de la dynamique aux exercices pour l'ensemble des participants [6].

2.1 Gestion de crise par simulation

Les simulations à base d'agents sont utilisées depuis plus de dix ans pour aider à étudier les situations de crise à large échelle géographique [8]. Cela permet de reproduire dans un environnement contrôlé des phénomènes qui impactent une population importante, que ce soit la propagation d'une épidémie, l'évacuation d'un bâtiment en feu ou encore la mise à l'abri d'une ville soumise à un risque technologique [4]. Il existe notamment un nombre important de systèmes qui permettent la simulation plus particulière de la gestion de crise à l'échelle d'une population face à des cas d'inondation [2].

En parallèle, des systèmes de simulations ont été mis au point pour reproduire le comportement d'automobilistes se déplaçant dans un milieu urbain [14]. Une fois le comportement en situation habituelle correctement reproduit il est possible de le simuler face à une crise ce qui est au centre du projet de recherche ESCAPE [3]. ESCAPE inclut aussi différentes modalités de transport (piéton, voiture, bus) et la possibilité pour les acteurs simulés de changer de modalité pendant la simulation, modélisant de façon plus réaliste l'évacuation d'une zone urbaine en situation de crise.

2.2 Jeux sérieux et gestion de crise

Il existe un grand nombre de jeux sérieux traitant de la problématique de la gestion de crise au sens large [12]. Ces jeux peuvent prendre l'apparence de jeux de cartes, de jeux de plateau, de jeux de rôle ou encore de jeux vidéo. Leur rôle est d'assister la cellule de crise dans sa formation en proposant de dérouler un scénario adaptatif aux actions entreprises par les stagiaires/apprenants.

Plus spécifiquement, il existe déjà des jeux sérieux numériques traitant de la gestion de l'évacuation de population sur de larges zones que ce soit pour des incendies de brousse [1] ou pour des inondations [10] [13]. Ces jeux sont créés autour de cas géographique précis et proposent d'observer et d'interagir avec la situation en ne prenant qu'un seul point de vue, celui de la personne morale disposant du plus de responsabilité.

Avec Escape-SG, nous souhaitons mettre au point une plateforme permettant de connecter un jeu sérieux avec une simulation de comportement de population sans être spécifiquement attaché à un cas précis d'utilisation, que ce soit pour la zone géographique étudiée ou le risque au centre de la situation.

3 Modélisation du trafic routier et crise

Escape-SG se découpe en deux parties majeures : une simulation à base d'agents déployée sur un serveur et une interface qui sert de client. Cette section évoque la simulation du trafic routier utilisée et son adaptation à Escape-SG.

3.1 Modèle ESCAPE initial

Le modèle ESCAPE [3] qui sert de base à la simulation utilisée dans Escape-SG repose sur les éléments suivants :

- **Environnement** : la zone géographique est modélisée d'après les données OpenStreet-Map [11] disponibles, que ce soit pour les routes ou pour l'emplacement des bâtiments.
- **Piétons** : les agents piétons partagent initialement un bâtiment avec leur ménage. Ensuite, ils se déplacent le long des routes et peuvent rentrer dans des bâtiments (résidentiels ou non).
- **Véhicules** : différentes modalités de véhicules sont proposées (voiture et bus), chaque véhicule suivant le plus court chemin vers sa destination.

Chaque agent dispose de son propre agenda quotidien dans une situation normale, issu des données de l'enquête ménages déplacements, et adapte son comportement lors d'une situation exceptionnelle de crise. Cela peut amener les agents à changer de modalité de transport pour se mettre à l'abri, chercher leur famille ou fuir, selon un comportement personnalisé [7].

3.2 Adaptations pour Escape SG

Afin d'assurer un fonctionnement en temps raisonnable, le modèle ESCAPE a été adapté pour fonctionner en configuration Escape-SG. Le moteur de simulation a été modifié pour intégrer SUMO [9] qui est spécialisé dans les simulations de trafic.

Le comportement des agents est partiellement simplifié. En condition normale, chaque agent conserve son agenda, se déplaçant d'un point à un autre et patientant dans des bâtiments entre ces deux déplacements. Lorsque la situation de crise est déclenchée, une nouvelle destination est imposée à chaque agent, les amenant à fuir ou se mettre à l'abri. Par ailleurs, les agents peuvent toujours changer de modalité, notamment commencer comme piéton et décider d'attendre à un arrêt de bus pour être évacué en transport en commun.

La zone simulée est aussi réduite au centre-ville avec quelques centaines d'agents simulés. La limitation de la zone simulée ainsi que l'optimisation des comportements des agents, au détriment d'une part du réalisme initialement présent dans le modèle ESCAPE, rendent possible une simulation en temps réel nécessaire pour l'interaction avec les utilisateurs.

4 De la simulation au jeu sérieux

La composante d'interface du projet Escape-SG agit comme un client vis à vis de la simulation qui fonctionne sur un serveur. C'est cette interface qui crée le jeu sérieux avec lequel les utilisateurs vont pouvoir interagir.

4.1 Interface graphique

Dans le cas d'une évacuation de masse, il est nécessaire pour les utilisateurs d'avoir une visualisation graphique de la situation simulée, afin de mieux comprendre la situation globale à chaque instant. La figure 1 montre le rendu graphique de la région étudiée avec des bâtiments reconstruits en trois dimensions.



FIGURE 1 – Interface de visualisation et d'interaction du projet Escape SG

En plus de pouvoir visualiser la simulation, les utilisateurs peuvent interagir en direct pour par exemple fermer des routes, lancer des norias de bus ou transformer des bâtiments en refuges. Pour cela, les joueurs disposent d'une série de boutons qui vont alors modifier l'environnement de la simulation, amenant à une mise à jour du comportement des agents.

4.2 Scénarios de jeu

L'utilisateur de l'interface du projet Escape-SG peut se connecter à la simulation sous-jacente en prenant différents rôles. Chacun de ces rôles représente un niveau de responsabilités et un accès différencié aux outils, suivant que l'on incarne la préfecture, le gestionnaire du réseau électrique ou le chef des services de secours pour prendre quelques exemples. Tous ces utilisateurs disposent alors d'un chat leur permettant de communiquer ensemble pendant l'exercice.

Un utilisateur peut aussi se connecter avec le rôle spécial de manager. Cette personne peut alors dérouler des scénarios pré-écrits concernant l'aléa, que ce soit son déclenchement, son évolution ou sa fin. Tous les joueurs sont donc connectés à la même simulation, chacun disposant en direct de sa propre interface d'interaction.

5 Conclusion

Cet article présente le jeu sérieux Escape-SG qui propose d'utiliser une simulation à base d'agents dans un contexte d'entraînement à l'évacuation de population lors d'une situation de gestion de crise. Pour cela, nous avons développé une plateforme connectant une simulation de trafic routier en milieu urbain, proposant notamment des comportements en situation de crise pour les acteurs simulés, avec une interface de visualisation en trois dimensions et d'interaction en

direct. Ainsi, Escape-SG peut être inclus dans des sessions de formation pour des cellules de gestion de crise en s'adaptant aux différents rôles des apprenants tout en proposant des scénarios qui peuvent s'adapter dynamiquement aux choix réalisés en direct.

Par la suite, nous projetons d'implémenter différentes options de scénarios pour à la fois rejouer des cas de crises déjà connus mais aussi permettre la création simplifiée de cas inédits. La suite du développement impliquera aussi d'améliorer l'interface de façon globale en offrant plus d'actions possibles aux différents rôles proposés lors des sessions de formation tout en assurant une meilleure optimisation de l'affichage. Enfin, une série de tests du jeu sérieux avec différents publics sera programmée pour avoir des retours d'utilisateurs et améliorer l'ensemble de l'expérience autour du jeu sérieux Escape-SG.

Remerciements

Projet soutenu par la Région Normandie dans le cadre des dispositifs « Réseaux d'Intérêt Normands » RIN-Tremplin 2019.

Références

- [1] Carole Adam, Charles Bailly, and Julie Dugdale. Communication during bushfires, towards a serious game for a serious matter : communication during bushfires. In *Research Anthology on Managing Crisis and Risk Communications*, pages 100–130. IGI Global, 2023.
- [2] Anshuka Anshuka, Floris F van Ogtrop, David Sanderson, and Simone Z Leao. A systematic review of agent-based model for flood risk management and assessment using the odd protocol. *Natural Hazards*, 112(3) :2739–2771, 2022.
- [3] Kevin Chapuis, Patrick Taillandier, Benoit Gaudou, Alexis Drogoul, and Eric Daudé. A multi-modal urban traffic agent-based framework to study individual response to catastrophic events. In *PRIMA 2018*, pages 440–448. Springer, 2018.
- [4] Eric Daudé, Kevin Chapuis, Patrick Taillandier, Pierrick Tranouez, Clément Caron, Alexis Drogoul, Benoit Gaudou, Sebastien Rey-Coyrehourcq, Arnaud Saval, and Jean-Daniel Zucker. Escape : exploring by simulation cities awareness on population evacuation. In *ISCRAM 2019*, 2019.
- [5] Éric Daudé and Pierrick Tranouez. Escape-sg : un simulateur d'évacuation massive de population pour la formation des acteurs à la gestion de crise. *Netcom. Réseaux, communication et territoires*, (34-3/4), 2020.
- [6] Ines Di Loreto, Simone Mora, and Monica Divitini. Collaborative serious games for crisis management : an overview. In *21st International Workshop on Enabling Technologies : Infrastructure for Collaborative Enterprises*, pages 352–357, 2012.
- [7] Justine Fenet and Éric Daudé. La population, grande oubliée des politiques de prévention et de gestion territoriales des risques industriels : le cas de l'agglomération rouennaise. *Cybergeog : European Journal of Geography*, 2020.
- [8] Glenn I Hawe, Graham Coates, Duncan T Wilson, and Roger S Crouch. Agent-based simulation for large-scale emergency response : A survey of usage and implementation. *ACM Computing Surveys (CSUR)*, 45(1) :1–51, 2012.
- [9] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MOBility. *International Journal On Advances in Systems and Measurements*, 5(3&4) :128–138, 2012.
- [10] Annabelle Moatty and Franck Sfiligoi Taillandier. Un jeu sérieux fondé sur un modèle agent pour sensibiliser aux solutions fondées sur la nature pour gérer les inondations. *Academic Journal of Civil Engineering*, 40(1) :1–4, 2022.
- [11] OpenStreetMap contributors. Planet dump retrieved from <https://planet.osm.org>. <https://www.openstreetmap.org>, 2023.
- [12] Aleksandra Solinska-Nowak, Piotr Magnuszewski, Margot Curl, Adam French, Adriana Keating, Junko Mochizuki, Wei Liu, Reinhard Mechler, Michalina Kulakowska, and Lukasz Jarzabek. An overview of serious games for disaster risk management—prospects and limitations for informing actions to arrest increasing risk. *International journal of disaster risk reduction*, 31 :1013–1029, 2018.
- [13] Franck Taillandier and Carole Adam. Games ready to use : A serious game for teaching natural risk management. *Simulation & Gaming*, 49(4) :441–470, 2018.
- [14] Pierrick Tranouez, Eric Daudé, and Patrice Langlois. A multiagent urban traffic simulation. *Journal of Nonlinear Systems and Applications*, pages 98–106, 2010.

Construction incrémentale d'un comportement collectif autour d'une proto-émotion de peur

Clément Raïevsky^a François Suro^a
clement.raievsky@univ-grenoble-alpes.fr francois.suro@univ-grenoble-alpes.fr

^aUniv. Grenoble Alpes, Grenoble INP, LCIS, 26000 Valence, France

Résumé

Partant de l'hypothèse que les émotions et la cognition sont indissociables, nous souhaitons doter des agents situés de prémisses des fonctions que les émotions jouent dans notre cognition. Pour ce faire, nous modélisons à l'aide de l'architecture MIND, un comportement réactif dans un contexte proie-prédateur combinant fuite, agrégation, alerte et inactivité. En s'appuyant sur les mécanismes de construction progressive de comportements offerts par MIND, nous étendons ce comportement par l'ajout d'un processus de régulation de l'intensité d'une proto-émotion de peur. Ce comportement individuel est ensuite étendu à un niveau collectif par l'ajout d'une nouvelle compétence s'apparentant à la contagion émotionnelle, à travers l'expression et la perception de l'état émotionnel des autres proies.

1 Introduction

Selon Piaget [11], la construction des structures cognitives qui sous-tendent notre intelligence est incrémentale. Cette théorie du développement cognitif a inspiré de nombreux travaux sur l'apprentissage développemental, cumulatif (ou *life-long learning*[10]) et l'apprentissage par curriculum[1], pour les agents situés et pour les robots en particulier.

C'est dans ce cadre que l'architecture Modular Influence Network Development (MIND) [15] a été proposée. Dans la lignée de ces travaux, nous proposons ici d'explorer une autre hypothèse de Piaget, selon laquelle l'affect ou les émotions sont indissociables de la cognition. Cette hypothèse a été reprise par de nombreux chercheurs en IA [8], [14] et il est désormais établi que l'affectif et le cognitif sont intriqués, aussi bien durant le développement que dans leur fonctionnement usuel.

Le travail présenté ici vise à étendre MIND par une démarche constructiviste en intégrant les prémisses de deux fonctions primaires des émotions dans la cognition humaine : la fonc-

tion de régulation de l'aspect énergétique du comportement [11] et la fonction de signal social des émotions [6], [9]. La première prend la forme d'une variable représentant l'intensité d'une proto-émotion de peur et son mécanisme de régulation associé, la seconde se traduit par l'échange de signaux dédiés. Nous parlons de proto-émotion car nous ne prétendons pas reproduire le processus cognitif correspondant à une émotion au complet.

Le contexte expérimental proie-prédateur a été choisi car il représente un bon compromis entre une relative simplicité des comportements et un environnement et des missions suffisamment riches pour valoriser des comportements collectifs par rapport à des comportements purement individuels et réactifs. À l'inverse de travaux existants sur le problème proie-prédateur, nous nous intéressons aux comportements des individus et non aux dynamiques de population. Nous sommes donc restés dans le contexte initial de MIND : la robotique.

2 État de l'art

Notre proposition porte sur l'utilisation des émotions artificielles dans la prise de décision des agents situés [7]. Nous ne visons pas à reproduire fidèlement les processus cognitifs humains, mais plutôt à reproduire les fonctions que ces processus remplissent dans la prise de décision d'êtres vivants simples.

Guzzi et al. [5] proposent une architecture de contrôle multi-robots intégrant des états émotionnels partagés qui influencent le comportement individuel. Ces travaux ne sont cependant pas positionnés par rapport au domaine des SMA et les états émotionnels conditionnent directement les comportements actifs du robot. Les biais introduits dans la génération des émotions et dans leurs influences rendent difficile la généralisation de ces résultats.

Le modèle Satisfaction-Altruisme [13], proposé spécifiquement pour la coordination dans les SMA, intègre des signaux d'intentions simples

provenant d'autres agents à la sélection d'actions basée sur des objectifs individuels. L'aspect réactif de cette architecture correspond plus à l'approche visée par MIND qu'aux approches BDI utilisés dans un contexte similaire [2]. Ce modèle réactif ne vise cependant pas à représenter un état émotionnel dans les agents.

3 Architecture de contrôle

MIND [15] est une architecture de contrôle conçue pour répondre aux besoins spécifiques de la robotique développementale et de l'apprentissage ouvert et continu.

Dans MIND les compétences sont encapsulées dans des modules ou *Skills*, représentés par des cercles noirs dans la Figure 1. Ces modules potentiellement hétérogènes et qui peuvent être appris ou programmés sont combinés hiérarchiquement pour former le mécanisme de prise de décision des agents.

L'une des caractéristiques distinctives de MIND est son mécanisme d'influence¹, une approche basée signal permettant le neurocontrôle et la composition vectorielle multi-niveaux de comportements. Par cette approche, l'intégration des informations capteurs et actionneurs s'effectue sous forme de signaux continus. Ce mécanisme de représentation est aussi employé dans la gestion interne d'information, pour la persistance et la formation de concepts, d'une manière proche des travaux sur les *cognitons* [3] ou des *espaces conceptuels* [4].

4 Expériences

Environnement de simulation Afin de placer les agents dans un environnement "physique", que ce soit pour leur perceptions que pour leur contrôle, les expériences ont été réalisées dans un environnement circulaire 2D doté d'un moteur physique.

Prédateur Le comportement du prédateur consiste à se diriger vers la proie la plus proche qui sera tuée s'il parvient à entrer en contact avec elle. Elle sera alors recréée instantanément à une position aléatoire de l'environnement. Sa portée de perception est illimitée.

Proies La Figure 1 présente l'instance de MIND qui contrôle les proies. Elles sont dotées de capteurs permettant : la détection passive du prédateur, simulant l'audition à courte distance ; la détection d'obstacles ; et la collecte des informations nécessaires au comportement d'agrégation.

Elles peuvent également activer un capteur (simulant la vision) qui permet de détecter la position précise du prédateur, mais qui est considéré comme interrompant l'activité de pâture des proies. Ce capteur est occulté par les obstacles et les autres proies.

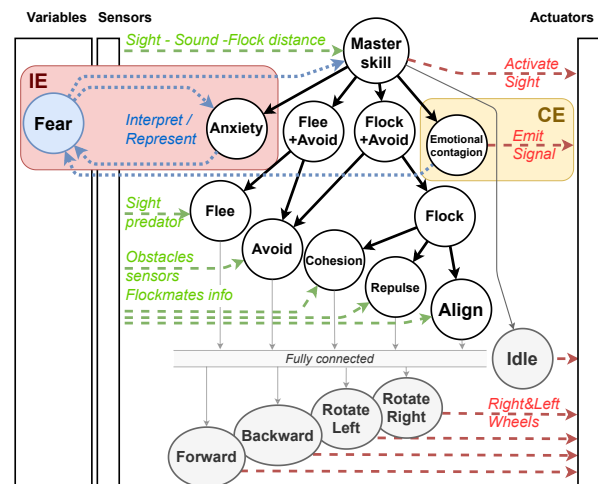


FIGURE 1 – Instance de MIND pour le comportement des proies. Les modules (*skills*) relatifs aux comportements Individuel Émotionnel (IE) et Collectif Émotionnel (CE) sont mis en valeur.

4.1 Comportements des proies

Notre étude porte sur trois étapes de développement successives : dans la première, les proies ont un comportement réactif, sans communication et uniquement basé sur la perception des positions du prédateur et des autres proies. Dans la seconde configuration, les proies sont dotées d'une variable associée à l'intensité d'une proto-émotion de peur qui influence leur comportement. Dans la troisième, l'intensité de cette variable est perçue par les proies voisines. La Figure 1 présente la hiérarchie de compétences correspondant aux trois configurations.

Individuel Réactif Le comportement de base des proies résulte de la combinaison de quatre compétences :

- Une compétence d'**agrégation** (*flocking*) classique [12] qui comporte trois composantes : cohésion, répulsion et alignement.
- Une compétence d'**évitement** d'obstacles
- Une compétence de **fuite** du prédateur qui utilise le capteur visuel lorsqu'il est actif pour s'en éloigner.
- Une compétence d'**inactivité** qui permet de représenter le temps passé à se nourrir.

Le comportement d'agrégation a été mis en place afin de valoriser le comportement collectif

1. Fig.1 : flèches pleines connectant les skills

en introduisant une confusion chez le prédateur qui poursuit la proie la plus proche.

La compétence maîtresse (*Master skill*) est en charge de l'activation de ces différentes compétences via deux compétences intermédiaires ainsi que de l'activation du capteur visuel. Le fonctionnement prédéterminé de cette compétence est le suivant : si le prédateur n'est pas perçu, l'inactivité et l'agrégation sont activées en fonction de la distance à la proie la plus proche et le capteur visuel est activé aléatoirement. Une fois le prédateur perçu, la fuite et l'agrégation sont combinées, la fuite étant d'autant plus prioritaire que le prédateur est proche. Dans ce cas, le capteur visuel reste activé.

Individuel Émotionnel Ce comportement inclue deux nouveaux éléments, encadrés en rouge dans la Fig. 1 : la variable *Fear* représentant l'intensité d'une proto-émotion de peur et une compétence chargée de la régulation de cette intensité nommée, par abus de langage, "*Anxiety*". Cette compétence augmente l'intensité de la peur lorsque le prédateur est perçu ou lorsqu'aucune proie n'est à proximité. Elle diminue la peur lorsque le prédateur n'est plus perçu ou que d'autres proies sont à proximité.

La compétence maîtresse prend en compte la variable *Fear* en augmentant à la fois la probabilité d'activation du capteur visuel et l'influence de la compétence d'agrégation, proportionnellement à son intensité.

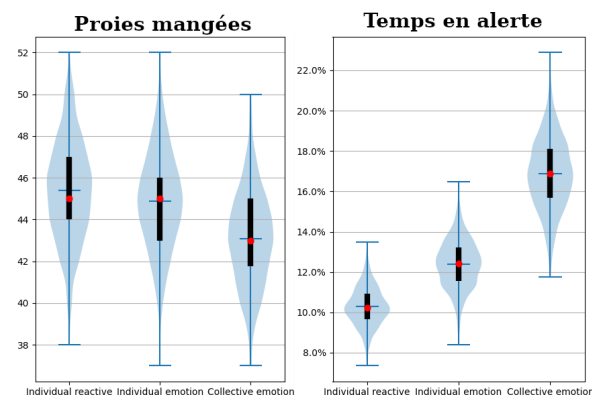


FIGURE 2 – Comparaison des 3 configurations du comportement des proies en termes de nombre de proies mangées et du temps passé en état d'alerte.

Collectif Émotionnel Enfin, la dernière configuration met en œuvre la propagation de l'émotion de peur via l'ajout d'une nouvelle compétence de contagion émotionnelle chargée de percevoir et d'exprimer l'état émotionnel des proies. Cette compétence active l'émission d'un signal sur un front montant de l'intensité de la peur au-

dessus d'un seuil haut. Ce signal reste actif tant que l'intensité de la peur ne diminue pas sous un seuil bas.

L'intensité de la peur est affectée par la contagion émotionnelle de deux manières :

- Fortement par l'apparition de nouveaux signaux.
- Faiblement en présence d'agents n'émettant pas le signal.

La combinaison de l'effet des compétences *Anxiety* et *Emotional contagion* sur la variable *Fear* est comparable au modèle Satisfaction-Altruisme [13] dans son intégration des aspects individuels et collectifs.

5 Résultats

L'introduction de la variable émotionnelle se traduit par une persistance dans le comportement de fuite individuelle. Cependant, la Figure 2 montre que cette persistance, observable dans le temps passé en alerte, n'apporte qu'une faible amélioration du taux de survie, non significative dans les résultats obtenus. L'ajout du mécanisme de contagion émotionnelle provoque des comportements de fuite collectifs, comme illustrés par la Figure 3, qui permettent d'améliorer le taux de survie du groupe, au prix d'un temps plus grand passé en alerte.

L'exploration de l'espace des paramètres ainsi que des variations des comportements décrits mènent à des résultats similaires. Cela suggère que notre implémentation des différents mécanismes émotionnels n'entraîne pas une répartition significativement plus pertinente du temps d'alerte et l'augmentation de la survie semble due à l'augmentation globale du temps d'alerte.

Cependant, l'observation qualitative des comportements montre bien une différence, comme le montre la Figure 3. Une vidéo des différents comportements est disponible à l'adresse suivante : <https://hal.science/hal-04060987>

6 Conclusion et Perspective

Ce travail préliminaire sur l'intégration de mécanismes émotionnels au sein de l'architecture de prise de décision MIND montre qu'elle permet d'intégrer la représentation d'une émotion simple, de clairement définir un processus de régulation de cette représentation et de construire de manière incrémentale un comportement collectif autour de celle-ci. Cela illustre les possibilités d'applications de MIND pour la modélisation de comportements dans les systèmes multi-agents.

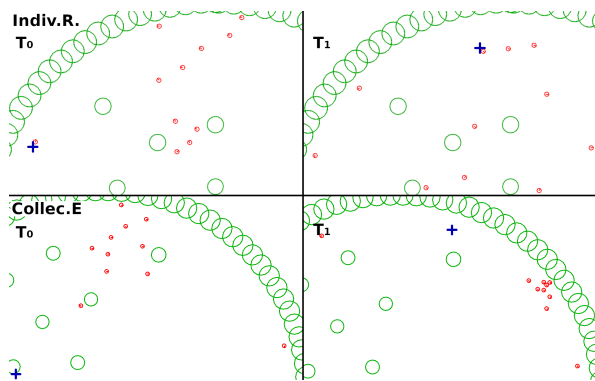


FIGURE 3 – Comparaison des comportements de référence et Collectif Émotionnel (CE) des proies (petits cercles rouges) face à une attaque du prédateur (croix bleue). Au temps T1 le comportement de fuite coordonné apparaît pour le comportement CE.

Malgré une modification subjective des comportements par l'ajout de ces mécanismes et un aspect plus naturel des comportements obtenus, cette première investigation n'a pas permis de mettre en évidence une amélioration quantitative du comportement des proies en terme de survie. Cette constatation peut avoir, selon nous, deux origines : soit les biais introduit par la conception ad hoc de compétences, combinés à un espace de paramètres trop grands, compensent négativement les gains potentiels de l'ajout des mécanismes émotionnels ; soit le fait de ne pas avoir reproduit le mécanisme de régulation de la préparation à l'action associée à ces mécanismes émotionnels a empêché les agents de profiter des fonctions bénéfiques qui leur sont associées.

Les deux suites immédiates au travail présenté ici sont donc l'apprentissage des différentes compétences constituant le comportement des agents d'une part et l'introduction d'un mécanisme de régulation de la préparation à l'action d'autre part.

Références

- [1] Y. BENGIO, J. LOURADOUR, R. COLLOBERT et J. WESTON, "Curriculum learning," in *Int. Conf. on Machine Learning*, ACM, 2009, p. 41-48.
- [2] M. BOURGAIS, P. TAILLANDIER et L. VERCOUTER, "Ben : An agent architecture for explainable and expressive behavior in social simulation," in *EXTRAAMAS, Montreal, QC, Canada, May, 2019*, Springer, 2019, p. 147-163.
- [3] J. FERBER, *Les Systèmes Multi-agents, Vers une intelligence collective*. InterEditions, Paris, 1995.
- [4] P. GÄRDENFORS, "Induction, conceptual spaces and AI," *Philosophy of Science*, t. 57, n° 1, p. 78-95, 1990.
- [5] J. GUZZI, A. GIUSTI, L. M. GAMBARDILLA et G. A. DI CARO, "A Model of Artificial Emotions for Behavior-Modulation and Implicit Coordination in Multi-Robot Systems," in *GECCO*, New York, NY, USA, 2018, p. 21-28.
- [6] S. HARELI et U. HESS, "The social signal value of emotions," *Cognition and Emotion*, t. 26, n° 3, p. 385-389, 2012.
- [7] Z. KOWALCZUK et M. CZUBENKO, "Computational Approaches to Modeling Artificial Emotion – An Overview of the Proposed Solutions," *Frontiers in Robotics and AI*, t. 3, 2016.
- [8] M. MINSKY, *The Society of Mind*. Simon et Schuster éditeurs, 1985.
- [9] K. OATLEY et P. JOHNSON-LAIRD, "Towards a Cognitive Theory of Emotions," *Cognition and Emotion*, t. 1, n° 1, p. 29-50, 1987.
- [10] E. OZTOP et E. UGUR, "Lifelong Robot Learning," in *Encyclopedia of Robotics*, Springer Berlin Heidelberg, 2020, p. 1-12.
- [11] J. PIAGET, "Les relations entre l'intelligence et l'alloctivité dans le développement de l'enfant (I, II, III)," *Bulletin de psychologie*, t. 7, n° 3, p. 143-150, 1954.
- [12] C. W. REYNOLDS, "Flocks, herds and schools : A distributed behavioral model," in *ACM SIGGRAPH computer graphics*, ACM, t. 21, 1987, p. 25-34.
- [13] O. SIMONIN et J. FERBER, "Modeling self satisfaction and altruism to handle action selection and reactive cooperation," in *SAB*, 2000, p. 314-323.
- [14] A. SLOMAN et M. CROUCHER, "Why robots will have emotions," in *Proceedings of the 7th IJCAI*, t. 1, 1981, p. 197-202.
- [15] F. SURO, J. FERBER, T. STRATULAT et F. MICHEL, "A hierarchical representation of behaviour supporting open ended development and progressive learning for artificial agents," *Autonomous Robots*, t. 45, n° 2, p. 245-264, 2021.

Optimisation et sécurité multi-agents

L'optimisation du covoiturage dynamique multi-saut

Corwin Fèvre^a
corwin.fevre@univ-lille.fr

Hayfa Zgaya-Biau^a
hayfa.zgaya-biau@univ-lille.fr

Philippe Mathieu^a
philippe.mathieu@univ-lille.fr

Slim Hammadi^a
slim.hammadi@centralelille.fr

^aUniv. Lille, CNRS, Centrale Lille, UMR 9189, CRISTAL, F-59000 Lille, France

Résumé

Dans cet article, nous proposons une approche multi-agent pour résoudre le problème du covoiturage dynamique multi-saut. Dans notre système, les passagers et les conducteurs sont représentés comme des agents autonomes et rationnels en perpétuelle interaction pour satisfaire leurs propres objectifs comme leur temps d'attente ou leur temps de trajet par exemple. Dans la solution proposée, les agents conducteurs et passagers ont une perception modélisée dynamiquement en utilisant des R-Arbres. Nous modélisons leurs préférences en matière de détournement et de trajet et montrons l'impact de celles-ci sur la résolution d'une instance de covoiturage dynamique. Les résultats présentés montrent que notre système permet de traiter dynamiquement des requêtes complexes de passagers tout en minimisant l'impact du partage de trajet pour les conducteurs, et ce, pour un large spectre de préférences et de comportements.

Mots-clés : Covoiturage, Simulation, Optimisation, Agents

Abstract

In this paper, we propose a multi-agent approach to solve the dynamic multi-hop ridesharing problem. In our system, passengers and drivers are represented as autonomous and rational agents in perpetual interaction to satisfy their own objectives such as their waiting time or their travel time. In the proposed solution, driver and passenger agents have a dynamically modeled perception using R-Trees. We model their detour and route preferences and show the impact of these on the resolution of a dynamic ridesharing instance. The presented results show that our system dynamically handles complex passenger requests while minimizing the impact of ridesharing for drivers across a wide spectrum of preferences and behaviors.

Keywords: Ridesharing, Simulation, Optimization, Agents

1 Introduction

Le covoiturage est devenu un moyen de transport à part entière, que ce soit pour les trajets quotidiens ou pour les voyages plus longs. Outre son intérêt écologique, il permet aux conducteurs de réduire les coûts liés à leurs déplacements tout en offrant la possibilité aux passagers de voyager avec plus de flexibilité que les moyens de transport conventionnels. Il en résulte une utilisation accrue des véhicules déjà déployés sur la route, une réduction de la congestion du trafic et donc une diminution des émissions polluantes.

Nous proposons dans ce travail d'étudier le covoiturage dynamique entre particuliers. Un système de covoiturage dynamique prends en compte l'état actuel de l'environnement des utilisateurs pour effectuer des associations de covoiturage. Un tel système collecte en temps réel différents flux d'information dans l'environnement comme les itinéraires et temps de trajets des conducteurs en cours de route ainsi que les requêtes des passagers. Le caractère dynamique réside alors dans le flux continu de conducteurs et de passagers entrant, évoluant et sortant du système. Cette forme de covoiturage est majoritairement étudiée à l'échelle de grandes métropoles [10, 29] et dans le cadre de trajets courts et spontanés [30, 21].

La flexibilité d'un système de covoiturage dynamique peut être modélisée et quantifiée par deux opérations. D'une part, les conducteurs peuvent choisir de faire des détours pour prendre ou déposer des passagers, et ainsi élargir l'offre de transport du système de covoiturage. D'autre part, les passagers peuvent choisir de passer d'un véhicule à l'autre, et donc d'effectuer des transferts, afin d'arriver à leur destination en plusieurs étapes, si cela est plus pratique pour eux. Un exemple de ces opérations est illustré dans la Figure 1.

Dans cette figure, deux conducteurs d_i et d_j suivent leur itinéraire, respectivement en jaune et rouge, reliant leur nœud de départ v_s et leur

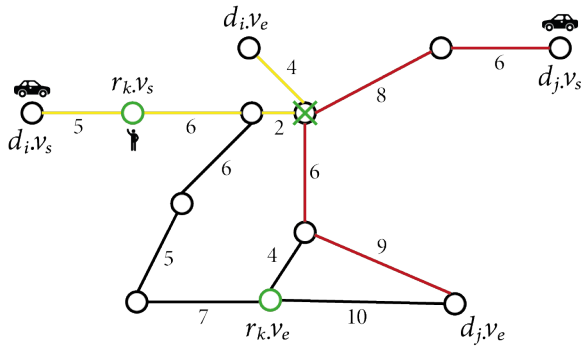


FIGURE 1 – Un exemple de covoiturage multi-saut

nœud d'arrivée v_e . Un passager r_k attend sur son nœud de départ $r_k.v_s$ et veut rejoindre son nœud d'arrivée $r_k.v_e$, tous deux entourés de vert. Les trajectoires des conducteurs se croisent sur le nœud avec une croix verte, un transfert est alors réalisable. Le conducteur d_i peut partager son trajet avec le passager r_k puis, une fois arrivé au nœud de transfert, le passager peut effectuer un transfert avec le conducteur d_j . Le conducteur d_j doit cependant faire un détour pour déposer le passager, ce qui rajoute 5 unités de temps à son temps de trajet.

Le défi d'un tel cadre est d'équilibrer l'impact du covoiturage sur l'offre et la demande. Cet équilibre implique de nombreux paramètres tels que le ratio conducteurs-passagers ou les détours acceptables par les utilisateurs en termes de temps et de distance. Le problème est alors d'associer au mieux passagers et conducteurs en exploitant leur marge de manœuvre afin de satisfaire les deux parties.

Un système de covoiturage peut être naturellement représenté par un système multi-agent, les conducteurs et les passagers étant des entités autonomes en interaction. C'est pourquoi nous proposons dans cet article un système multi-agent permettant de décrire des comportements spécifiques pour chaque utilisateur, garantissant le caractère hétérogène de notre simulation. La limite de perception de ces agents est définie dynamiquement par leurs préférences spatio-temporelles. Ils disposent ainsi d'une rationalité limitée.

Pour traiter l'important flux d'informations généré par un tel système, nous utilisons une structure d'indexation spatiale nommée R-Arbre [22]. Cette structure, et les algorithmes qui y sont associés, permet de stocker efficacement ces données et d'effectuer des requêtes de superposition

sur des points et des zones de l'espace. Les R-Arbres nous permettent aussi de réduire la complexité associée à l'identification du nœud de transfert optimal dans le cas d'un covoiturage multi-saut.

Après un état de l'art sur le problème du covoiturage et ses approches d'optimisation et de simulation dans la section 2, nous définissons notre problème de covoiturage en section 3. Nous proposons en section 4 une solution basée sur un système multi-agent dont le comportement des agents est défini par divers algorithmes. Enfin, en section 5 nous étudions différents comportements typiquement observables dans des scénarios réels de covoiturage. Nous les confrontons à notre modèle et étudions les résultats obtenus sur des cartes réelles.

2 Travaux connexes

À l'occasion d'un covoiturage, un passager peut être acheminé par un seul et unique conducteur, c'est ce que l'on nomme **le covoiturage simple saut** (ou *single-hop* en anglais). Le problème du covoiturage simple-saut consiste à identifier le meilleur conducteur pour acheminer un passager entre deux positions. C'est la première forme de covoiturage étudiée dans la littérature [19, 1, 10] de par ses similarités avec les problèmes d'allocation de ressources.

Il est aussi possible pour un passager d'utiliser plusieurs conducteurs, et donc d'effectuer des transferts afin d'arriver à sa destination. Cette variante s'appelle **le covoiturage multi-saut** (ou *multi-hop* en anglais) ou encore le covoiturage avec transferts [8]. Cette méthode permet d'améliorer les résultats obtenus avec le covoiturage simple-saut en augmentant les possibilités de covoiturage [17, 30]. Le covoiturage multi-saut implique une complexité accrue, car il faut cette fois identifier la meilleure combinaison de conducteurs ainsi que le lieu de transfert optimal pour tous les acteurs du covoiturage. Le problème du covoiturage multi-saut a été associé à de nombreux problèmes NP-Durs et NP-Complets de la littérature [7, 21].

Pour gérer cette complexité, l'espace de recherche doit être organisé et exploré de manière optimale. De nombreux travaux de recherche hiérarchisent l'espace [23] ou le divisent en zones pour en réduire la complexité [29, 28]. Ces divisions et réductions de l'espace sont cependant la plupart du temps effectuées de manière empirique et non dynamique forçant

une approximation importante de l'espace. Dans ce contexte, une structure d'indexation spatiale connue sous le nom de R-Arbre (*R-Tree* [15]) permet d'indexer dynamiquement les objets dans un arbre et d'optimiser les requêtes de superposition de zones ou de nœuds de l'espace [22]. Cette structure a notamment été utilisée pour le partage de taxi et s'est révélée être très efficace [30]. Dans nos travaux, nous proposons d'étendre le champ d'utilisation des R-Arbres au problème du covoiturage entre particuliers.

Il existe de nombreuses façons de modéliser le problème, notamment en ce qui concerne la granularité de la population. D'une part, l'approche centrée-groupe vise à optimiser des objectifs communs à l'ensemble de la population et donc à privilégier le système par rapport à l'individu [10, 1]. Cette méthodologie permet une simulation moins coûteuse en ressources mais s'avère éloignée de la réalité vis-à-vis du caractère hétérogène de la population. En revanche, l'approche centrée-individu cherche à résoudre la demande de chaque utilisateur, au cas par cas, en tenant compte de ses préférences pour définir les objectifs d'optimisation. Ainsi, chaque individu a son propre comportement et sa propre perception. Dans ce contexte, l'approche multi-agent [12] permet de modéliser un tel système en garantissant l'autonomie de chaque utilisateur et en permettant à la simulation d'être plus proche de la réalité [3, 19, 18, 24, 9].

Le problème du covoiturage a d'abord été étudié de manière mono-objectif avec des approches de programmation par contraintes [1, 10]. Des problèmes de covoiturage multi-objectif ont ensuite été étudiés et résolus à l'aide d'heuristiques évolutionnaires et génétiques [17, 16]. Ces méthodes permettent une optimisation plus rapide - bien qu'approximative - et donc plus applicable dans le monde réel. Avec l'avènement du covoiturage multi-saut et afin d'obtenir une solution exacte et multi-objectif, la recherche s'oriente vers une approche du problème de covoiturage via le problème du ou des plus proches voisins d'un groupe (GNN/ANN) [25]. En effet, puisque le multi-saut implique la recherche du nœud de transfert optimal, des heuristiques basées sur des algorithmes de recherche de plus proches voisins ont été développées [31] et ont été améliorées pour être plus efficaces dans un contexte réel [30, 6]. C'est dans cette perspective que nous proposons d'étendre l'algorithme de référence IER (Incremental Euclidean Restriction) [31] pour évaluer conjointement les solutions à sauts multiples et à sauts uniques de

manière multi-objectif.

3 Formulation du problème

3.1 Spécification du réseau routier et des agents du système

Le réseau routier de notre système est modélisé par un graphe orienté et connexe $G = \langle V, E \rangle$ dans lequel V est l'ensemble des nœuds du graphe et E est l'ensemble des arêtes liant ces nœuds. Les arêtes représentent des routes et les nœuds représentent des intersections de routes ou des impasses. Un nœud $v_i = (x, y)$ est référencé spatialement par un couple de coordonnées : une longitude x et une latitude y . Différents poids peuvent être attribués aux arêtes désignant le coût du trajet entre deux nœuds, comme le temps de trajet par exemple.

Nous utilisons une simulation en temps discret, le temps évolue à intervalles constants et le pas de simulation courant est dénoté par la variable *time*. À chaque pas de simulation, tous les passagers et conducteurs présents dans le système peuvent effectuer une action en fonction de leur comportement.

Un agent utilisateur du système $u_i = \langle v_s, v_l, v_e, det_{max} \rangle$, avec $u_i \in U$ représentant un passager ou un conducteur dans le système de covoiturage, est initialisé par quatre éléments : son nœud de départ v_s , son nœud de localisation actuel v_l , son nœud d'arrivée v_e et son facteur de détour det_{max} . Le facteur de détour quantifie la volonté de détour d'un agent utilisateur. Ainsi, s'il vaut 0, l'utilisateur ne souhaite pas faire de détour, et s'il vaut 1, l'utilisateur est prêt à doubler son trajet initial pour effectuer des détours. Nous faisons varier ce facteur entre 0 et 1 dans nos expériences. Cet élément permet de définir les limites initiales d'action et de perception d'un agent.

Un agent conducteur $d_i = \langle v_s, v_l, v_e, det_{max}, c_{max} \rangle$, avec $d_i \in D$, est initialisé par cinq éléments. Un conducteur étant un utilisateur, il hérite des quatre premiers paramètres définis précédemment. Le dernier paramètre c_{max} définit le nombre maximal de sièges disponibles dans le véhicule du conducteur. Un conducteur peut effectuer un nombre infini de détours et d'arrêts tant qu'il respecte son objectif d'atteindre sa destination avant son heure d'arrivée la plus tardive. Il suit donc son trajet de nœud en nœud, en prenant systématiquement le chemin le plus court. S'il doit faire un détour pour prendre et

déposer un passager, il doit être capable de déterminer si cette action est possible vis-à-vis de son heure d'arrivée la plus tardive. Pour ce faire, cet agent maintient un ordonnanceur de voyage où il stocke différentes informations telles que ses prochains arrêts *stops* (nœuds du graphe de route), les temps de trajets minimum *arr* et maximum *ddl*, les marges de détours *slk* ou les capacités restantes *c* (nombre de sièges disponibles) entre ses arrêts. La modélisation exacte de cet ordonnanceur est détaillée dans nos précédents travaux [13].

Un agent passager $r_i = \langle v_s, v_l, v_e, det_{max}, wt_{max}, pref \rangle$, avec $r_i \in R$, est initialisé par six éléments. Un passager étant un utilisateur du système, il hérite des quatre premiers paramètres définis précédemment. Le paramètre wt_{max} correspond au temps d'attente acceptable au maximum par le passager pour la totalité du trajet. S'il est dépassé, l'agent passager s'impatiente et disparaît du système, ce qui affecte le taux de service global du système de covoiturage. La perception p d'un agent passager représente la zone des conducteurs accessibles. Elle permet d'exclure les conducteurs trop éloignés pour le prendre en charge. Le paramètre $pref$ représente le vecteur de préférence du passager, c'est-à-dire son profil/comportement. Chaque élément $pref_i \in [0, \alpha]$ et la somme des éléments de $pref$ doit être égale à α : $sum(pref_i, \dots, pref_k) = \alpha$. On peut alors, par exemple, fixer α à 10 et faire varier la pondération des préférences d'un agent passager entre 0 et 10. De cette manière, nous pouvons attribuer un poids à chaque objectif contenu dans cette liste de préférences et détailler différents profils de passagers. Ceci implique que, si $pref_i > pref_j$, alors le passager accorde plus d'importance à la préférence $pref_i$ qu'à la préférence $pref_j$ lors de l'optimisation de son trajet. Un passager a donc pour objectif d'atteindre sa destination avant son heure d'arrivée la plus tardive tout en sélectionnant le ou les meilleurs covoiturages en fonction de ses préférences.

Un agent service de transport $t_{sa_i} = \langle RT_p \rangle$ est responsable de la connexion des agents utilisateurs. Il remplit deux fonctions. La première consiste à stocker et à mettre à jour toutes les perceptions des agents conducteurs, à leur demande, lorsqu'un événement entraîne une modification de leur perception. La seconde fonction est de répondre aux requêtes spatiales des agents passagers du système qui souhaite identifier des candidats au covoiturage. L'agent service de transport est donc implémenté comme un "tableau noir" afin de mettre à jour et de transmettre

les informations des agents du système lorsque cela est demandé.

3.2 Contraintes et Formulations

Nous limitons notre étude au covoiturage à un et deux sauts, c'est-à-dire au covoiturage limité à deux conducteurs pour un transfert. En effet, plusieurs articles comme [23] ont montré que l'augmentation du nombre de transferts possibles améliore à peine les performances d'un système de covoiturage tandis que la complexité du problème explose exponentiellement. Dans cet article, chaque requête de covoiturage est associée à un seul et unique passager.

Pour qu'un covoiturage entre un passager et un conducteur soit réalisable, nous définissons plusieurs contraintes spatio-temporelles :

- **Contrainte temporelle** : Les conducteurs et les passagers doivent arriver à leur destination avant leur heure d'arrivée souhaitée. Un passager doit être pris en charge avant que son temps d'attente maximum ne soit atteint.
- **Contrainte de capacité** : La capacité courante d'un conducteur ne doit jamais dépasser la capacité maximale de son véhicule.
- **Contrainte de l'ordre des arrêts** : Un conducteur partageant son itinéraire avec un passager doit d'abord passer par le nœud de prise en charge du passager avant de visiter le nœud de dépôt de ce dernier.

Une association ("*a match*" en anglais) $m \in r_i.M$ est définie par plusieurs variables. Un ou deux conducteurs $d_j, d_k \in D$ (simple-saut : $j = k$, multi-saut : $j \neq k$); un nœud de transfert v_{tsf} s'il y a un transfert; et enfin quatre critères : un nombre de transferts nb_{tsf} (0 ou 1 dans cette étude), un temps d'attente total $total_{wt}$, un temps d'arrivée arr_{time} , et une distance additionnelle de détour add_{dist} . Chaque passager a pour but de choisir l'association la plus optimale dans M selon ses préférences $pref$. Le problème étudié est donc multi-objectif en impliquant la minimisation simultanée de ces quatre critères.

La complexité de notre problème peut être résumée par : quel est le nombre de combinaisons de covoiturage possible? Avec une approche individu-centrée cette complexité peut être formalisée comme suit :

$$O(R * H^{D*N}) \quad (1)$$

avec H : le nombre de sauts possibles; D : le nombre d'agents conducteurs; R : le nombre

d'agents passagers ; N : le nombre de nœuds du graphe de route. Ce problème est ainsi associé aux problèmes de classe de complexité exponentielle : $O(2^n)$.

4 Le système multi-agent proposé

À chaque étape de la simulation, le tour d'un agent utilisateur est divisé en deux phases : la phase de mise à jour de sa perception de l'environnement et la phase de prise de décision.

4.1 La mise à jour de la perception des agents

La perception d'un agent utilisateur représente ce qu'il est capable d'utiliser dans l'environnement pour atteindre son objectif. Dans nos travaux, cette perception est dynamique et doit être mise à jour régulièrement.

Pour établir sa perception, l'agent utilisateur collecte d'abord les nœuds du plus court chemin entre son origine et sa destination. Il récupère ensuite les latitudes et longitudes minimales et maximales de tous ces nœuds pour définir le rectangle de délimitation minimum (MBR) de sa perception. Cette perception initiale est étendue en y ajoutant la marge de temps de détour ou d'attente restante de l'agent afin de former le rectangle englobant (BBOX) de sa perception. Ce processus est basé sur nos travaux antérieurs [13] et est illustré dans la Fig.2.

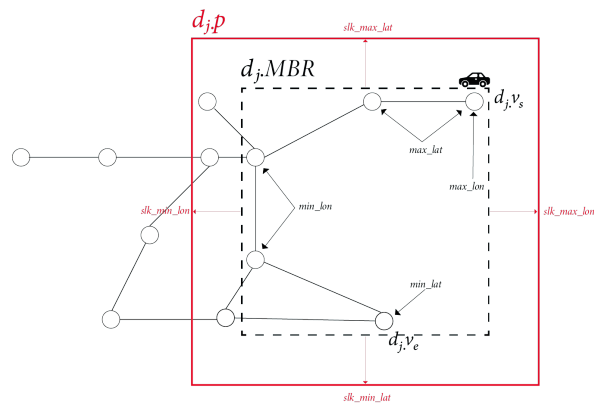


FIGURE 2 – Exemple de calcul de la perception d'un agent conducteur.

Cette méthode est utilisée pour la perception d'un conducteur, entre chaque arrêt de son ordonnanceur, et pour la perception d'un passager, de son point de départ à son point d'arrivée. Cette perception est approximative, ainsi,

un test de faisabilité plus rigoureux, basé sur les contraintes de la section 3.2, est nécessaire avant tout covoiturage.

Un agent conducteur ne met à jour sa perception que dans deux cas : (1) s'il atteint un nœud du graphe de route ; (2) si un contrat de covoiturage impliquant un détour est conclu avec un passager. Un agent passager met à jour sa perception à chaque étape de la simulation. Si un nouveau conducteur apparaît dans sa perception, il cherche à déterminer s'il peut lui permettre de rejoindre sa destination.

4.2 L'identification du covoiturage optimal

Le processus d'identification du covoiturage optimal est résumé dans l'algorithme 1. Comme expliqué précédemment, un conducteur peut permettre à un passager de rejoindre sa destination de deux façons : (1) simple-saut : en assurant la totalité du trajet (2) multi-saut : en lui offrant la capacité d'effectuer un transfert avec un autre conducteur. Pour identifier les conducteurs candidats à un covoiturage, un passager effectue des requêtes à l'agent service de transport. Un conducteur candidat au covoiturage est un conducteur dont au moins l'une des perceptions chevauche la perception du passager ainsi qu'au moins l'un de ces deux éléments : le nœud de départ et/ou d'arrivée du passager (PMO L1-3). Ainsi, un conducteur dont la perception chevauche les deux nœuds est candidat au simple-saut, tandis que si sa perception ne chevauche que l'un des nœuds, il est nécessaire de trouver un autre conducteur complémentaire pour espérer effectuer covoiturage multi-saut. Un conducteur est complémentaire à un autre s'il chevauche une partie de sa perception, la zone de rencontre des perceptions des deux conducteurs et du passager est appelée zone de transfert (voir Figure 3).

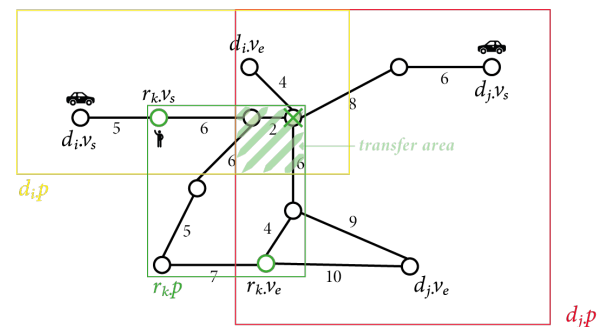


FIGURE 3 – Identification des candidats au covoiturage à l'aide des différentes perceptions.

L'évaluation d'un covoiturage simple saut est peu complexe. Le passager évalue les différents objectifs relatifs au trajet et établit une valeur indiquant la performance de cette solution en fonction de ses préférences (PMO L7-10). La fonction multi-objectif responsable de cette évaluation est détaillée plus tard dans cet article. À contrario, l'évaluation d'un covoiturage multi-saut induit une complexité très importante. Il faudrait effectuer cette même méthode pour chaque nœud de transfert possible pour les conducteurs. Il n'est pas envisageable de couvrir toutes ces possibilités. C'est pourquoi nous avons développé une heuristique, nommée *CIER Constrained Incremental Euclidean Restriction* visant à prioriser l'évaluation des nœuds de transferts les plus proches du passager et des conducteurs (PMO L12-17)[14].

Cette heuristique est basée sur l'algorithme IER (*Incremental Euclidean Restriction*) [26] permettant la résolution du problème des plus proches voisins d'un groupe (ANN/GNN) [25]. Ce problème est semblable au problème de l'identification du nœud de transfert optimal auquel nous sommes confrontés. L'objectif est d'identifier le(s) nœud(s) d'un graphe le(s) plus proche(s) d'un groupe d'autres nœuds de ce même graphe. Un exemple simple est le problème du restaurant : quel est le restaurant le plus proche de trois individus éparpillés dans une ville ? Les nœuds de transferts sont alors les restaurants et les conducteurs et le passager sont les individus. La notion du plus proche peut alors être complexe : minimisant la somme des distances (utilitaire), la moyenne des distances (égalitaire)...

Nous avons choisi une approche utilitariste ayant pour objectif de minimiser la distance additionnelle de détour occasionnée par le covoiturage. L'intérêt de cette approche est de limiter l'impact du covoiturage sur les conducteurs afin de conserver leur marge de détour pour les autres passagers du système. L'étude et la comparaison d'autres politiques feront certainement l'objet d'une étude de notre part. Une fois le nœud de transfert optimal identifié, le passager évalue la solution de la même façon que pour un covoiturage simple-saut.

Pour diversifier l'offre de covoiturage proposée à un passager, nous proposons une solution basée sur le mécanisme de dominance de Pareto [27]. L'idée générale est de maintenir une liste des meilleures solutions de covoiturage *best_matches*. Une meilleure solution de covoiturage est une solution minimisant au moins

un des objectifs présentés dans la section 3.2 (temps d'attente, heure d'arrivée, distance additionnelle et nombre de transferts). Elle est, dans ce cas, Pareto dominante et est ajoutée à la liste *best_matches*. De cette façon, nous préservons des solutions optimales pour chaque objectif et le passager peut, selon ses préférences, choisir celle qui lui convient le mieux. Si une solution de covoiturage optimise tous les objectifs, alors elle est Pareto optimale, la liste est alors vidée et cette solution lui est ajoutée (PMO, L19-24). Le nombre de solutions optimales est ainsi compris entre 0, quand il n'y a pas de solution, et 4 si il existe une solution pareto dominante par objectif.

En plus de cette liste des meilleures solutions, nous maintenons deux autres listes : la liste des pires valeurs *worst_goals* et des meilleures valeurs *best_goals* atteintes pour chaque objectif. Ces listes servent d'intervalles de référence à un agent passager pour évaluer la performance d'une solution. En effet, en effectuant une différence proportionnelle pondérée par son vecteur de préférence, l'agent passager obtient un pourcentage de performance pour chaque objectif ainsi qu'en moyenne. Ce pourcentage est calculé à l'aide de la formule 2 :

$$\frac{\sum_i^k (1 - \frac{abs(candidate_goals_i - worst_goals_i)}{worst_goals_i - best_goals_i} * 100) * pref_i}{k + sum(pref)} \quad (2)$$

Avec k le nombre d'objectifs à considérer. De cette façon, un objectif plus important pour le passager a plus d'impact sur l'évaluation finale de la solution. Le passager sélectionne enfin la solution la plus performante (PMO, L28).

5 Expérimentations et résultats

Dans cette section, nous proposons de simuler différents comportements de passagers et de conducteurs à l'aide du modèle agent présenté dans cet article. Les résultats présentés montrent l'impact de ces comportements sur des instances de problème de covoiturage.

5.1 Création des profils utilisateurs

Nous appelons "*un profil utilisateur*" un ensemble de préférences visant à simuler un comportement réel. Pour les conducteurs, nous faisons essentiellement varier la préférence de détour det_{max} entre 0 (c'est-à-dire pas de détour) et

Algorithm 1 PMO - Preferential Matching Optimization

Input : G = road infrastructure graph, r_i = a rider agent
Output : $best_match$ = the best match

```

1:  $candidates\_s$  : drivers which perception overlaps starting node
2:  $candidates\_e$  : drivers which perception overlaps ending node
3:  $best\_add\_dist, best\_wt, best\_arr\_time, best\_nb\_tsf := \infty$ ,
4:  $best\_matches := list() \#r_i.M$ 
5: for all  $d\_s \in candidates\_s$  do
6:   for all  $d\_e \in candidates\_e$  do
7:     if  $d\_s.id = d\_e.id$  and match is feasible then
8:       #single hop
9:       eval  $m.total\_wt, m.arr\_time, m.nb\_tsf, m.add\_dist$ 
10:      set match driver  $m.d = d\_s$ 
11:     else
12:       #multi hop, CIER heuristic
13:        $Q$  : set of drivers et rider position nodes
14:        $T$  : set of candidate transfer nodes
15:       find :  $t \in T$  minimizing aggregated distance from  $Q$  and feasible
16:       eval :  $m.total\_wt, m.arr\_time, m.nb\_tsf, m.add\_dist$ 
17:       set match drivers :  $m.d_0 = d\_s, m.d_1 = d\_e, m.v\_tsf = t$ 
18:     end if
19:     if  $m$  minimizes all current best values of the objectives (pareto optimal) then
20:       overwrite  $best\_matches := set(m)$  #the only element of the
21:       set is now  $m$ 
22:       update all objectives :
23:        $best\_add\_dist, best\_wt, best\_arr\_time, best\_nb\_tsf$ 
24:     else if  $m$  minimizes at least one objective (pareto dominant) then
25:       add  $m$  to  $best\_matches$ 
26:       update concerned objectives
27:     end if
28:   end for
29: end for
30:  $best\_match = best$  match among  $best\_matches$  according rider preferences
31: return  $best\_match$ 

```

1, correspondant à une capacité de détour de la taille du trajet initial du conducteur. Nous pouvons ainsi générer tout un spectre de comportement de conducteurs, allant du plus égoïste au plus altruiste. Concernant les passagers, nous définissons 5 profils d'utilisateurs présentés dans la Table 1 en fonction des 4 objectifs présentés en section 3.2.

TABLE 1 – Tableau des différents profils des agents passagers

Profil	$total_wt$	add_dist	arr_time	tsf_nb
Équilibré	3	3	3	1
Pluie	7	1	1	1
Écologique	1	7	1	1
Pressé	1	1	7	1
Confort	1	1	1	7

Chaque valeur de ce tableau représente le poids associé à chaque objectif : plus le poids est élevé, plus le profil a de l'intérêt pour l'objectif concerné. Concrètement, on peut imaginer un curseur ajustant les poids de ces objectifs dans une application smartphone de covoiturage dynamique. Le profil "équilibré" représente un passager moyen, sans réelles préférences, et sert de profil de référence. Le profil "pluie" représente un agent passager sous la pluie qui souhaite être

pris en charge rapidement. Il donne alors la priorité à la minimisation du temps d'attente total du trajet. Le profil "écologiste" représente un passager préoccupé par l'impact environnemental lié à la conduite d'une voiture. Ce profil privilégie le partage de trajet qui implique peu de détours et cherche donc à minimiser la distance additionnelle de détour liée au covoiturage. Le profil "pressé" représente un passager qui souhaite atteindre sa destination le plus rapidement possible. Ce profil donne la priorité à la minimisation de l'heure d'arrivée à sa destination. Enfin, le profil "confort" souhaite éviter le désagrément de devoir changer de véhicule au cours d'un trajet. Ce profil donne la priorité aux covoiturages simple-sauts et minimise donc le nombre de transferts de véhicules.

5.2 Protocole expérimental

Nous avons privilégié l'étude de petites instances à de nombreuses reprises afin d'obtenir des résultats reproductibles et fiables. De cette manière, les biais liés aux paramètres du système sont contrôlés. Nous effectuons une simulation avec une génération continue de conducteurs nous permettant de reproduire une situation réelle où il y a un flux entrant et sortant de conducteurs dans le système. En variant la densité de ce flux, nous pouvons simuler tout un panel de trafics routiers et évaluer les comportements en situation de rareté ou d'abondance de l'offre de covoiturage. La résolution de l'instance se termine lorsque tous les agents passagers ont quitté le système : en atteignant leur destination ou en s'impatientant.

Nos instances sont composées d'un graphe de 150 nœuds provenant d'un réseau routier réel de la ville de San Francisco, extrait d'Open Street Map et modélisé à l'aide de la bibliothèque OSMNX [5]. Ce graphe est illustré dans la Figure 4. Sur ce graphe en grille, 30 conducteurs et 20 passagers interagissent. Leurs nœuds de départ et de destination sont générés de manière aléatoire et uniforme. Les itinéraires initiaux des conducteurs sont dérivés du plus court chemin entre leurs nœuds de départ et d'arrivée. Nous supposons que les passagers ont un temps d'attente maximal de 3 minutes et un facteur de détour fixé à 0,2 pour créer une situation difficile.

5.3 Résultats

Dans cette sous-section, nous détaillons les résultats obtenus avec notre modèle. L'objectif est

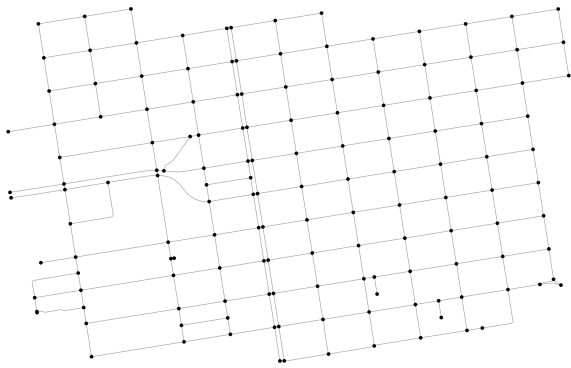


FIGURE 4 – Le graphe étudié : une section du côté nord de la ville de San Francisco

de mettre en évidence l'impact des différentes stratégies sur des indicateurs tels que : le temps d'attente moyen, le taux de service (succès du covoiturage), la distance additionnelle moyenne et le nombre moyen de véhicules nécessaires au covoiturage. Chaque figure comprend plusieurs courbes correspondant aux profils des passagers mentionnés précédemment. Chaque point d'une courbe est le résultat de 50 expériences moyennées. Ces expériences correspondent aux 50 mêmes expériences pour chaque profils de passager. Enfin, l'axe des abscisses est indexé par le facteur de détournement des conducteurs.

L'impact du profil "écologiste". La Figure 5 montre comment le profil "écologiste" (courbe verte/triangles) minimise la distance additionnelle moyenne de détournement des conducteurs par rapport aux autres profils. Ce phénomène est amplifié par l'augmentation du facteur de détournement des conducteurs. On observe ainsi des détours 2 à 3 fois moins importants avec ce profil qu'avec les profils concurrents. En conséquence, les autres objectifs sont très largement négligés par ce profil (Figures 6, 7, 8).

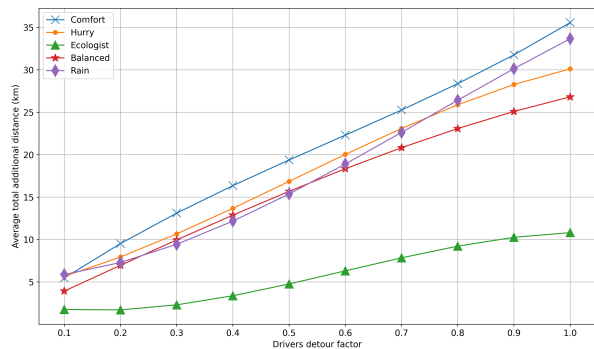


FIGURE 5 – Distance additionnelle moyenne de détournement des conducteurs

L'impact du profil "confort". Le profil "confort" (courbe bleue/croix) dans la Figure 6, vise à minimiser le nombre de transferts (changements de véhicules) pour atteindre la destination cible. On constate que ce profil est capable de minimiser sa cible de plus en plus (environ de 1.4 à 1.1 véhicule impliqué en moyenne) avec l'augmentation du facteur de détournement des conducteurs. Il est également bien meilleur que les autres profils en montrant entre 20% et 70% de performance supplémentaire sur cet objectif. En revanche, il est le plus faible pour la distance de détournement supplémentaire (Figure 5) et est similaire à la majorité des autres profils pour les autres objectifs (Figures 7 et 8). Nous pouvons conclure ici que la réduction du nombre de transferts entre véhicules et la favorisation du covoiturage simple-saut augmente la distance de détournement.

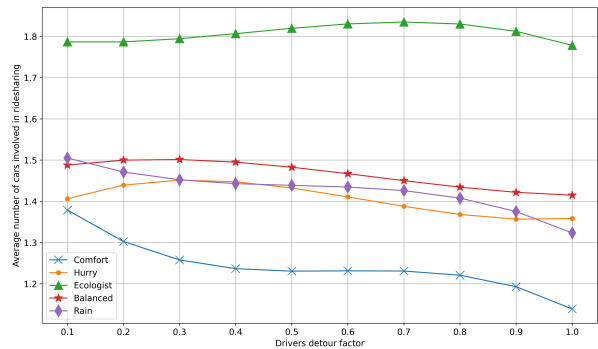


FIGURE 6 – Nombre moyen de véhicules impliqués dans le covoiturage

L'impact des profils "pluie", "pressé" et "équilibré". Nous avons choisi de traiter ces trois profils conjointement, car, dans la majorité des cas, la tendance des courbes et leurs résultats sont similaires. Cela s'explique par la difficulté de dissocier le temps de parcours, le temps d'attente et la distance de parcours. Ces derniers sont en effet étroitement liés et interdépendants. Néanmoins, le profil "pluie" (courbe mauve/losanges) réussit à réduire son temps d'attente (Figure 7) quand les conducteurs sont peu disposés à effectuer des détours (facteur inférieur à 0.5). Il se confond ensuite avec les autres profils.

Le profil "Pressé", cherchant à arriver le plus rapidement possible, vise indirectement à réduire son temps d'attente. C'est pourquoi nous le voyons se positionner très légèrement comme le leader de la minimisation du temps d'attente sur la Figure 7 pour un facteur de détournement du conducteur de 0.5 à 0.9. Notre profil "équilibré"

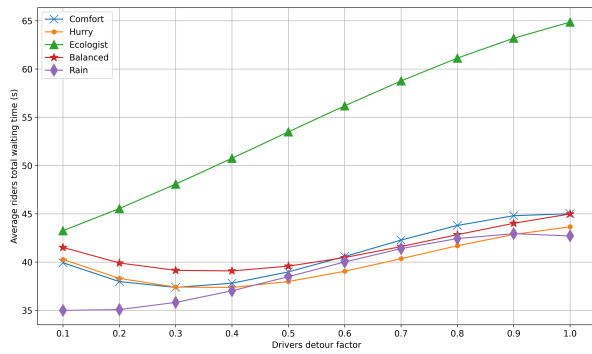


FIGURE 7 – Temps d'attente moyen d'un passager

remplit parfaitement son objectif d'être le juste milieu entre les différents profils et leurs objectifs, comme on peut le voir sur la totalité des figures.

La performance globale des profils : le taux de service. Pour conclure sur l'analyse de ces résultats, nous pouvons considérer la performance des profils du point de vue du taux de service moyen (Figure 8). Le taux de service représente la part des passagers qui ont réussi à se rendre à leur destination en covoiturant.

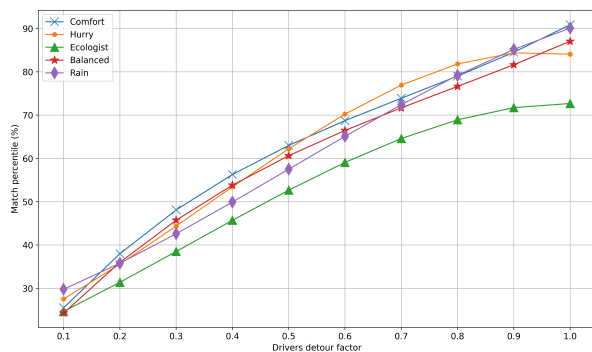


FIGURE 8 – Taux de service moyen du système

On constate tout de suite que plus le facteur de détour augmente, plus il existe de solutions potentielles et donc plus le taux de service augmente. Ce phénomène est normal et démontre le bon fonctionnement de notre modèle. Nous remarquons que le profil "écologiste" (triangles verts) présente un taux de service relativement faible par rapport au reste des profils avec une différence moyenne de 5 à 15%. Aussi, il représente le pire taux de service atteint avec environ 72% pour un facteur de détour des conducteurs de 1. Si on émet l'hypothèse que les utilisateurs qui souhaitaient être passagers ont finalement pris leur véhicule pour rejoindre leur destination, il se peut que ce profil ne permette pas de

réduire, autant qu'espéré, les émissions globales du système, curieux paradoxe !

6 Conclusion et perspectives

Dans cet article, nous proposons un système multi-agent pour optimiser le covoiturage dynamique en fonction des préférences des conducteurs et des passagers. Nous modélisons les différents acteurs d'un système de covoiturage et leur spectre de comportements possibles. Nous apportons une approche originale, basée sur des techniques d'indexation spatiale, pour modéliser la perception de nos agents et pour réduire le coût computationnel associé à la recherche du covoiturage optimal. Les expériences présentées ont montré que l'approche proposée permet une optimisation individuelle et efficace.

Cette approche révèle certaines limitations pour les profils favorisant le temps d'attente et le temps de trajet. En effet, l'optimisation par agrégation des pondérations semble insuffisante lorsque les objectifs sont interdépendants. Pour répondre à ce constat, d'autres techniques peuvent être envisagées, comme l'agrégation par l'intégrale de Choquet, qui s'est avérée efficace pour l'optimisation multicritère [4]. Une autre perspective consiste à rendre le comportement des agents dynamique. En effet, le comportement des agents est pour l'instant défini à leur apparition et ne change pas au cours du temps. En fonction de l'état de leur environnement (trafic, météo, pénurie d'offres ou de demandes, etc.), leur comportement pourrait évoluer et les mener à changer leurs objectifs principaux [2]. En étendant notre approche à des agents au comportement dynamique, nous nous rapprocherions encore davantage du monde réel et de ses caractéristiques.

Des critères et mécanismes tels que le nombre d'arrêts maximum souhaité par un conducteur ou encore la possibilité de réserver un seul covoiturage pour plusieurs personnes sont essentiels dans le cadre d'une mise en application de notre solution. Il seront inclus et étudiés dans nos prochains articles.

Enfin, il serait fructueux de considérer également la complémentarité avec d'autres modalités de transport, en particulier les transports publics tels que le bus et le métro [20, 11]. Nous passerions alors de la mono-modalité à la comodalité, c'est-à-dire la multimodalité associée au covoiturage.

Références

- [1] N. Agatz, A. L. Erera, M. W. Savelsbergh, and X. Wang. Dynamic Ride-Sharing : A Simulation Study in Metro Atlanta. *Procedia - Social and Behavioral Sciences*, 17 :532–550, 2011.
- [2] F. Balbo and S. Pinson. Dynamic modeling of a disturbance in a multi-agent system for traffic regulation. *Decision Support Systems*, 41(1) :131–146, 2005.
- [3] H. A. N. C. Bandara and D. Dias. A multi-agent system for dynamic ride sharing. In *2009 International Conference on Industrial and Information Systems (ICIIS)*, page 199–203. IEEE, Dec 2009.
- [4] S. Ben Cheikh-Graiet, M. Dotoli, and S. Hammadi. A Tabu Search Based Metaheuristic for Dynamic Carpooling Optimization. *Computers & Industrial Engineering*, Feb. 2020.
- [5] G. Boeing. OSMnx : New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65 :126–139, Sept. 2017.
- [6] Z. Chen, B. Yao, Z.-J. Wang, X. Gao, S. Shang, S. Ma, and M. Guo. Flexible Aggregate Nearest Neighbor Queries and its Keyword-Aware Variant on Road Networks. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.
- [7] J.-F. Cordeau and G. Laporte. The dial-a-ride problem : Models and algorithms. *Ann Oper Res*, page 18, 2007.
- [8] C. Cortés, M. Matamala, and C. Contardo. The pickup and delivery problem with transfers. *European Journal of Operational Research*, 200(3) :711–724, Feb. 2010.
- [9] A. Daoud, F. Balbo, P. Gianessi, and G. Picard. Un modèle agent générique pour la comparaison d'approches d'allocation de ressources dans le domaine du transport à la demande. In *JFSMA29*, pages 127–136. Cépaduès, 2021.
- [10] A. Di Febbraro, E. Gattorna, and N. Sacco. Optimization of Dynamic Ridesharing Systems. *Transportation Research Record : Journal of the Transportation Research Board*, 2359(1) :44–50, Jan. 2013.
- [11] A. O. Diallo, G. Lozenguez, A. Doniec, and R. Mandiau. Usage des parkings relais dans les comportements de déplacements intermodaux. In *JFSMA29, June 28-30, 2021*, pages 83–92. Cépaduès, 2021.
- [12] J. Ferber. Multi-agent systems : An introduction to distributed artificial intelligence, 1999.
- [13] C. Fevre, H. Zgaya-Biau, P. Mathieu, and S. Hammadi. Multi-agent Systems and R-Trees for Dynamic and Optimised Ridesharing. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1352–1358, Oct. 2021.
- [14] C. Fevre, H. Zgaya-Biau, P. Mathieu, and S. Hammadi. Preferential Optimization of Multi-hop Dynamic Ridesharing based on R-Trees and Multi-Agent Systems. *Under review*, 2022.
- [15] A. Guttman. *R Trees : A Dynamic Index Structure for Spatial Searching*, volume 14. Jan. 1984.
- [16] W. Herbawi and M. Weber. Evolutionary Multiobjective Route Planning in Dynamic Multi-hop Ridesharing. In *Evolutionary Computation in Combinatorial Optimization*, pages 84–95, Berlin, Heidelberg, 2011. Springer.
- [17] W. Herbawi and M. Weber. Modeling the Multi-hop Ridematching Problem with Time Windows and Solving It Using Genetic Algorithms. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 1 :96, Nov. 2012.
- [18] K. Jeribi, H. Mejri, H. Zgaya, and S. Hammadi. Distributed graphs for solving co-modal transport problems. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1150–1155, Oct. 2011.
- [19] E. Kamar and E. Horvitz. Collaboration and Shared Plans in the Open World : Studies of Ridesharing. In *IJCAI International Joint Conference on Artificial Intelligence*, page 187, Jan. 2009.
- [20] J. Lin, S. Sasidharan, S. Ma, and O. Wolfson. A Model of Multimodal Ridesharing and Its Analysis. In *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, pages 164–173, Porto, June 2016. IEEE.
- [21] S. Ma and O. Wolfson. Analysis and evaluation of the slugging form of ridesharing. In *SIGSPATIAL'13*, pages 64–73, Orlando, Florida, 2013. ACM Press.
- [22] Y. Manolopoulos, Y. Theodoridis, and V. J. Tsotras. Spatial Indexing Techniques. In *Encyclopedia of Database Systems*, pages 1–7. Springer New York, New York, NY, 2014.
- [23] N. Masoud and R. Jayakrishnan. A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system. *Transportation Research Part B : Methodological*, 106 :218–236, Dec. 2017.
- [24] P. Mathieu and A. Nongaillard. Effective evaluation of autonomous taxi fleets. In *ICAART (1)*, pages 297–304, 2018.
- [25] D. Papadias, Y. Tao, K. Mouratidis, and C. K. Hui. Aggregate nearest neighbor queries in spatial databases. *ACM Transactions on Database Systems*, 30(2) :529–576, June 2005.
- [26] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. Query Processing in Spatial Network Databases. In *Proceedings 2003 VLDB Conference*, pages 802–813. Elsevier, 2003.
- [27] V. Pareto. *Cours d'économie politique*. Librairie Droz, 1964.
- [28] Shuo Ma, Yu Zheng, and O. Wolfson. T-share : A large-scale dynamic taxi ridesharing service. In *ICDE29'*, pages 410–421, Brisbane, QLD, Apr. 2013. IEEE.
- [29] A. Tafreshian and N. Masoud. Trip-based graph partitioning in dynamic ridesharing. *Transportation Research Part C : Emerging Technologies*, 114 :532–553, May 2020.
- [30] Y. Xu, L. Kulik, R. Borovica-Gajic, A. Aldwyish, and J. Qi. Highly Efficient and Scalable Multi-hop Ride-sharing. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pages 215–226, Seattle WA USA, Nov. 2020. ACM.
- [31] M. Yiu, N. Mamoulis, and D. Papadias. Aggregate nearest neighbor queries in road networks. *IEEE Transactions on Knowledge and Data Engineering*, 17 :820–833, July 2005.

Taxonomie des vulnérabilités liées aux incitations dans les blockchains.

Hector Roussille^{a, b} Önder Gürçan^a Fabien Michel^b
 hector.roussille@cea.fr onder.gurcan@cea.fr fmichel@lirmm.fr

^aUniversité Paris-Saclay, CEA, List, F-91120, Palaiseau, France.

^bUniversité de Montpellier, LIRMM, CNRS, France.

Résumé

Ce papier présente une taxonomie des vulnérabilités d'incitations qui peuvent affecter les blockchains publiques et consortium. Cette taxonomie a pour but d'aider les chercheurs et développeurs à mieux comprendre les différentes menaces qui pèsent sur ces systèmes afin de les rendre plus résilients. Dans ce but, la taxonomie repose sur un modèle blockchain organisationnel et multi-agent (AGR4BS) et expose clairement les liens entre vulnérabilités et les différents rôles que peuvent jouer les participants, elle se veut exhaustive, mais toujours centrée sur les causes des déviations plutôt que sur leurs conséquences. Les vulnérabilités sont des déviations de comportement par rapport à leurs définitions nominales, elles sont catégorisées par rapport aux rôles et comportements identifiés dans AGR4BS.

Mots-clés : Multi-Agent, Blockchain, Vulnérabilités, Taxonomie

Abstract

This paper presents a taxonomy of incentive vulnerabilities that can affect public and consortium blockchain-based networked intelligent systems. The taxonomy aims to help researchers and developers better understand the related threats and design more secure systems. To this end, the proposed taxonomy is grounded in a generic multi-agent organizational model for blockchain systems (AGR4BS) and establishes a relationship between the vulnerabilities and the dedicated agent roles. It is exhaustive but focused on the root causes of the deviations rather than on their consequences. We expressed the vulnerabilities as behavior deviations and classified them according to the roles and behaviors identified in AGR4BS to form the categories and refine the subcategories of the taxonomy.

Keywords: Multi-Agent, Blockchain, Vulnerabilities, Taxonomy

1 Introduction

Les systèmes blockchain connectent des participants intelligents et leur permettent d'interagir et de coopérer. Ces systèmes ont créé de nouvelles opportunités d'innovation dans plusieurs secteurs comme les maisons ou villes intelligentes, les chaînes d'approvisionnement et la finance. La blockchain apporte un moyen sécurisé et transparent de stocker et gérer des données, et se positionne d'ores et déjà comme une technologie essentielle pour connecter différents systèmes dans une architecture décentralisée.

La blockchain est particulièrement attractive dans le sens où elle permet de maintenir un registre public, immuable et ordonné de transactions garantissant l'auditabilité. Les gains de popularité récents de la blockchain motivent le développement de nouveaux outils ou applications principalement financiers, allant de la simple crypto monnaie aux applications décentralisées, ce qui attire les investisseurs particuliers et, plus récemment, professionnels. Au fur et à mesure que ces systèmes se développent et gagnent en popularité, de plus en plus de participants les rejoignent, car ils sont motivés financièrement par plusieurs mécanismes. Cependant, les systèmes blockchain sont vulnérables, les conséquences d'une attaque peuvent aller d'un simple ralentissement, au vol de plusieurs millions de dollars ou à l'arrêt pur et simple du système.

Récemment, de nombreuses attaques ou exploitations ont ciblé les systèmes blockchain [15]. Les systèmes blockchain sont des projets majoritairement open-source, ainsi, les attaquants ont accès aux implémentations, ce qui facilite une exploitation, car leur cible n'est pas une boîte noire. De plus, dans le cas des blockchains publiques, ils peuvent entrer ou sortir du système sans aucune restriction.

Les blockchains sont aussi des systèmes socio-économiques [9], des participants malicieux

peuvent exploiter des vulnérabilités d'incitations existantes. Les motivations sont des mécanismes de récompense ou de punition qui permettent de guider les comportements rationnels vers les comportements nominaux attendus.

Dans la littérature, une vulnérabilité est un défaut qui peut produire des comportements incorrects et non désirés. Ainsi, une *vulnérabilité d'incitations* peut être définie comme un non-alignement entre le comportement d'un agent, tel qu'attendu par le protocole, et le comportement rationnel qu'impose une interprétation utilitaire des mécanismes d'incitations. Une telle vulnérabilité motive la déviation comportementale pour tout participant rationnel, ce qui la rend particulièrement dangereuse. De plus, une exploitation est un processus par lequel une ou plusieurs vulnérabilités sont exploitées pour attaquer le système avec des intentions malveillantes ou pour optimiser une métrique égoïste pouvant avoir des conséquences similaires, mais pas d'intention malveillante. Identifier ces vulnérabilités est essentiel pour empêcher des attaques, mais la diversité et les multiples interactions des participants rendent cette tâche non triviale. De plus, certains participants peuvent attaquer le système sans montrer la moindre rationalité par rapport à ces mécanismes d'incitations. Comprendre les relations entre vulnérabilités et motivations est donc cruciale dans le but de sécuriser durablement les systèmes blockchain. Une telle compréhension peut faciliter le développement d'un cadre de travail, exposé, par exemple, comme un ensemble d'environnements blockchain permettant de tester différentes approches multi-agents (*ex.*, l'apprentissage par renforcement) pour évaluer la sécurité des systèmes blockchain et automatiser la découverte d'attaques potentielles.

Dans ce cadre, ce papier présente une taxonomie des vulnérabilités d'incitations pour les systèmes blockchain, construite autour du concept de rôle. Nous nous concentrons principalement sur les blockchains publiques et consortiums, vu que les blockchains privées ne requièrent pas nécessairement d'incitations intrinsèques au système. Cette taxonomie est basée sur un modèle multi-agent organisationnel pour les systèmes blockchain appelé AGR4BS [14] composé de trois principaux niveaux d'abstraction : Agent, Groupe et Rôle. AGR4BS permet d'identifier les comportements de chaque rôle qui sont sujets à une ou plusieurs déviations pouvant donner lieu à une exploitation. Les contributions de cet article sont les suivantes :

- Nous explorons de manière systématique les

vulnérabilités d'incitations dans les systèmes blockchain, en lien avec les rôles joués par les participants.

- Nous identifions les déviations possibles pour chaque rôle et les associations avec des vulnérabilités connues ou suspectées.
- Nous proposons une hiérarchie des différentes déviations identifiées par rapport à leurs impacts potentiels ainsi que leur faisabilité.
- Nous comparons la littérature existante avec une attention particulière pour les taxonomies concernant la sécurité des systèmes blockchain.

Cet article est organisé comme suit : la section 2 introduit les concepts essentiels utilisés pour définir la taxonomie. La section 3 énumère les déviations et vulnérabilités pour chaque rôle de AGR4BS, définissant ainsi la taxonomie. La section 4 propose une discussion sur la manière dont cette taxonomie et AGR4BS peuvent être utilisés pour sécuriser les systèmes blockchain. Enfin, la section 6 conclut cet article et met en avant des perspectives futures.

2 Concepts essentiels

2.1 Aperçu des systèmes blockchain

Un système blockchain permet à ses participants de construire collectivement un système distribué de nature économique, sociale et technologique où ils peuvent effectuer des transactions vérifiables et sûres sans nécessiter de tiers de confiance [9]. Certains participants utilisent la blockchain comme un service transactionnel, tandis que d'autres sont motivés financièrement pour fournir ce service en participant activement au mécanisme de consensus. Les transactions sont typiquement incluses dans des blocs qui sont liés entre eux par une fonction de hachage.

Nous pouvons identifier deux catégories principales de blockchain : privée et publique. Dans une blockchain privée, la participation et la contribution sont conditionnées par un système de permissions. Les contributeurs sont le plus souvent motivés par un mécanisme d'incitations défini par la structure (*i.e.*, entreprise ou consortium) contrôlant la blockchain, plutôt que par le système blockchain lui-même. Les blockchains publiques n'ont pas de système de permission : la participation et la contribution sont accessibles à tous et motivés par un mécanisme propre au système, comme des récompenses pour la création de blocs. Dans de tels systèmes, les participants

ont un intérêt dans la stabilité du système à long-terme. Contribuer à un système blockchain se fait par le biais de la participation au consensus, au cours duquel les contributeurs s'accordent sur la transition d'état à effectuer. Typiquement, dans un consensus Proof-of-Work (PoW), les contributeurs (*i.e.*, mineurs) sont en compétition directe par le biais de leur puissance de calcul. Dans un consensus Proof-of-Stake (PoS) les contributeurs sont choisis de manière déterministe proportionnellement à leurs *stakes* (*i.e.*, investissement) pour proposer un nouveau bloc.

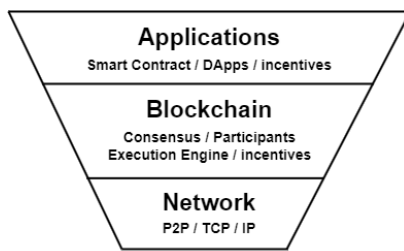


FIGURE 1 – Couches technologiques d'un système blockchain

Les systèmes blockchain peuvent être représentés comme une hiérarchie de couches technologiques, comme présenté dans la figure 1. La couche *Network*, la plus basse, contient les primitives de communication ainsi que les protocoles bas niveau requis pour bâtir un réseau blockchain. La couche *Blockchain* contient les participants, leurs motivations ainsi que l'environnement d'exécution et fournit les protocoles haut niveau ainsi que les structures de données nécessaires au consensus et au maintien du registre. La couche *Applications* contient quant à elle les *contrats intelligents*.

Contrairement aux travaux séparant le consensus des motivations, nous les considérons tous deux comme partie intégrante de la couche *Blockchain*, car une participation au consensus mène à des récompenses ou des punitions en cas de mauvais comportement. De plus, une décomposition structurelle aboutit souvent à des vulnérabilités traversant plusieurs couches, (*ex.*, le selfish mining est souvent défini comme lié au consensus, aux motivations ainsi qu'au réseau). Une représentation plus concise est possible avec une approche basée rôles, dans la mesure où cela permet d'identifier directement les acteurs.

2.2 Le modèle AGR4BS

Dans le modèle AGR, les Systèmes Multi-Agents (SMA) sont modélisés selon une perspective organisationnelle par le biais de trois

concepts principaux : Agent, Groupe et Rôle [8]. Les agents sont des entités communicantes jouant un ou plusieurs rôles au sein de groupes. Les groupes servent à identifier des schémas d'activité (*i.e.* rôles) partagés par plusieurs agents. Les rôles sont des représentations abstraites permettant de définir le comportement d'un agent au sein d'un groupe.

Dans le contexte de la blockchain, le modèle AGR4BS permet d'identifier les agents, groupes et rôles génériques, ainsi que de spécifier les attributs et comportements nécessaires pour chaque fonctionnalité de rôle (figure 2). Ainsi, une combinaison spécifique de ces rôles permet de définir une entité logique dans une implémentation blockchain spécifique (*ex.*, un mineur Bitcoin est composé des rôles suivants : Blockchain Maintenir, Block Proposer, Block Endorser et Investor). AGR4BS propose une approche multi-agent cohérente pour modéliser les systèmes blockchain, et pose des bases solides pour une analyse des vulnérabilités d'incitations grâce à une représentation concrète des rôles et comportements des participants impliqués.

2.3 Vulnérabilité d'incitations, déviation comportementale et contre mesure

Une vulnérabilité peut être définie formellement comme une faille au sein d'un système qui peut être exploitée par un agent afin d'impacter négativement un système. Dans cette étude, nous nous concentrons sur un type spécifique de vulnérabilités des systèmes blockchain : les *vulnérabilités d'incitations*, que nous définissons comme un problème d'alignement entre (1) le comportement attendu d'un agent selon le protocole et (2) le comportement obtenu en suivant une interprétation rationnelle des motivations présentes dans le système. Ce manque d'alignement incite les participants à dévier de leur comportement nominal. En ce sens, un comportement est dit déviant lorsqu'il n'adhère pas strictement à l'implémentation officielle (*i.e.*, au comportement nominal). Si une telle déviation impacte le système ou ses participants, des contre-mesures doivent être mises en place afin de réduire sa faisabilité et / ou son impact. Nous considérons une vulnérabilité d'incitations comme la cause principale d'une déviation.

2.4 Caractéristiques de la taxonomie

Afin de classifier, catégoriser et quantifier des vulnérabilités, nous utilisons les concepts sui-

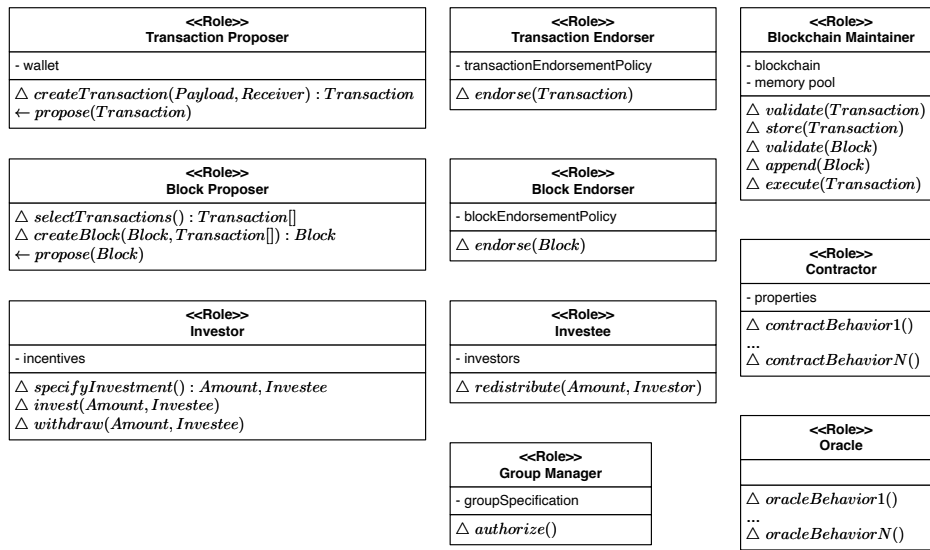


FIGURE 2 – Les rôles ainsi que leurs attributs et comportements dans un système blockchain [14].

vants : Impact, Sévérité, Risque, Échelle, Priorité et Système.

Impact se réfère au type d’impact attendu pour une vulnérabilité spécifique. Nous considérons trois catégories principales : Équité, Économie, Sécurité. Un impact sur l’équité se produit dès qu’une discrimination entre les agents a lieu pour des raisons qui ne font pas partie du protocole. De plus, tout déséquilibre entre la proportion des ressources investies et celle des récompenses reçues est aussi considéré dans cette catégorie. Un impact sur l’économie se produit dès que cette dernière est perturbée, par exemple une augmentation artificielle des frais de transactions. Un impact sur la sécurité a lieu dès lors qu’au moins une des propriétés fondamentales du système est partiellement ou complètement compromise, tel que la finalité ou l’intégrité de la chaîne de blocs.

Sévérité mesure la gravité d’une attaque réussie et peut avoir pour valeur : Très haut, Haut, Moyen, Bas, Très Bas. Ces dernières sont des alias pour $1, \frac{4}{5}, \frac{3}{5}, \frac{2}{5}$ et $\frac{1}{5}$ respectivement. Ces niveaux ne sont pas basés sur une notion quantifiable de sévérité, mais sont utilisés pour catégoriser les vulnérabilités de manière informelle et calculer leur niveau de priorité relatif.

Très Bas implique qu’un agent ou groupe restreint d’agents est faiblement impacté, mais toujours fonctionnel, sans impact observable sur les groupes de plus haut niveau, ni sur le système. Bas implique aussi qu’un agent ou groupe restreint d’agents est impacté et sont potentiellement non fonctionnels, tandis que les groupes

de haut niveau et le système sont toujours fonctionnels. Moyen désigne un impact touchant les agents et les groupes, sans compromettre le système, mais ayant des conséquences globales et visibles sur au moins une des propriétés suivantes : Équité, Économie, Sécurité. Une Sévérité de niveau Haut implique des conséquences non négligeables sur le système. Enfin, Très Haut implique un impact conséquent ayant de sérieuses répercussions, tel qu’un manque total d’équité ou une possibilité d’arrêt du système.

Risque se rapporte à la faisabilité d’une attaque en termes de ressources requises pour la mener à bien. Les niveaux de risque sont similaires à ceux définis pour la sévérité : Très Haut, Haut, Moyen, Bas, Très Bas. Ces niveaux sont eux aussi des alias sur les mêmes valeurs numériques. Très Haut signifie qu’une vulnérabilité est facilement exploitable, car requérant peu ou pas de ressources. Haut se réfère à une vulnérabilité dont l’exploitation requiert des ressources, mais qui restent accessibles pour la plupart des participants. Moyen est utilisé pour une vulnérabilité qui demande une quantité non négligeable de ressources. Bas et Très Bas sont utilisés pour décrire des vulnérabilités dont l’exploitation requiert une grande quantité de ressources.

En ce qui concerne la définition du risque, et plus particulièrement la faisabilité, un point important est que les ressources requises pour exploiter une vulnérabilité dépendent du type de cette dernière. Par exemple, une attaque se faisant par le biais du minage de blocs requiert de la puissance de calcul, tandis qu’une attaque

réseau demande plus généralement de la bande passante. Notre définition de `ressource` est donc intentionnellement vague pour accommoder les différents types d'attaques.

Échelle. Sachant que les systèmes blockchain sont décentralisés, nous devons différencier le risque et la sévérité d'une exploitation en fonction de l'échelle de l'attaque. Nous considérons deux niveaux : *Petite échelle* et *large échelle*. En fonction du type de vulnérabilité, l'échelle peut se rapporter au nombre d'agents exploitants (*i.e.*, attaque sybil), à la puissance de calcul totale mise en œuvre (*i.e.*, attaque de minage) ou encore à la valeur financière requise.

Priorité classe les vulnérabilités en fonction de leur sévérité et du risque d'exploitation réussie : elle est définie comme le produit de ces deux variables. Nous pouvons calculer le score de priorité pour les deux échelles considérées, et adoptons une approche pessimiste en ne retenant que le score le plus élevé.

Système décrit les types de systèmes blockchain sujets à une vulnérabilité particulière, sachant que la plupart des systèmes considérés sont profondément liés au mécanisme de consensus utilisé : PoW (Proof of Work), PoS (Proof of Stake), PoA (Proof of Authority), PBFT (Practical Byzantine Fault Tolerant), Approbation explicite de blocs ou de transactions, Tous.

Dans la section suivante, nous présentons chaque rôle ainsi que leurs déviations respectives avec, pour chacune d'entre elle, leur `Impact`, `Sévérité`, `Risque` ainsi que leur `Priorité` (résumé dans la table 1).

3 Taxonomie basée rôle

La table 1 montre une classification des vulnérabilités d'incitations liées aux rôles définis dans AGR4BS.

3.1 Block Proposer

Comportement Nominal. Block Proposer sélectionne le sous ensemble des transactions les plus intéressantes financièrement, et essaie de créer un nouveau bloc valide, en étendant la chaîne principale tel que défini par le protocole. S'il réussit, il propose immédiatement le nouveau bloc à son voisinage dans le réseau.

Censure de transactions. À travers une déviation du comportement `selectTransactions`, un Block

Proposer peut censurer certaines transactions et ainsi impacter l'équité du système. Cela peut arriver lorsqu'un Block Proposer exclut volontairement des transactions spécifiques de son mécanisme de sélection, quand bien même elles seraient financièrement attractives. Cette censure ciblant l'identité des participants a pour but de délayer ou bien empêcher la transaction. Pour que cette déviation ait un impact, une majorité des Block Proposer doit être disposée à appliquer la même politique de censure en raison du caractère décentralisé et redondant des systèmes blockchain. Bien que potentiellement délayés, les agents ciblés par cette censure peuvent toujours participer, soit par le biais des Block Proposer nominaux restants, ou en devenant eux même Block Proposer. À petite échelle, cette déviation n'a quasiment aucun impact.

Propagation sélective de blocs. La proposition et propagation de bloc sur le réseau peut être intentionnellement biaisée par un Block Proposer suite à une déviation du comportement `proposeBlock`. Par exemple, un Block Proposer peut intentionnellement exclure un compétiteur de sa proposition / propagation de blocs et ainsi légèrement délayer ses connaissances sur la transition d'état. À grande échelle, cette déviation peut causer un délai significatif à l'agent ciblé ou lui empêcher tout accès à l'information.

Délai du consensus. Délayer le consensus ou l'arrêter est principalement lié aux systèmes ayant un consensus inspiré de PBFT où les Block Proposer peuvent soit proposer des blocs invalides, soit ne pas en proposer du a une double déviation des comportements `createBlock` et `proposeBlock`. Une attaque réussie sur le consensus aura un impact direct sur tous les participants. Dans ces systèmes, les contributeurs, souvent appelés *Validateurs*, peuvent comploter pour contrôler plus de 33% de nœuds au sein d'un comité.

Selfish / Stubborn Block Creation Un Block Proposer peut ne pas miner un nouveau bloc pour étendre ce qui est considéré comme la chaîne canonique, mais pour étendre une chaîne adverse privée. Un tel comportement est possible grâce à une déviation des comportements `createBlock` et `proposeBlock`. Les principales victimes sont les autres contributeurs. Cette déviation n'est pertinente que pour les systèmes Proof-of-Work (PoW) et a été étudiée dans [7].

Valeur Maximale Extractible (VME). Une autre vulnérabilité ciblant l'économie des blockchains publiques est la possibilité de réordonner les

Rôle	Déviations Exploitant une vulnérabilité d'incitation				Métriques						
	Déviation	Comportements Déviés	Rôles Impactés	Références	Impact	Petite Échelle		Grande Échelle		Priorité	Système
						Sévérité	Risque	Sévérité	Risque		
Block Proposer	Censure de Transactions	selectTransactions	Transaction Proposer	[9]	Équité	●○○○○	●●○○○	●●○○○	●●○○○	0.16	Tous
	Propagation Sélective de Blocs	proposeBlock	Blockchain Maintenir Block Proposer	N/A		●○○○○	●○○○○	●●○○○	●●○○○	0.24	Tous
	Délai du Consensus	createBlock proposeBlock	Tous	N/A		●○○○○	●○○○○	●●●●●	●○○○○	0.80	PBFT
	Selfish / Stubborn Block Creation	createBlock proposeBlock	Blockchain Maintenir Block Proposer	[7]	Équité Sécurité	●○○○○	●○○○○	●●●●●	●○○○○	0.25	PoW
	Valeur Maximale Extractible	selectTransaction createBlock	Transaction Proposer	[4]	Équité Économie	●○○○○	●●○○○	●○○○○	●●○○○	0.32	Tous
Block Endorser	Censure de Blocs	endorseBlock	Block Proposer Transaction Proposer	N/A	Équité	●○○○○	●●○○○	●●○○○	●○○○○	0.16	Approbation Explicite
Transaction Endorser	Censure de Transactions	endorseTransaction	Transaction Proposer	[13]		●○○○○	●●○○○	●●○○○	●○○○○	0.16	Approbation Explicite
Transaction Proposer	Double Dépense	createTransaction	All	[3]	Équité	●○○○○	●○○○○	●●●●●	●○○○○	0.25	Tous
	Délit d'Initié	createTransaction	Transaction Proposer	[6]	Économie	●○○○○	●●●●●	●●○○○	●●○○○	0.60	Tous
Blockchain Maintenir	Défaut de Validation de Transaction	validateTransaction	None	[12]	Sécurité	●○○○○	●●○○○	●●●●●	●●○○○	0.40	Tous
	Défaut de Validation de Bloc	validateBlock	None			●○○○○	●●○○○	●●●●●	●●○○○	0.40	Tous
	Défaut d'Exécution	validateTransaction executeTransaction	None			●○○○○	●○○○○	●●●●●	●●○○○	0.40	Tous
	Défaut de Diffusion	diffuseTransaction	Blockchain Maintenir			[5]	Équité	●○○○○	●●○○○	●●○○○	●●○○○
Oracle	Oracle Corrompu	oracleBehavior	Contractor Investor Investee	[2]	Économie	●○○○○	●●○○○	●●●●●	●○○○○	0.40	Tous
Investee	Redistribution Partielle	redistribute	Investor	N/A	Équité Économie	●○○○○	●●○○○	●●○○○	●●○○○	0.32	Tous

TABLE 1 – Taxonomie des vulnérabilités d'incitation basée rôles.

Très Bas : ●○○○○ , Bas : ●●○○○ , Moyen : ●●●○○ , Haut : ●●●○○ , Très Haut : ●●●●●

transactions au moment de la sélection, afin de maximiser les commissions et les coûts d'exécutions¹ [4]. Cette optimisation agressive est le résultat d'une déviation des comportements *selectTransaction* et *createBlock* et impacte directement les Transaction Proposer qui en sont victimes. Ce genre de comportement, bien que normal pour des participants rationnels, n'est pas désiré, car cela peut conduire à une augmentation artificielle des commissions de transactions et ainsi réduire l'accessibilité du système. De plus, les blocs ainsi créés peuvent avoir une récompense totale si importante que d'autres Block Proposers peuvent être incités à créer une branche adverse pour récupérer la récompense de bloc pour eux même. L'ordre des transactions ne respectant pas nécessairement leur ordre de création / propagation ni un ordre reposant sur les commissions offertes par les utilisateurs ce qui impacte directement l'équité du système.

3.2 Block Endorser

Comportement Nominal. Block Endorser approuve l'inclusion de blocs dans la chaîne en suivant une politique d'approbation bien définie.

Censure de blocs. Block Endorser peut volontairement refuser d'approuver des blocs ayant cer-

1. Quantifying Blockchain Extractable Value : How dark is the forest? - <https://arxiv.org/abs/2101.05511> dernier accès le : 28/10/2022

taines caractéristiques par une déviation du comportement nominal de *endorseBlock*. Ceci impacte directement le Block Proposer ayant créé et proposé le bloc et, indirectement, les Transactions Proposer dont les transactions sont incluses dans ce bloc. En fonction de la politique d'approbation requise par le protocole, cette déviation peut empêcher un bloc d'être inclus dans la chaîne pour une raison indépendante du consensus et non voulue par le système, dont l'équité est donc impactée. Dans le pire des cas, ce type d'exploitation peut empêcher purement et simplement la production de nouveau blocs. Cette censure est relativement simple à mettre en place pour un agent, mais n'aurait généralement que peu d'impact, car les systèmes reposant sur les Block Endorser requièrent toujours plus d'une seule approbation pour éviter de s'exposer à cette vulnérabilité. Une telle attaque a grande échelle est peu probable, car elle demanderait de contrôler la majorité des Block Endorser.

3.3 Transaction Endorser

Comportement nominal. Transaction Endorser approuve l'inclusion de transactions en suivant une politique d'approbation bien définie.

Censure Transactions (censure de transactions) De manière similaire au rôle *Block Endorser*, *Transaction Endorser* est sujet à une déviation du comportement *endorseTransaction* permettant une censure d'un ou plusieurs *Transaction*

Proposer. Transaction Endorser peut choisir de refuser d'approuver des transactions sans aucun motif légitime au sens du protocole. Dans un système où les transactions doivent être approuvées, tel que Hyperledger Fabric ² un tel comportement peut impacter des participants essayant d'interagir avec le reste du système. À petite échelle, cette déviation est peu impactante, car la majorité des *Transaction Endorser* se comporteraient normalement. À grande échelle, l'impact est plus important, car il est possible d'exclure des participants du système, mais cela requiert une majorité de *Transaction Endorser* déviants.

3.4 Transaction Proposer

Comportement nominal. *Transaction Proposer* crée une transaction valide prévoyant une commission suffisante pour son inclusion dans un bloc et la propose au reste du réseau.

Double dépense. Dans le contexte d'une blockchain autorisant les embranchements, un *Transaction Proposer* peut dévier de son comportement *createTransaction* nominal pour tenter une double dépense [3]. Une telle attaque est généralement accompagnée d'une autre visant à générer artificiellement des embranchements tel que du selfish mining. *Transaction Proposer* propose alors deux transactions conflictuelles, ou bien deux fois la même transaction sur au moins deux chaînes candidates. L'impact de ce type d'attaque serait principalement réputationnel et financier, car le système ne serait donc plus en mesure d'assurer un niveau de sécurité élémentaire et les participants auraient intérêt à quitter le système.

Délit d'initié Les transactions étant publiques et diffusées dans le réseau avant leur inclusion dans un bloc, tous les participants ont connaissance des futures transitions d'états avant qu'elles aient lieu. Cette caractéristique des réseaux blockchain facilite le délit d'initié. Par exemple, cela permet à un *initié* de tirer avantage d'une transaction en attente et devant effectuer un ordre d'achat / vente massif sur un échange décentralisé à travers une déviation du comportement nominal de *createTransaction*. Un *initié* peut obtenir la priorité par rapport à la transaction cible par le biais du système de commissions des transactions, car cette commission est le principal critère d'inclusion d'une transaction dans un bloc par les Block Creator. Au sens strict, le *délit d'initié* n'est pas une déviation, car

² Hyperledger Fabric, <https://www.hyperledger.org/use/fabric>, accédé le 09/12/2022.

cela n'est pas contraire au protocole. Cependant, les effets négatifs liés et le fait que tout participant rationnel doit adopter ce comportement nous permettent de le considérer comme une vulnérabilité d'incitation. Le *délit d'initié* est étroitement lié au VME. En effet, un initié rationnel accepte d'abandonner 99.99% de son profit sous forme de commissions de transactions au Block Creator. L'impact du *délit d'initié* varie en fonction de l'échelle, et plus précisément du nombre de participants qui cherchent activement des opportunités. Peu d'*initiés* n'auront qu'un impact minime. En revanche, si ce comportement est adopté par une majorité des participants, les conséquences sur l'économie et l'équité sont importantes, car cela contribue à l'augmentation des commissions de transactions du fait de la compétition entre les *initiés*, et incite les utilisateurs nominaux à quitter le système si rien n'est fait pour le contrôler.

3.5 Blockchain Maintenir

Comportement nominal. Blockchain Maintenir valide tous les blocs et transactions reçus. Les transactions valides sont stockées en vue d'une inclusion dans un futur bloc. Les Blocs valides sont ajoutés à la blockchain et ses transactions sont exécutées pour effectuer la transition d'état.

Défaut de validation de transaction. La validation des transactions n'étant pas explicitement récompensée, un agent rationnel peut être incité à ne pas la faire à travers une déviation du comportement *validateTransaction*. Cette déviation sacrifie une partie de la sécurité du système pour un avantage de temps et de ressources. À petite échelle, cela n'a aucun impact sur le système, car les autres contributeurs assurent la sécurité. Cependant, l'agent déviant est légèrement avantageux. À grande échelle, si une majorité de Blockchain Maintenir refuse de valider les transactions, cette déviation ne permet plus de garantir la cohérence du registre. Cette vulnérabilité est connue sous le nom de *Verifier's Dilemma* [12].

Défaut d'Exécution. Dans le cas nominal, lorsqu'un agent reçoit une transaction liée à un contrat intelligent, il doit l'exécuter. L'agent peut ne pas savoir si le contrat contient une logique erronée, déclenchant une attaque ou bien simplement des actions invalides. Le coût d'une exécution en temps et ressources de calcul peut être conséquent pour un agent, puisque que les exécutions de contrats sont liées aux transactions, la principale vulnérabilité est similaire à celle du comportement *validate(Transaction)* où les

comportements *validateTransaction* et *executeTransaction* seraient déviants.

Défaut de Validation de Bloc. Valider un bloc peut s'avérer coûteux pour un Blockchain Maintainer. Lorsqu'un agent est à la fois Blockchain Maintainer et Block Proposer, il peut choisir de ne pas valider les blocs et potentiellement inclure et propager des blocs invalides par une déviation du comportement *validateBlock*. La validation d'un bloc requiert la validation de sa structure, des méta-données et transactions qu'il contient, ce qui nécessite leur exécution.

Défaut de Diffusion. La diffusion des transactions n'étant pas récompensée, les participants peuvent opter pour un comportement égoïste qui, par le biais d'une déviation du comportement nominal de *diffuseTransaction*, leur permet de délayer ou ne pas diffuser les transactions reçues. Cette déviation est particulièrement pertinente dans les systèmes compétitifs ouverts comme PoW. Les systèmes blockchain actuels n'ont pas de mécanisme d'incitation pour la diffusion de transactions et reposent principalement sur les intérêts à long terme des contributeurs (*i.e.*, Block Proposers et Blockchain Maintainers). Si aucune transaction n'était diffusée, l'utilisabilité du système serait compromise. Ainsi, une telle attaque à grande échelle est improbable, car elle va à l'encontre des intérêts des contributeurs.

3.6 Oracle

Comportement Nominal. Le rôle *Oracle* contient les comportements nécessaires à l'apport au sein du système d'informations extérieures valides et non altérées.

Oracle corrompu. Un Oracle peut être corrompu et volontairement transmettre des informations erronées par une déviation d'un de ses comportements spécifiques, ou simplement du fait d'une logique erronée. Ceci mènerait le système, et particulièrement les *smart contracts*, à prendre des décisions sur la base d'information incorrectes. De plus, la source d'information peut être corrompue alors que l'oracle ne l'est pas, ce dernier étant dans l'incapacité de le détecter. Ces deux cas sont indiscernables du point de vue de la blockchain et peuvent tous deux avoir de sérieuses conséquences. Le fait de faire confiance à des données extérieures est connu comme le *Oracle Problem* [2], un paradoxe existe entre la nécessité d'avoir des oracles pour rendre la blockchain utile à des applications en lien avec le monde réel et la nature même des systèmes blockchain qui ne reposent pas sur la confiance.

3.7 Investee

Comportement Nominal. Investee reçoit des investissements de la part d'autres participants investisseurs, fournit un service, et leur redistribue des récompenses proportionnellement à leurs contributions respectives.

Pas de redistribution ou redistribution partielle. Si un Investee ne redistribue pas correctement les richesses obtenues grâce à ses investisseurs du fait d'une déviation du comportement *redistribute*, il peut obtenir un avantage financier. Cependant, cela aurait un coût réputationnel, économique et d'équité sur le système. Un tel comportement peut facilement être contrôlé, car ces informations sont par définition publiques : un système de réputation et de liste noire peut ainsi être créé pour dénoncer et punir ce type de déviation, comme dans la blockchain Tezos³.

4 Discussion

La taxonomie présentée dans cet article, ainsi que le modèle sur lequel elle se base, AGR4BS, peuvent être utilisés pour évaluer les vulnérabilités d'incitations dans les systèmes blockchain. Premièrement, il convient de créer ou étendre un modèle spécifique de la blockchain en question en utilisant les abstractions AGR4BS.

Vu que plusieurs systèmes blockchain partagent des fonctionnalités et une logique commune, un rôle peut être sujet à la même déviation à travers différents systèmes. Une évaluation de la sécurité des incitations d'une blockchain commencerait par une exploration de la faisabilité et de l'impact de vulnérabilités connues sur les systèmes similaires. Ensuite, le modèle et la taxonomie existante peuvent être utilisés pour guider la recherche de vulnérabilités en ciblant les rôles critiques en priorité, typiquement, Block Proposer et Blockchain Maintainer.

Cette taxonomie a aussi des limites et des contraintes, par exemple, une vulnérabilité ne peut être classifiée que lorsque les rôles et comportements déviants qu'elle implique ont été correctement modélisés par le biais de AGR4BS. Ce travail supplémentaire pourrait rendre l'utilisation de la taxonomie prohibitive si de trop nombreuses vulnérabilités sont à ajouter. Les autres taxonomies existantes, bien que moins représentatives de la dynamique multi-agent des systèmes blockchains ne connaissent pas ce problème, cela est particulièrement vrai pour les

3. Tezos, <https://tezos.com/>, dernier accès le 12/10/2022

Caractéristiques	Références					
	Saad et al. [15]	Hameed et al. [10]	Sayed et al. [16]	Alkhalifah et al. [1]	Li et al. [11]	Ours
Couches	- Application - Blockchain - Network	- Application - Blockchain - Network	- Application	- Application - Blockchain - Network	- Application - Blockchain - Network	- Blockchain
Proposition de contre mesures	Oui	Oui	Oui	Oui	Oui	Oui
Classification	Couches technologiques	Couches technologiques	Types d'attaques	Couches technologiques	Risque & Vulnérabilité	Basée rôles
Incentives Focused	Non	Non	Non	Non	Non	Oui

TABLE 2 – Comparaison des études proposant une taxonomie de la sécurité des systèmes blockchain

taxonomies qui classifient les vulnérabilités en fonction des couches technologiques.

5 Littérature

Il existe plusieurs études et taxonomies exhaustives dans la littérature [1, 10, 11, 15, 16] (voir table 2 pour une comparaison).

Saad et al. [15] définissent une taxonomie des attaques blockchain à travers trois catégories principales : Structure, Peer-to-Peer et Applications. Ils listent différentes attaques connues et exposent des contre-mesures existantes ou théoriques. Hameed et al. [10] définissent plusieurs taxonomies avec un fort intérêt pour les applications industrielles. Ces taxonomies portent sur le design, la sécurité et la confidentialité. Ils présentent plusieurs attaques ayant ou pouvant impacter la blockchain sur chacune de ses couches technologiques, et mettent en avant différentes contre-mesures atténuant leurs impacts. Sayeed et al. [16] proposent une étude centrée sur les *smart contracts*, donc restreinte à la couche applicative de la blockchain. Ils ne considèrent que la blockchain Ethereum. Cette étude définit implicitement une taxonomie par une catégorisation des principaux types d'attaques et met en avant des outils augmentant la sécurité des contrats dans Ethereum. Alkhalifah et al. [1] définissent une taxonomie des menaces et vulnérabilités propres à la blockchain répartie sur les catégories suivantes : vulnérabilités client, vulnérabilités du mécanisme de consensus, vulnérabilités des *mining pools*, vulnérabilités réseau, et enfin vulnérabilités des contrats intelligents, restreinte là aussi à Ethereum. Li et al. [11] étudient la sécurité des systèmes blockchain et proposent une taxonomie succincte des risques liés à la cryptographie, au consensus et aux transactions. Ils définissent, eux aussi, une taxonomie des vulnérabilités pour les *smart contracts* Ethereum.

Les études exhaustives se concentrent principalement sur le type d'impact des attaques sur le système et sur la couche technologique dans laquelle elles ont lieu. Ainsi, les contre-mesures sont souvent réduites à traiter les conséquences du problème (*i.e.*, système de détection, amélioration

de la résilience). Les études se concentrant précisément sur une attaque particulière partent souvent des raisons à l'origine de la déviation et, dans ce cadre, proposent des contre-mesures modifiant le mécanisme d'incitation afin de rendre l'attaque irrationnelle. Le but de cette taxonomie est de combiner les deux approches par le biais d'une classification exhaustive, mais toujours centrée sur les causes d'une déviation plutôt que sur ses conséquences.

6 Conclusion et travaux futurs

La taxonomie des vulnérabilités d'incitations des systèmes blockchain définie dans cet article a pour but d'aider les chercheurs et développeurs à mieux comprendre les menaces existantes, et ainsi la construction et la maintenance de systèmes plus résilients. Cette taxonomie repose sur le modèle organisationnel générique multi-agent de blockchains AGR4BS et permet de calculer le score de priorité de chaque vulnérabilité d'incitation. Elle permet de définir les vulnérabilités en tant que déviations des rôles et comportements nominaux. Cette taxonomie liste et classe différentes vulnérabilités connues, mais permet aussi de quantifier et classifier celles qui pourraient être découvertes plus tard. Nous recommandons aux chercheurs de se concentrer sur les vulnérabilités ayant les scores de priorité les plus élevés (Table 1), tel que le *Délai du consensus* lié au rôle *Block Proposer*.

Cette taxonomie s'inscrit dans un processus de recherche en 3 phases (modélisation, classification, recherche) visant à automatiser la recherche de vulnérabilités d'incitations dans les systèmes blockchain au sein d'un cadre de travail clairement défini. AGR4BS fournit ainsi les éléments nécessaires à la modélisation des systèmes ainsi que des vulnérabilités qui peuvent les impacter. Cette taxonomie permet quant à elle de classer et prioriser ces vulnérabilités de manière exhaustive tout en se concentrant sur les causes des différentes déviations afin de proposer des solutions pérennes. Enfin, la recherche automatique de vulnérabilités d'incitations peut être effectuée par le *Multi-Agent Reinforcement Learning* (MARL). Cette approche permet l'étude de

participants rationnels et non rationnels dans un même système. Le MARL peut-être utilisé pour sécuriser le processus de mise à jour d'un système blockchain, par exemple si le mécanisme d'incitations est modifié.

Références

- [1] Ayman Alkhalifah, Alex Ng, A. S. M. Kayes, Javed Chowdhury, Mamoun Alazab, and Paul Watters. *A Taxonomy of Blockchain Threats and Vulnerabilities*, pages 3–25. CRC Press, United States, 1 edition, August 2020.
- [2] Giulio Caldarelli. Understanding the blockchain oracle problem : A call for action. *Information (Switzerland)*, 11(11) :1–19, 2020.
- [3] Usman W. Chohan. The Double Spending Problem and Cryptocurrencies. *SSRN Electronic Journal*, n/a(n/a) :11p, 2018.
- [4] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0 : Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 910–927. IEEE, 2020.
- [5] Oğuzhan Ersoy, Zhijie Ren, Zekeriya Erkin, and Reginald L. Lagendijk. Transaction propagation on permissionless blockchains : Incentive and routing mechanisms. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 20–30. IEEE, 2018.
- [6] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. Sok : Transparent dishonesty : Front-running attacks on blockchain. In Andrea Bracciali, Jeremy Clark, Federico Pintore, Peter B. Rønne, and Massimiliano Sala, editors, *Financial Cryptography and Data Security*, pages 170–189, Cham, 2020. Springer Inter. Publishing.
- [7] Ittay Eyal and Emin Gün Sirer. Majority is not enough : Bitcoin mining is vulnerable. *Commun. ACM*, 61(7) :95–102, jun 2018.
- [8] Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From agents to organizations : An organizational view of multi-agent systems. In Paolo Giorgini, Jörg P. Müller, and James Odell, editors, *Agent-Oriented Software Engineering IV*, pages 214–230, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [9] Önder Gürcan, Antonella Del Pozzo, and Sara Tucci-Piergiovanni. On the bitcoin limitations to deliver fairness to users. In H. Panetto, C. Debruyne, W. Gaaloul, M. Papazoglou, A. Paschke, C-A. Ardagna, and R. Meersman, editors, *On the Move to Meaningful Internet Systems. OTM 2017 Conferences*, pages 589–606, Cham, 2017. Springer International Publishing.
- [10] Khizar Hameed, Mutaz Barika, Saurabh Garg, Muhammad Bilal Amin, and Byeong Kang. A taxonomy study on securing blockchain-based industrial applications : An overview, application perspectives, requirements, attacks, countermeasures, and open issues. *J Ind Inf Integr*, 26 :100312, 2022.
- [11] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107 :841–853, 2020.
- [12] Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. Demystifying incentives in the consensus computer. *Proc. of the ACM Conference on Computer and Communications Security*, 2015-Octob :706–719, 2015.
- [13] Pierre-Yves Piriou, Olivier Boudeville, Gilles Deleuze, Sara Tucci-Piergiovanni, and Önder Gürcan. Justifying the dependability and security of business-critical blockchain-based applications. In *2021 Third Inter. Conf. on Blockchain Computing and Applications (BCCA)*, pages 97–104. IEEE, 2021.
- [14] Hector Roussille, Önder Gürcan, and Fabien Michel. Agr4bs : A generic multi-agent organizational model for blockchain systems. *Big Data and Cognitive Computing*, 6(1) :41p, 2022.
- [15] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles Kamhoua, Sachin Shetty, Dae Hun Nyang, and David Mohaisen. Exploring the Attack Surface of Blockchain : A Comprehensive Survey. *IEEE Communications Surveys and Tutorials*, 22(3) :1977–2008, 2020.
- [16] Sarwar Sayeed, Hector Marco-Gisbert, and Tom Caira. Smart Contract : Attacks and Protections. *IEEE Access*, 8 :24416–24427, 2020.

Community-OrBAC : un modèle de contrôle d'accès établi à partir des agents pour les systèmes de collaboration centrés sur la communauté

Rodrigue N'goran^{a,b}

kouadio-rodrique.ngoran@imt-atlantique.fr

Yvon Kermarrec^a

yvon.kermarrec@imt-atlantique.fr

Olivier Asseu^b

olivier.asseu@esatic.edu.ci

Jean-Louis Tetchueng^c

jean-louis.tetchuengfoping@univ-rennes1.fr

^aLab-STICC, IMT-Atlantique, 29280 Brest, France

^bLASTIC, INP-HB, Yamoussoukro, Côte d'Ivoire

^cUniversité Rennes 1, Rennes, France

Résumé

Le besoin accru et constant de partage de ressources au sein des entreprises et entre organisations favorise la création de systèmes de collaboration pour des communautés spécifiques. Ces infrastructures distribuées et complexes se caractérisent par la nécessité de garantir l'autonomie des entités membres et la sécurité des informations sensibles de diverses natures échangées. Par conséquent, La problématique de la mise en place de mécanismes de sécurité et de contrôle des collaborations se pose. Dans cet article, nous proposons le Community-OrBAC, un modèle de contrôle d'accès établi à partir des agents pour des systèmes centrés sur la communauté. Le modèle vise à fournir des techniques de négociation de contrat de collaboration et de spécification dynamique de règles de politique de sécurité. Par ailleurs, notre approche étend la notion de contexte du modèle de contrôle d'accès OrBAC et présente une étude de cas dans un cloud communautaire.

Mots-clés : Multi-agents, Contrôle d'accès, Confiance, Cloud Communautaire, Négociation, OrBAC

Abstract

The increased and constant need to share resources within companies and between organizations favors the creation of collaboration systems for specific communities. These distributed and complex infrastructures are characterized by the need to guarantee the autonomy of member entities and the security of sensitive information of various kinds exchanged. Therefore, the problem of setting up security and control mechanisms for collaborations arises. In this paper, we propose Community-OrBAC, an agent-based

access control model for community-centric systems. The model aims at providing techniques for collaborative contract negotiation and dynamic specification of security policy rules. Furthermore, our approach extends the notion of context of the OrBAC access control model and presents a case study in a community cloud.

Keywords: Multi-agents, Access control, Trust, Community Cloud, Negotiation, OrBAC

1 Introduction

L'émergence des technologies de l'information et de la communication favorise de plus en plus le partage de ressources entre utilisateurs et entre organisations. Des systèmes de collaboration destinés à des communautés d'utilisateurs spécifiques peuvent ainsi être créés dans le but de produire des données, partager des ressources pour satisfaire leurs besoins respectifs. La diversité des organisations et des ressources, la sensibilité des données échangées font émerger des challenges en matière de sécurité, de confidentialité des données et d'autonomie des organisations. Il est donc primordial de mettre en place des mécanismes de protection des ressources partagées et de garantir la confiance au sein de ces communautés. Les modèles de contrôle d'accès se présentent comme l'un des moyens les plus utilisés dans la sécurisation des infrastructures informatiques. Plusieurs études ont été effectuées pour proposer des modèles de contrôle d'accès qualifiés de modèles classiques[3]. Par ailleurs, la nature dynamique et complexe des environnements de collaboration a favorisé des propositions de techniques de contrôle d'accès adaptés à ces systèmes[27]. Cependant, les systèmes collaboratifs construits autour de commu-

nauté d'organisations présentent certaines singularités. Ces communautés donnent une priorité aux besoins, aux intérêts et à l'autonomie de leurs membres, favorisent des relations sociales entre eux [24]. Par conséquent, le respect de ces exigences spécifiques dans la définition des politiques de sécurité pour ce type d'environnement est crucial.

Nous proposons dans cet article, le *Community-OrBAC*, un modèle de contrôle d'accès reposant sur un système multi-agents pour des environnements collaboratifs centrés sur la communauté. Ce modèle intègre une technique d'évaluation de la confiance entre les entités et un mécanisme de négociation et de création de contrats intelligents et dynamiques de collaborations. Le reste du document est organisé comme suit : la section 2 présente les travaux connexes. La section 3 décrit notre modèle *Community-OrBAC*. La section 4 présente un cas d'étude d'application et d'évaluation de notre modèle dans un environnement de cloud communautaire. Enfin, la section 5 conclut le document et présente des perspectives pour la suite de nos travaux.

2 Travaux connexes

Les modèles de contrôle d'accès ont longtemps fait l'objet d'une attention particulière de la part des chercheurs[3][27]. Parmi les propositions, le modèle *OrBAC* (Organization Based Access Control)[19] établi à partir des préceptes du *RBAC* (Role-Based Access Control Models)[25] présente des avancées significatives dans la définition de politiques de sécurité orientées organisation. La particularité de ce modèle est qu'il permet d'exprimer des politiques indépendamment de leurs mises en œuvre. Par ailleurs, *OrBAC* permet de formaliser des règles de sécurité en considérant le contexte et apporte de la souplesse dans l'administration des politiques[19]. Toutefois, le modèle *OrBAC* présente des limites dans la spécification de politiques de contrôle pour des systèmes de collaboration entre organisations autonomes. Dans la section suivante, nous présentons des travaux effectués en vue d'adapter *OrBAC* aux systèmes collaboratifs.

2.1 Modèles de contrôle d'accès pour les systèmes collaboratifs

Les systèmes de collaboration permettent à des entités (utilisateurs ou organisations) de collaborer par le partage de données et de services. Frederic Cuppens *et al.* ont présenté dans [11],

le modèle *O2O* permettant de gérer l'interopérabilité dans une collaboration entre des entités ayant défini leurs propres politiques de sécurité. Dans [18], les auteurs ont proposé une extension du modèle *OrBAC* à travers le concept de rôle dans l'organisation (*RiO*). Le modèle appelé *Multi-OrBAC* permet de spécifier des politiques de sécurité propre à chaque organisation et des règles pour gérer les interactions. Le *PolyOR-BAC*, introduit dans [12], est une approche qui utilise le modèle *OrBAC* pour définir la politique de sécurité au sein de chaque organisation d'une part et d'autre part, la technologie des services Web pour faciliter la collaboration et l'interopérabilité entre les organisations.

Les modèles présentés ci-dessus proposent des techniques pour résoudre la question de l'autonomie des organisations dans la définition des règles de contrôle d'accès aux ressources lors d'une collaboration. Cependant, la problématique de la confiance entre ces entités autonomes pour l'établissement de relations durables demeure et doit être abordée.

2.2 Modèles de contrôle d'accès et gestion de la confiance

La confiance entre les organisations est un facteur important pour inciter à la collaboration et garantir la sécurité des ressources partagées. Dans [29], les auteurs ont proposé le modèle *TRUST-OrBAC*. Ce modèle étend *OrBAC* avec la notion de confiance. Ils définissent des vecteurs de confiance permettant d'attribuer des rôles dynamiques aux utilisateurs, et ainsi définir des règles de sécurité pour des environnements multi-organisationnels. Le *Multi-Trust_OrBAC*, a été présenté dans [4]. Ce modèle introduit un tiers de confiance, le *TTP* (Third Trust Party), pour garantir la confiance entre les utilisateurs de différentes organisations dans le cloud. Dans [1], les auteurs ont proposé le modèle *Tr-OrBAC*. Il permet l'évaluation de la confiance entre les organisations sur la base de la logique floue. Les organisations prennent la décision de collaborer ou non en évaluant leurs pairs sur la base d'une valeur de confiance calculée.

En plus de la nécessité de garantir la confiance entre les acteurs, il est important d'apporter plus de flexibilité et de réduire les conflits de règle dans la définition des politiques de sécurité pour ces systèmes distribués de plus en plus complexes, hétérogènes et dynamiques.

2.3 Collaboration et Multi-Agents

La modélisation des interactions complexes entre diverses entités dans les environnements distribués de collaboration est un réel défi. Plusieurs axes de résolution ont été proposés sur la base des systèmes Multi-Agents (MAS)[13]. Un agent est une entité capable de s'adapter, de prendre des décisions et d'exécuter des actions complexes de manière autonome et intelligente. Il est également capable de négocier et de coopérer avec d'autres agents [28]. L'intégration des agents dans les systèmes collaboratifs permet de déployer des infrastructures composées d'entités proactives, autonomes et des mécanismes de partage flexible et dynamique [5]. Dans [2], les auteurs ont présenté un modèle de contrôle d'accès aux ressources partagées dans une coalition dynamique. Ce modèle garantit plus de flexibilité dans la gestion des départs ou l'intégration de nouveaux acteurs dans une coalition. Idrissi *et al.* ont proposé, dans [15], un modèle de contrôle d'accès établi à partir des agents mobiles et des principes du RBAC. Le modèle utilise les caractéristiques de mobilité et d'autonomie des agents mobiles pour combler les limites de communication. Dans [5], les auteurs ont proposé *MA-MOrBAC*, un modèle qui étend le *Multi-OrBAC* grâce à des agents mobiles pour des environnements collaboratifs distribués. Une architecture composée d'agents mobiles permet au modèle d'apporter des améliorations en termes de flexibilité et de robustesse. Un modèle de gestion dynamique de politiques de sécurité à partir d'agents a été proposé dans [22]. Dans cette approche, les agents sont utilisés pour la négociation et l'établissement de contrat intelligent pour le contrôle des accès aux données et de la protection de la vie privée. Un contrat intelligent permet de formaliser un accord applicable automatiquement. Cet accord porte sur les conditions de fourniture d'un service, sur la qualité attendue et les modifications en cas de violations des termes établis[26]. Ce type de contrat peut être déployé au travers d'algorithme auto exécutable utilisant les systèmes multi-agents et la technologie blockchain[6].

2.4 OrBAC

OrBAC est un modèle de contrôle d'accès dérivé du RBAC. Ce modèle a la particularité de permettre la définition de politique de contrôle d'accès en deux niveaux : un niveau constitué d'entités abstraites (Rôle, Vue, Activité) et un niveau d'entités concrètes (Sujet, Objet, Action)[19]. Les entités concrètes exécutent des actions sur

des objets en fonction des règles de la politique de sécurité. À chaque entité abstraite est associée une entité concrète. Un rôle est alors une abstraction d'un groupe d'utilisateurs, une vue représente un ou des objets et une activité fait référence à une ou plusieurs actions. Le modèle *OrBAC* considère le contexte qui permet de modéliser les circonstances dans lesquelles les sujets sont autorisés à réaliser des actions sur des objets[19]. Ce modèle permet de spécifier les relations ci-dessous entre les entités de l'organisation :

- *Permission*(*org*, *r*, *v*, *a*, *c*) : l'organisation *org* autorise le rôle *r* à effectuer l'activité *a* sur la vue *v* dans un contexte *c* ;
- *Habilite*(*org*, *s*, *r*) : l'organisation *org* habilite un sujet *s* dans un rôle *r* ;
- *Utilise*(*org*, *o*, *v*) : l'organisation *org* utilise l'objet *o* dans la vue *v* ;
- *Considère*(*org*, α , *a*) : l'organisation *org* considère l'action α comme faisant partie de l'activité *a* ;
- *Définit*(*org*, *s*, α , *o*, *c*) : l'organisation *org* autorise l'action α du sujet *s* sur l'objet *o* si le contexte *c* est vrai.

TABLE 1 – Spécification d'une permission avec OrBAC

$org \in Organisations, s \in Sujets, \alpha \in Actions, o \in Objets, a \in Activités, v \in Vues, c \in Contextes,$ $Permission(org, r, v, a, c) \wedge$ $Habilite(org, s, r) \wedge$ $Utilise(org, o, v) \wedge$ $Considère(org, \alpha, a) \wedge$ $Définit(org, s, \alpha, o, c) \wedge$ $\rightarrow Est_Permis(s, \alpha, o)$
--

OrBAC permet de formaliser des obligations, des permissions, des interdictions et des recommandations. Une occurrence d'une permission d'un sujet autorisé à effectuer une action sur un objet est présentée dans le tableau 1 ci-dessus. Cette règle signifie que si dans l'organisation *org*, le rôle *r* est autorisé à effectuer l'activité *a* sur la vue *v* quand le contexte *c* est vrai, et si le rôle *r* est assigné au sujet *s*, l'action α fait partie de l'activité *a*, l'objet *o* fait partie de la vue *v*, le contexte *c* est vrai pour les entités (*org*, *s*, α , *o*), alors le sujet *s* est autorisé à réaliser l'action α sur l'objet *o*.

2.5 Discussion

Les systèmes collaboratifs centrés sur la communauté sont composés d'entités réunies dans le but

de collaborer et de partager des ressources. Ces communautés présentent des spécificités particulières en plus de celle des systèmes collaboratifs classiques. En effet, ces systèmes sont définis par des relations transactionnelles, durables et évolutives. Ces relations représentent des connexions sociales contextuelles entre les entités. Par ailleurs, les systèmes axés sur la communauté sont hétérogènes et fondamentalement orientés sur les besoins et intérêts des membres de la communauté [24]. Il ressort de la littérature ci-dessus, que les modèles proposés, traitent soit de l'autonomie des entités collaboratrices dans la spécification des règles de sécurité, soit de la confiance dans les systèmes de contrôle d'accès ou de négociation et d'établissement de contrat. Par conséquent, aucun modèle présenté ne couvre toutes les exigences évoquées ci-dessus d'un système de collaboration centré sur la communauté. Il convient donc de proposer, un système de contrôle d'accès couvrant largement les caractéristiques fondamentales de cette catégorie de système collaboratif. Ce modèle pourra permettre, la définition dynamique et autonome des politiques de contrôle d'accès aux ressources, la gestion de la confiance entre les entités et la formalisation de l'engagement mutuel de chaque entité dans les processus de collaboration. La section suivante décrit notre proposition pour atteindre cet objectif.

3 Modèle proposé

3.1 Contexte dans les systèmes collaboratifs centrés sur la communauté

Une collaboration entre des entités dépend de différents paramètres contextuels liés à ces entités ou aux ressources partagées. Il est essentiel de considérer ces conditions dans la définition des règles de politiques de sécurité des entités engagées. Nous étendons le contexte dans OrBAC décrit dans [10] avec deux nouveaux concepts : le contexte de sécurité et le contexte social comme représenté dans la figure 1.

- **Le contexte de sécurité :**

Le contexte de sécurité est un ensemble d'informations contextuelles permettant de caractériser le niveau de sécurité d'une entité (utilisateurs, ressources) et adapter son comportement en fonction de celui-ci. Ces informations contextuelles peuvent être de divers types, notamment les valeurs de niveau de sécurité, la robustesse des protocoles et des mécanismes de sécurité[17]. Le contexte

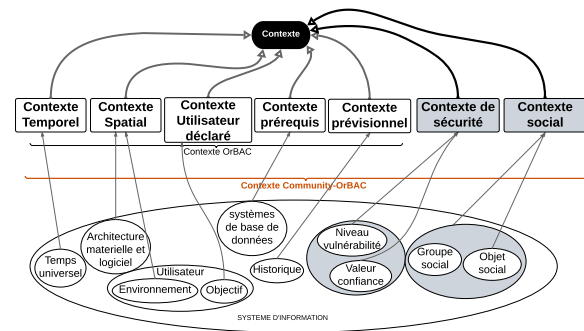


FIGURE 1 – Paramètres contextuels Community-OrBAC

de sécurité permet de spécifier qu'une action donnée d'un sujet sur un objet n'est autorisée qu'en fonction du niveau de vulnérabilité de cet objet et de la confiance accordée à ce sujet. Ainsi, la validation d'un accès requiert l'identification des niveaux de vulnérabilité des ressources et l'évaluation de la confiance entre les organisations engagées dans les interactions. À chaque niveau de vulnérabilité sera associé un seuil de valeur de confiance. Nous désignons le contexte de sécurité par *Contexte_sécurité* et ses deux composantes : *Niveau_vulnérabilité* et *Valeur_confiance*. Les conditions requises pour un contexte donné sont exprimées formellement par des règles logiques et une algèbre de contexte dans des cas de contextes composites [10]. Le contexte de sécurité est formulé dans le tableau 2 ci-dessous et traduit que dans l'organisation *org*, un sujet *s* est autorisé à effectuer une action α sur un objet *o* donné si *Contexte_sécurité* composé de *Niveau_vulnérabilité* de l'objet et *Valeur_confiance* du sujet est vrai.

TABLE 2 – Règle de contexte de sécurité

$org \in Organisations, s \in Sujets, \alpha \in Actions, o \in Objets,$ <i>Définit</i> (<i>org</i> , <i>s</i> , α , <i>o</i> , <i>Niveau_vulnérabilité</i> & <i>Valeur_confiance</i>) \rightarrow <i>Définit</i> (<i>org</i> , <i>s</i> , α , <i>o</i> , <i>Contexte_sécurité</i>)

- **Le contexte social :**

Une communauté requiert la mise en place de stratégies et d'actions communes au profit de ses membres. Par conséquent, nous décrivons le contexte social comme l'intérêt d'une action, d'un sujet ou d'une ressource pour la communauté. Cette perception permet de scinder la communauté en différents groupes

sociaux comme dans [17]. Le contexte social permet d'exprimer une règle qui accorde un privilège pour une action d'un sujet sur un objet en fonction de la portée communautaire de l'objet et du centre d'intérêt du groupe social auquel appartient le sujet. Nous exprimons le contexte social *Contexte_social* au travers de deux attributs : *Objet_social* et *Groupe_social*. Le tableau 3 présente la définition d'une règle de contexte social. Elle signifie que dans l'organisation *org*, un sujet *s* peut exécuter une action α sur un objet *o* donné si le contexte social *Contexte_social* est vrai. Le contexte social est constitué de la portée sociale de l'objet *Objet_social* et du centre d'intérêt du groupe social du sujet *Group_social*.

TABLE 3 – Règle de contexte social

$org \in Organisations, s \in Sujets, \alpha \in Actions, o \in Objets,$
Défini(*org*, *s*, α , *o*, *Objet_social* & *Group_social*)
 \rightarrow *Défini*(*org*, *s*, α , *o*, *Contexte_social*)

3.2 Fonctionnement Community-OrBAC

La collaboration entre les membres de la communauté doit garantir leur autonomie et un accès régulé aux ressources partagées. Pour atteindre cet objectif, nous proposons le *Community-OrBAC*, un modèle de contrôle d'accès utilisant des agents autonomes. Notre modèle vise à fournir des moyens pour définir des politiques de sécurité fiables, dynamiques, tenant compte des paramètres contextuels décrits dans la figure 1 ci-dessus. La collaboration entre entités autonomes peut être source de conflit ou de renoncement partiel ou total à leur autonomie[30]. Nous apportons une réponse à cette problématique d'autonomie des organisations grâce aux systèmes multi-agents. En effet, les agents autonomes sont capables de s'engager dans divers types d'interactions sociales et de coopération avec d'autres agents. Le modèle de résolution de problèmes de coopération proposé dans [30] par M. Wooldridge constitue le fondement des phases du processus de collaboration entre entités de notre système. Ces différentes étapes sont : l'expression du besoin, l'engagement collectif, la négociation d'un contrat intelligent de coopération et l'action collective. Les entités ou sujets intervenants dans une collaboration sont représentés par des agents.

- **L'expression du besoin :**

L'atteinte d'un objectif est conditionnée par l'identification des différentes actions à mener. Une entité s'engage dans une collaboration si elle reconnaît le besoin de collaboration et croit en l'existence d'une entité ou groupe d'entités pouvant lui permettre d'atteindre son but[30]. Par conséquent, le processus de collaboration démarre avec l'expression explicite du besoin par l'agent demandeur de la ressource. Toutefois, une demande ne sera acceptée que si la ressource sollicitée est disponible à la période désirée et son accès est autorisé par la politique de sécurité de l'entité propriétaire. Nous pouvons alors introduire une nouvelle relation *Disponible* entre les entités *organisation* et *objet* dans un contexte temporel. Cette relation sera associée aux relations *Demande*(*org*, *s*, α , *o*) et *Est_Accepté*(*s*, α , *o*) exposées dans [10]. L'expression explicite d'un besoin présente plusieurs avantages. Elle constitue un moyen de suivi du respect des engagements et de gestion des conflits. Le tableau 4 ci-dessous présente la formalisation d'une règle d'acceptation d'une action d'un sujet sur un objet.

TABLE 4 – Règle d'acceptation d'une action sur un objet

$org \in Organisations, s \in Sujets, \alpha \in Actions, o \in Objets, contexte_temporel \in Contextes,$
Demande(*org*, *s*, α , *o*) \wedge
Disponible(*org*, *o*, *contexte_temporel*) \wedge
Est_Permis(*s*, α , *o*) \wedge
 \rightarrow *Est_Accepté*(*s*, α , *o*)

- **L'engagement collectif :**

Cette étape consiste à trouver une entité partenaire capable de contribuer totalement ou partiellement à la réalisation de l'objectif visé. Ce processus commence par l'identification des sujets potentiels pour une collaboration donnée, ensuite l'évaluation de la confiance des entités et enfin la sélection du partenaire idéale.

- **L'identification des potentiels partenaires :**

Les membres de la communauté exposent dans le registre des objets de la communauté les ressources dont ils disposent. Les objets sont déclarés sur la base de règles de

contrôle d'accès établis dans les politiques locales de chaque entité. La prise en compte des contextes social et de sécurité est obligatoire dans le choix du partenaire. Ainsi, une matrice de gouvernance des objets doit définir les exigences de ces contextes. Cette matrice met en exergue le niveau de confiance requis pour un niveau de vulnérabilité d'un objet donné d'une part et d'autre part, le groupe social du sujet exigé pour une ressource de portée sociale donnée.

- **L'évaluation de la confiance :**

La confiance entre les entités est un élément clé favorisant le partage et la collaboration. Cette confiance sera déterminée à partir d'un modèle d'évaluation de la confiance. Ce modèle a été exposé dans une précédente étude présentée dans [23]. Nous évaluons la valeur de confiance de chaque entité relativement à une action donnée sur un objet sur la base d'interactions directes ou recommandées et de la réputation spécifique. Cette valeur de confiance est exprimée comme ci-dessous :

$$\omega_{O_j}^{O_i} = \beta DRT_{O_j}^{O_i} + (1 - \beta) sr_{O_j} \quad (1)$$

$DRT_{O_j}^{O_i}$ est l'opinion de confiance sur la base d'interactions directes ou recommandées entre le propriétaire O_j d'une ressource et le demandeur O_i . Cette valeur est calculée grâce à la logique subjective [16]. sr_{O_j} est la réputation spécifique du propriétaire de l'objet et β le poids de la confiance directe ou recommandée représentant l'influence des interactions antérieures dans la détermination de la valeur la confiance.

- **La sélection du partenaire de collaboration :**

L'entité idéale pour la collaboration est choisie sur la base de la valeur de confiance calculée, de l'algorithme de sélection présenté dans [23] et des exigences définies dans les règles de la politique de sécurité d'accès. La sélection d'une entité ayant exposé volontairement sa capacité à collaborer en vue de contribuer à satisfaire un besoin formellement exprimé par une entité paire constitue l'engagement collectif des deux sujets [30].

- **Négociation et création d'un contrat dynamique de collaboration :**

Lors de cette étape, les agents engagés doivent être d'accord sur les différentes actions à mener dans le but d'atteindre l'objectif de la collaboration. En effet, l'exigence d'autonomie des organisations et l'éventualité de conflits dans les actions nécessitent de parvenir à un accord entre les entités sur la conduite à tenir [30]. Une négociation pourra permettre d'aboutir à cet accord.

La négociation se traduit généralement par des propositions, des contre-propositions et des suggestions pour aboutir à un consensus sur le résultat final. Les actions à effectuer et le cadre de suivi de cette collaboration seront consolidés dans un contrat dynamique et intelligent. Ce processus de négociation est crucial et peut s'avérer complexe. D'où l'utilisation des agents intelligents, autonomes, capables de rendre dynamique la définition, le suivi des politiques et d'éviter ou réduire l'intervention humaine [30]. La mise en place du contrat sera effectuée par des agents représentant les organisations et charger de négocier, de définir et d'actualiser éventuellement les termes de cet accord.

Plusieurs travaux ont mis en exergue la négociation, la définition et la gestion de contrat électronique dans les collaborations entre entités d'organisations distribuées [26][22]. Nous proposons une approche fondée sur le *Web Services Agreement (WS-Agreement)* [14], un standard de spécification des accords de niveaux de service intégrant un protocole de négociation et de renégociation de contrat. Ce protocole composé de trois couches (négociation, accord et service) est adapté à différents types d'environnements dont les infrastructures utilisant les systèmes multi-agents [22][7]. Par ailleurs, le *WS-Agreement* a été associé à des règles de sécurité *OrBAC* afin de permettre à des utilisateurs et des fournisseurs de services cloud de négocier, créer et surveiller des accords de niveau de service (*SLA*) dans [20].

Pour illustrer le fonctionnement de ce protocole, considérons un agent *orgA* demandeur d'une ressource et un agent *orgB* fournisseur. La négociation commence par l'envoi d'une requête de demande d'une ressource exposée par l'agent *orgB* dans le registre des objets. À la réception de la requête, *orgB* répond en transmettant la structure de base d'un contrat *WS-Agreement*. Sur la base de ce modèle

et de ses besoins, $orgA$ construit une offre puis l'envoie à $orgB$. Cette offre peut être validée directement ou faire l'objet d'une contre-proposition en cas de non-conformité avec les contraintes prédéfinies par $orgB$. Cette opération est répétée jusqu'à ce qu'un accord soit trouvé ou non. Une fois l'offre validée, l'agent $orgB$ crée un contrat qu'il signe et le soumet à $orgA$. Ce dernier à son tour marque son approbation en signant l'accord et le partage avec l'agent $orgB$.

• **L'action collective :**

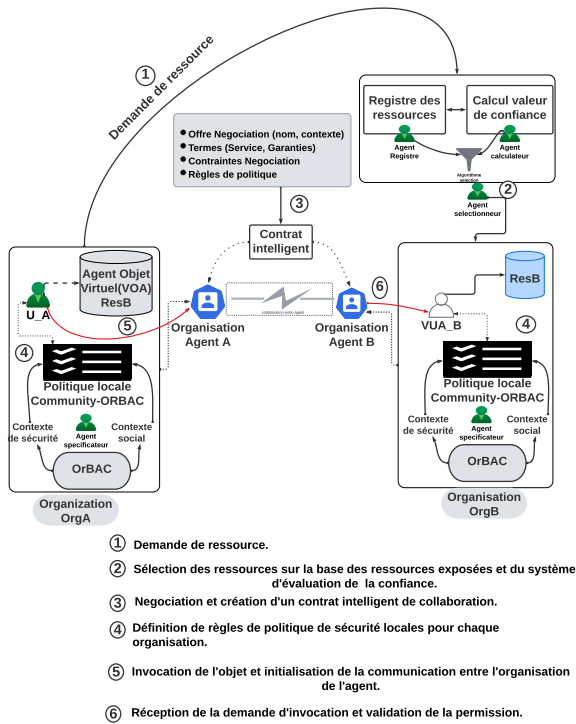


FIGURE 2 – Architecture Community-OrBAC

Les négociations effectuées, les entités ont donc un accord sur l'action collective à réaliser et un contrat pour suivre le processus en vue de l'atteinte de l'objectif. En partant du scénario présenté lors de la phase de négociation, le but de l'action collective est de permettre à un utilisateur u_A de l'organisation $orgA$ d'accéder à une ressource Res_B de l'organisation $orgB$. Afin de préserver l'autonomie des entités dans la définition des règles de sécurité, l'approche adoptée est la procédure d'invocation et de partage de services présentée dans [12] en introduisant les notions d'objet agent virtuel (VOA) et de sujet agent virtuel (VUA). Ces éléments représentent respectivement l'objet distant invoqué et le sujet dans l'organisation fournisseur de l'objet.

De ce fait, un objet VOA_{Res_B} est créé dans la politique locale de $orgA$ et lié à une action $invoker$. Une permission autorisant le rôle associé au sujet u_A d'exécuter l'activité correspondant à l'action $invoker$ sur la vue représentant l'objet VOA_{Res_B} est également définie dans la politique locale de $orgA$. Cette règle est exprimée dans le tableau 5 ci-dessous.

TABLE 5 – Permission au niveau de l'organisation demandeur

$$\begin{aligned}
 & orgA \in Organisations, u_A \in Sujets, invoquer \in Actions, VOA_{Res_B} \in Objets, consulter \in \\
 & Activités, c \in Contexte, \\
 & Permission(orgA, r, consulter, vue_VOA_{Res_B}, c) \wedge \\
 & Habilité(orgA, u_A, r) \wedge \\
 & Utilise(orgA, VOA_{Res_B}, vue_VOA_{Res_B}) \wedge \\
 & Considère(orgA, invoquer, consulter) \wedge \\
 & Définit(orgA, u_A, invoquer, VOA_{Res_B}, c) \wedge \\
 & \rightarrow Est_Permis(u_A, invoquer, VOA_{Res_B})
 \end{aligned}$$

Par ailleurs, dans la politique locale de $OrgB$, un utilisateur virtuel vua_B est créé et associé à un rôle disposant d'une permission permettant d'exécuter une activité sur la vue de l'objet Res_B . L'action $invoker$ va déclencher une communication entre l'agent associé à l'objet virtuel de $OrgA$ et celui de l'organisation $OrgB$. Cette communication se fait conformément aux dispositions du contrat établi entre les deux entités. À la réception du message de l'agent de l'organisation $OrgA$ suite à l'invocation, les actions liées au sujet vua_B sur l'objet Res_B seront exécutées comme illustré dans la figure 2.

Le tableau 6 ci-dessous présente la règle dans la politique locale de $OrgB$.

TABLE 6 – Permission au niveau de l'organisation fournisseur

$$\begin{aligned}
 & orgB \in Organisations, vua_B \in \\
 & Sujets, exécuter \in Actions, Res_B \in \\
 & Objets, Afficher \in Activités, c \in Contexte, \\
 & Permission(orgB, r, Afficher, vue_{Res_B}, c) \wedge \\
 & Habilité(orgB, vua_B, r) \wedge \\
 & Utilise(orgB, Res_B, vue_{Res_B}) \wedge \\
 & Considère(orgB, exécuter, Afficher) \wedge \\
 & Définit(orgB, vua_B, exécuter, Res_B, c) \wedge \\
 & \rightarrow Est_Permis(vua_B, exécuter, Res_B)
 \end{aligned}$$

Notons que toute action de toute entité de la collaboration est conditionnée par une authentification. Cependant, le processus d'authentification des entités est hors du cadre de notre étude.

4 Étude de cas

Cette section décrit une étude de cas de l'application de **Community-OrBAC** dans un cloud communautaire.

Un Cloud Communautaire (3C) permet de regrouper au sein d'une communauté des organisations ayant des exigences (sécuritaire, juridique, etc) et des besoins communs. Le 3C vise à favoriser la coopération entre les membres, améliorer la sécurité de l'infrastructure et réduire les coûts d'investissements [21]. Notre étude repose sur une architecture de 3C fédéré.

Tout d'abord, nous présentons l'architecture de notre cloud communautaire. Ensuite, le scénario d'échange et de négociation d'un contrat intelligent de coopération et enfin la spécification de règles sur la base du modèle *Community-OrBAC*.

4.1 Architecture Cloud communautaire

L'architecture de notre cloud communautaire est présentée dans la figure 3 ci-dessous. Elle est composée de startups regroupées dans une communauté d'entreprise *Com_Startup*. Chaque startup représente une organisation. Ces organisations sont réunies dans l'optique de partager des ressources de type : *Software as a Service(SaaS)*, *Platform as a Service(PaaS)* et *Infrastructure as a Service(IaaS)*. Ces ressources sont référencées dans un registre de ressources. Chaque organisation membre est soit fournisseur et/ou demandeur de ressources.

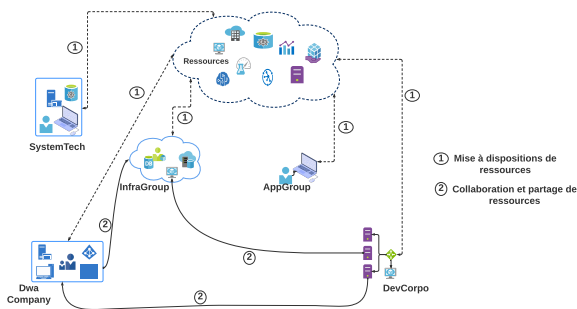


FIGURE 3 – Architecture Cloud Communautaire Com_Startup

4.2 Scénario de collaboration et spécification de règles Community-OrBAC

Nous décrivons dans cette partie un partage de ressources entre les organisations et la définition de règles de politiques d'accès sur la base du modèle *Community-OrBAC*. Considérons que

la startup *DevCorpo* sollicite un cluster de serveurs pour la mise en place d'infrastructures de développement d'applications métiers. La ressource correspondante à ce besoin dans le registre est le *Clusterkub* fourni par la startup *InfraGroup*. Les deux organisations négocient et mettent en place un contrat de collaboration dynamique et évolutif durant toute la période de partage. La figure 4 ci-dessous présente le scénario de collaboration entre les agents *InfraGroup* et *DevCorpo* représentant respectivement le fournisseur et le demandeur.

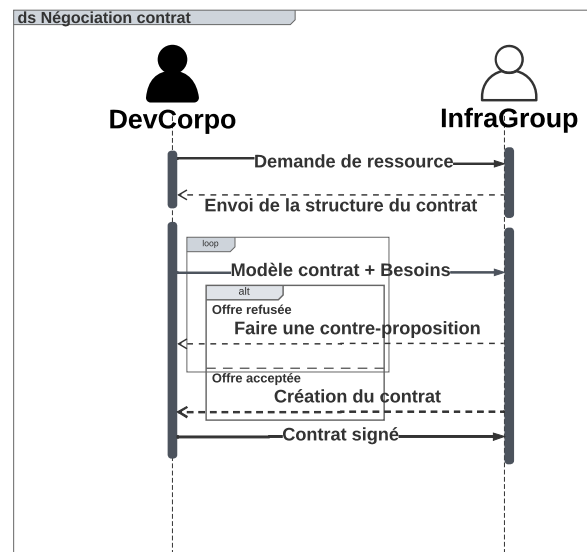


FIGURE 4 – Négociation de contrat de coopération entre deux agents

Le contrat de collaboration établi, nous présentons ci-dessous les règles *Community-OrBAC* pour le partage de la ressource *Clusterkub*. Comme présenté dans le fonctionnement du *Community-OrBAC*(section 3.2), l'accès à une ressource distante nécessite la création de l'agent objet virtuel *VOA_clusterkub*, représentant la ressource désirée *clusterkub* dans l'organisation demandeur, et de l'agent utilisateur virtuel *vua_B* dans la politique de sécurité du propriétaire de la ressource (voir figure 2).

La permission dans la politique locale *DevCorpo* est présentée dans le tableau 7 ci-dessous. Cette permission est accordée dans un contexte *c* considérant le niveau de vulnérabilité et la portée sociale de la ressource *VOA_clusterkub*.

TABLE 7 – Règle dans la politique locale du demandeur *DevCorpo*

$DevCorpo \in Com_Startup, s \in$ $Sujets, invoquer \in$ $Actions, VOA_clusterkub \in$ $Objets, consulter \in Activités, c \in$ $Contexte, r \in Roles$
$Permission(DevCorpo, r, consulter,$ $vue_VOA_clusterkub, c) \wedge$ $Habilite(DevCorpo, s, r) \wedge$ $Utilise(DevCorpo, VOA_clusterkub,$ $vue_VOA_clusterkub) \wedge$ $Considere(DevCorpo, invoquer, consulter) \wedge$ $Définit(DevCorpo, s, invoquer,$ $VOA_clusterkub, c)$ $\rightarrow Est_Permis(s, invoquer, VOA_clusterkub)$

Le tableau 8 décrit la règle d'accès dans la politique du propriétaire *InfraGroup* dans un contexte évalué sur la base de la valeur de confiance et du groupe social du sujet vua_B .

TABLE 8 – Règle dans la politique locale du fournisseur *Infragroup*

$InfraGroup \in Com_Startup, vua_B \in$ $Sujets, exécuter \in Actions, Clusterkub \in$ $Objets, Afficher \in Activités, c \in$ $Contexte, r \in Roles,$
$Permission(InfraGroup, r, Afficher,$ $Vue_clusterkub, c) \wedge$ $Habilite(InfraGroup, vua_B, r) \wedge$ $Utilise(InfraGroup, Clusterkub,$ $Vue_clusterkub) \wedge$ $Considere(InfraGroup, exécuter, Afficher) \wedge$ $Définit(InfraGroup,$ $vua_B, exécuter, Clusterkub, c)$ $\rightarrow Est_Permis(vua_B, exécuter, Clusterkub)$

L'étude de cas présente différents avantages apportés par notre modèle dans une collaboration entre deux entités de façon générale et spécifiquement entre membres d'une même communauté. En effet, l'utilisation des agents autonomes et la démarche de négociation proposée exposent des interactions de type pair à pair, consensuelles, garantissant une souplesse et une indépendance de chaque entité dans la gestion de ses utilisateurs et du contrôle des ac-

cès à ses ressources. Par ailleurs, au regard des apports significatifs apportés par la combinaison des systèmes multi-agents et des technologies blockchain [9][6], la négociation et la création de contrats dynamiques peuvent être déployées grâce à des contrats intelligents auto-exécutables. Les agents serviront de sources de données (modifications des clauses du contrat, pénalités, etc) pour des algorithmes représentant les contrats intelligents. En outre, l'identification des niveaux de sécurité des ressources partagées et l'évaluation de la confiance est fortement recommandée dans la mise en place de stratégie de cybersécurité dans le *cloud computing*[8]. De façon spécifique pour des organisations centrées sur la communauté, les membres peuvent mettre à la disposition de leurs pairs des ressources avec des exigences d'accessibilité réduites (niveau de confiance, sans contrepartie financière, etc). Cette action visant à enrichir la communauté et dans l'intérêt de ses membres peut être une prérogative à l'adhésion. Un tel scénario serait difficilement envisageable pour des systèmes non communautaires. Le contexte de sécurité et le contexte social constituent par ailleurs des critères supplémentaires et fiables pour définir des règles de sécurité robustes et personnalisées.

5 Conclusion and Perspectives

Dans cet article, nous avons présenté le *Community-OrBAC*, un modèle de contrôle d'accès aux ressources utilisant les agents pour les systèmes de collaboration centrés la communauté. Le modèle s'appuie sur des agents autonomes pour représenter les entités du système et favoriser la spécification dynamique et autonome des politiques de sécurité. Les systèmes multi-agents permettent d'exposer un processus de collaboration consensuel et évolutif à travers la négociation et la création de contrat de coopération entre les entités. Par ailleurs, le modèle étend *OrBAC* avec les notions de contexte de sécurité et contexte social et intègre un système d'évaluation de la confiance entre les entités. En outre, un cloud communautaire a servi de cadre d'application du modèle. Dans de futurs travaux, nous envisageons de proposer des mécanismes de sécurisation des échanges, d'authentification des agents et de suivi des accords négociés. Une implémentation du modèle est également en cours d'expérimentations.

Références

- [1] N. A. Aali, A. Baina, and L. Echabbi. Tr-OrBAC : Towards a Trust Framework for Collaborative Sys-

- tems in Critical Information Infrastructures. *Journal of Network and Innovative Computing*, 4 :106–115, 2016.
- [2] B. I. Abdelkrim, A. Baina, C. Feltus, J. Aubert, M. Bellafkih, and D. Khadraoui. Coalition-OrBAC : An agent-based access control model for dynamic coalitions. *Advances in Intelligent Systems and Computing*, 745(May 2020) :1060–1070, 2018.
- [3] R. Ausanka-Cruces. Methods for access control : Advances and limitations. *Harvey Mudd College*, pages 1–5, 2001.
- [4] M. B. Saidi and A. Marzouk. Multi-Trust_OrBAC : Access Control Model for Multi-Organizational Critical Systems Migrated To the Cloud. *International Journal of Soft Computing and Engineering*, (3) :2231–2307, 2013.
- [5] Z. B. Yahya, F. B. Ktata, and K. Ghedira. MA-MOrBAC : A distributed access control model based on mobile agent for multi-organizational, collaborative and heterogeneous systems. *Lecture Notes in Computer Science*, 10694 LNCS :101–114, 2018.
- [6] Ricardo Barbosa, Ricardo Santos, and Paulo Novais. Smart Contracts Based on Multi-agent Negotiation. In Fernando De La Prieta, Alia El Bolock, Dalila Durães, João Carneiro, Fernando Lopes, and Vicente Julian, editors, *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Social Good. The PAAMS Collection*, pages 104–114, Cham, 2021. Springer International Publishing.
- [7] D. Battr, F. M. T. Brazier, K. P. Clark, M. Oey, A. Papaspyrou, W. Oliver, P. Wieder, and W. Ziegler. A Proposal for WS-Agreement Negotiation. pages 233–241, 2010.
- [8] Oliver Borchert and Allen Tan. Implementing a Zero Trust. *national insutute of standard and technology (NIST)*, (March), 2020.
- [9] Davide Calvaresi, Alevtina Dubovitskaya, Jean Paul Calbimonte, Kuldar Taveter, and Michael Schumacher. Multi-agent systems and blockchain : Results from a systematic literature review. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10978 LNAI(June) :110–126, 2018.
- [10] F. Cuppens and N. Cuppens-Boulahia. Modeling contextual security policies. *International Journal of Information Security*, 7(4) :285–305, 2008.
- [11] F. Cuppens, N. Cuppens-Boulahia, and C. Coma. O2O : Virtual private organizations to manage security policy interoperability. *Lecture Notes in Computer Science*, 4332 LNCS :101–115, 2006.
- [12] Y. Deswarte and A. A. E. Kalam. PolyOrBAC : An Access Control Model for Inter-Organizational Web Services. *IGI Global*, pages 901–923, 2009.
- [13] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. Multi-Agent Systems : A Survey. *IEEE Access*, 6 :28573–28593, 2018.
- [14] John Rofrano Ibm. Web Services Agreement Specification (WS-Agreement). pages 1–81, 2007.
- [15] H. Idrissi, M. Ennahbaoui, E. M. Souidi, A. Revel, and S. Elhajji. Access control using mobile agents. *International Conference on Multimedia Computing and Systems*, pages 1216–1221, 2014.
- [16] A. Jøsang, S. Pope, and R. Hayward. Trust Network Analysis with Subjective Logic. *Conferences in Research and Practice in Information Technology Series (2006)*, 48 :85–94, 2006.
- [17] V. Jovanovikj, D. Gabrijelčič, and T. Klobučar. A conceptual model of security context. *International Journal of Information Security*, 13(6) :571–581, 2014.
- [18] A. A. El Kalam and Y. Deswarte. Multi-OrBAC : A New Access Control Model for Distributed, Heterogeneous and Collaborative Systems. *8th IEEE International Symposium on Systems and Information Security*, 1, 2006.
- [19] A. A.E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Mieke, C. Saurer, and G. Trouessin. Organization based access control. *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, (May 2014) :120–131, 2003.
- [20] Y. Li, N. Cuppens-Boulahia, J. M. Crom, F. Cuppens, and V. Frey. Expression and enforcement of security policy for virtual resource allocation in IaaS cloud. *IFIP International Federation for Information Processing*, 471 :105–118, 2016.
- [21] K. Marzantowicz and L. Paciorkowski. Community cloud : Closing the gap between public and private. *IGI Global*, pages 39–55, 2017.
- [22] J. E. Mokhtari, A. A. E. Kalam, S. Benhaddou, and J. P. Leroy. Dynamic Management of Security Policies in PrivOrBAC. *International Journal of Advanced Computer Science and Applications*, 12(6) :693–701, 2021.
- [23] R. N'goran, J.-L. Tetchueng, G. Pandry, Y. Kermarrec, and O. Asseu. Trust Assessment Model Based on a Zero Trust Strategy in a Community Cloud Environment. *Engineering*, 14(11) :479–496, 2022.
- [24] F. Paci, A. Squicciarini, and N. Zannone. Survey on access control for community-centered collaborative systems. *ACM Computing Surveys*, 51(1), 2018.
- [25] Ravi S Sandhu, Hal L Feinstein, Charles E Youman, and Edward J Coyne. Role-Based Access Control Models. 29(2) :38–47, 1996.
- [26] V. Scoca, R. B. Uriarte, and R. D. Nicola. Smart Contract Negotiation in Cloud Computing. *IEEE International Conference on Cloud Computing*, 2017-June :592–599, 2017.
- [27] P. G. Shynu and K. John Singh. A comprehensive survey and analysis on access control schemes in cloud environment. *Cybernetics and Information Technologies*, 16(1) :19–38, 2016.
- [28] Kwang Mong Sim. Agent-based approaches for intelligent intercloud resource allocation. *IEEE Transactions on Cloud Computing*, 7(2) :442–455, 2019.
- [29] K. Toumi, C. Andrés, and A. Cavalli. Trust-OrBac : A trust access control model in multi-organization environments. *Lecture Notes in Computer Science*, 7671 LNCS(August 2015) :89–103, 2012.
- [30] M. Wooldridge. *Reasoning about Rational Agents*. The MIT Press, 2003.

