



HAL
open science

Actes des 29es Journées Francophones sur les Systèmes Multi-Agents

Jean-Paul Jamont

► **To cite this version:**

Jean-Paul Jamont. Actes des 29es Journées Francophones sur les Systèmes Multi-Agents. Plate-Forme Intelligence Artificielle, Association Française pour l'Intelligence Artificielle, 2021. hal-04566129

HAL Id: hal-04566129

<https://ut3-toulouseinp.hal.science/hal-04566129v1>

Submitted on 2 May 2024

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

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



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



AfIA

Association française
pour l'Intelligence Artificielle

JFSMA

Journées Francophones sur les Systèmes Multi-Agents

PFIA 2021



Crédit photo : [Flicr/xlibber](#)

Table des matières

Jean-Paul JAMONT

Éditorial	4
Comité de programme	5
A. Bonnefond, O. Simonin, I. Guerin-Lassous	
Extension des Modèles de Flocking aux Environnements avec Obstacles et Communications Dégradées	7
N. Aky, S. Giroux, D. Payet, R. Courdier	
Modèle réutilisable et plus inclusif de système ambiant dédié à l'assistance basée sur le paradigme multi-agent (présentation courte)	16
E. Beauprez, A.-C. Caron, M. Morge, J.-C. Routier	
Une stratégie de négociation multi-agents pour réduire la durée moyenne de réalisation	27
F. Quinton, C. Grand, C. Lesire	
Enchères pour le Maintien des Communications lors de l'Allocation de Tâches	37
E. Beauprez, L. Bigand, A.-C. Caron, M. Morge, J.-C. Routier	
Réaffectation de tâches de la théorie à la pratique : état de l'art et retour d'expérience (présentation courte)	47
K. Chapuis, P. Taillandier, B. Gaudou, A. Brugière, A. Drogoul	
COMOKIT : un environnement générique et modulaire pour analyser les impacts des politiques d'intervention contre l'épidémie de COVID-19 (présentation courte)	57
V. Sicard, M. Andraud, S. Picault	
L'organisation comme Design Pattern dans les systèmes multi-agents multi-niveaux (présentation courte)	67
A. O. Diallo, G. Lozenguez, A. Doniec, R. Mandiau	
Usage des parkings relais dans les comportements de déplacements intermodaux : Génération de demande de population d'agents à partir de données réelles (présentation courte)	77
R. Chaput, J. Duval, O. Boissier, M. Guillermin, S. Hassas	
Approche multi-agent combinant raisonnement et apprentissage pour un comportement éthique 87	
M. Bettinelli, M. Ocelllo, D. Genthial.	
ABSG : un modèle d'agent socialement inspiré pour la formation de coalitions	97
G. Picard	
Planification multi-utilisateurs et multi-satellites de tâches d'observation dans des constellations avec portions d'orbites exclusives	107
A. Daoud, F. Balbo, P. Gianessi, G. Picard	
Un modèle agent générique pour la comparaison d'approches d'allocation de ressources dans le domaine du transport à la demande	117
J.-P. Delahaye, P. Mathieu	
Coopération généralisée et attracteurs au dilemme itéré du prisonnier	127
J. Guéron, G. Bonnet	
De la diversité des jeux de coalitions à utilité transférable (présentation courte)	138

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 de chercheurs francophones qui étudient, utilisent et font évoluer le paradigme multi-agent pour adresser des problématiques issues de domaines liés à l'informatique (intelligence et vie artificielle, génie logiciel, robotique collective, etc.), à l'automatique et aux sciences humaines et naturelles (économie, sociologie, éthologie, etc.). Ces journées visent à faire le point sur les différents travaux de la communauté scientifique du domaine mais également à créer un espace de débats d'idées et d'échanges de proximité entre les chercheurs confirmés et les jeunes chercheurs de la discipline. Ces journées sont également largement ouvertes aux entreprises impliquées dans des projets de recherche, de développement ou d'innovation.

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

Ces actes rassemblent les **versions préliminaires** des 14 contributions sélectionnées et présentées lors de la vingt-neuvième édition des JFSMA du 28 au 30 juin 2021. Ils réunissent des articles qui proposent des modèles, des méthodologies, des techniques et des outils qui permettent notamment de répondre à différents problèmes que l'on peut décliner selon quatre aspects :

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

Je remercie le comité de programme pour la qualité de son arbitrage dans le processus de relecture des articles soumis, ainsi que les comités de programmation et d'organisation de la plate-forme IA pour avoir géré l'ensemble des aspects techniques et "logistiques". Je remercie aussi particulièrement les bergers qui accompagnent la révision de certains articles et les relecteurs sollicités tardivement pour proposer de nouveaux éclairages sur certaines contributions.

Merci également au comité consultatif pour la confiance qu'il m'a témoignée en me proposant la présidence des JFSMA'21.

Mes remerciements vont aussi aux 91 chercheurs francophones d'universités du Cameroun, du Canada, de France, du Luxembourg, de Madagascar, du Sénégal, de Suède et du Viet-Nam pour avoir participé au processus de soumission.

Enfin, je remercie très chaleureusement les 13 doctorants qui ont participé à la session "Jeunes Chercheurs en SMA".

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

Les **versions finales** des articles sont disponibles dans les actes édités des Journées Francophones sur les Systèmes Multi-Agents :

Jean-Paul Jamont: Collectifs Cyber-Physiques — Vingt-neuvièmes Journées Francophones sur les Systèmes Multi-Agents, JFSMA 2021, Bordeaux, France, 25 - 30 juin, 2021. Cépaduès 2020, ISBN 9782364939004.

Jean-Paul JAMONT

Comité de programme

Président

- Jean-Paul Jamont (LCIS, Université Grenoble Alpes, France)

Comité consultatif

- Catherine Garbay (LIG, CNRS, Université Grenoble Alpes, France)
- Fabien Michel (LIRMM, Université de Montpellier, France)
- Gauthier Picard (DTIS - SYD, Onera, France)
- Nicolas Sabouret (LIMSI, Université Paris Sud, France)
- Olivier Simonin (CITI lab, INSA Lyon, France)

Membres

- Emmanuel Adam (LAMIH, Université Polytechnique Hauts-De-France, France)
- Carole Adam (LIG, CNRS, Université Grenoble Alpes, France)
- Flavien Balbo (Limos, Ecole des Mines de Saint-Etienne, France)
- Aurélie Beynier (LIP6, Université Pierre et Marie Curie, France)
- Olivier Boissier (Limos, Ecole des Mines de Saint-Etienne, France)
- Grégory Bonnet (GREYC, Université Caen Normandie, France)
- Valérie Camps (IRIT, Université Paul Sabatier de Toulouse, France)
- Florent Carlier (CREN, Le Mans Université, France)
- Anne-Cécile Caron (CRISTAL, Université de Lille, France)
- François Charpillat (LORIA, Inria, Nancy, France)
- Vincent Chevrier (LORIA, Université de Lorraine, France)
- Caroline Chopinaud (Hub France IA, France)
- Rémy Courdier (LIM, Université de la Réunion, France)
- Yves Demazeau (LIG, CNRS, Université Grenoble Alpes, France)
- Stéphane Galland (CIAD, Université de Technologie Belfort-Montbéliard, France)
- Emmanuelle Grislin (LAMIH, INSA, Université Polytechnique Hauts-De-France, France)
- Zahia Guessoum (LIP6 & CReSTIC, Université de Reims, France)
- Salima Hassas (LIRIS, Université de Lyon, France)
- Cédric Herpson (LIP6, Université Pierre et Marie Curie, France)
- Guillaume Hutzler (IBISC, Université Evry Val d'Essonne, France)
- Jean-Paul Jamont (LCIS, Université Grenoble Alpes, France)
- Elsy Kaddoum (IRIT, Université Toulouse Jean Jaurès, France)
- Christophe Lang (FEMTO-ST, Université de Franche-Comté, France)
- Domitile Lourdeaux (UTC, Université de Technologie de Compiègne)
- René Mandiau (LAMIH, Université Polytechnique Hauts-De-France, France)
- Nicolas Marilleau (UMMISCO, IRD, France)
- Philippe Mathieu (CRISTAL, Université de Lille, France)
- Bruno Mermet (GREYC, Université Caen Normandie, France)
- Frédéric Migeon (IRIT, Université Paul Sabatier de Toulouse, France)
- Maxime Morge (CRISTAL, Université de Lille, France)
- Gildas Morvan (LG2IA, Université d'Artois, France)
- Jean-Pierre Müller (SENS, CIRAD, Univ Montpellier, France)
- Antoine Nongaillard (CRISTAL, Université de Lille, France)
- Michel Occello (LCIS, Univ. Grenoble Alpes, France)
- Wassila Ouerdane (MICS, École CentraleSupélec, France)
- Alexandre Pauchet (LITIS, INSA Rouen Normandie, France)
- Denis Payet (LIM, Université de la Réunion, France)
- Sébastien Picault (INRAE, UMR BIOEPAR, France)
- Suzanne Pinson (LAMSADE, Université Paris Dauphine, France)
- Éric Platon (CosmosX, Japon)
- Clément Raievsky (LCIS, Université Grenoble Alpes, France)
- Lilia Rejeb (SMART Lab, Institut Supérieur de Gestion, Tunisie)

- Valérie Renault (CREN, Le Mans Université, France)
- Julien Saunier (LITIS, INSA Rouen Normandie, France)
- Cristina Stoica Maniu (L2S, École CentraleSupélec, France)
- Patrick Tailliandier (COGIT, IGN, France)
- Paul Théron (Thalès TCS, France)
- Laurent Vercouter (LITIS, INSA Rouen Normandie, France)
- Mahdi Zargayouna (IFSTTAR Université Marne-la-Vallée, France)

Extension des Modèles de Flocking aux Environnements avec Obstacles et Communications Dégradées

Alexandre Bonnefond^{1,2}, Olivier Simonin¹, Isabelle Guerin-Lassous²

¹ CHROMA team, CITI, INSA Lyon & Inria

² DANTE team, LIP, Lyon 1 & Inria

prenom.nom@inria.fr

Résumé

Dans ce papier, nous étudions des modèles de flocking existants et proposons des extensions afin d'améliorer leurs performances dans des environnements ayant des obstacles impactant les communications. Souvent décrits comme des systèmes robustes, il y a cependant un manque de compréhension quant à la comparaison de ces modèles et à l'impact de la qualité des communications sur l'échange des données de contrôle. Nous étendons deux modèles standards afin de renforcer leurs capacités à rester connectés dans des environnements avec différentes distributions d'obstacles. En tenant compte de la propagation radio, nous modélisons comment les obstacles impactent les communications dans un simulateur que nous utilisons notamment pour optimiser les paramètres du flocking. Les résultats des simulations montrent l'efficacité des modèles proposés et la façon dont ils s'adaptent à ces nouvelles contraintes environnementales.

Mots-clés

Modèles de flocking, robotique en essaim, simulation des communications.

Abstract

In this paper, we study existing flocking models and propose extensions to improve their abilities to deal with environments having obstacles impacting the communication quality. Often depicted as robust systems, there is yet a lack of understanding how flocking models compare and how they are impacted by the communication quality when they exchange control data. We extend two standard models to improve their ability to stay connected while evolving in environments with different obstacles distributions. By taking into account the radio propagation, we model the obstacles' impact on communications in a simulator that we use to optimize flocking parameters. The simulation results show the efficiency of the proposed models and how they adapt to different environmental constraints.

Keywords

Flocking models, swarm robotics, simulation of communications.

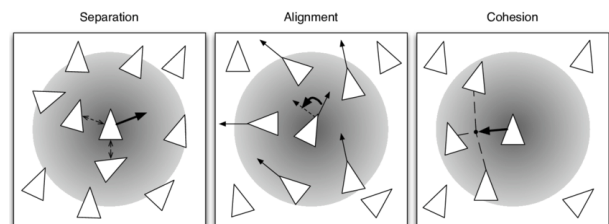


FIGURE 1 – Flocking : les 3 interactions définies par Reynolds (illustration de Bouraqadi et Doniec [1])

1 INTRODUCTION

Depuis sa formulation par Craig Reynolds [2] en 1987, le principe du flocking n'a cessé de susciter un intérêt grandissant. Fondé sur trois interactions simples entre agents, la cohésion, la séparation et l'alignement (voir Fig. 1) le flocking est un modèle décentralisé qui permet à un essaim de se déplacer dans divers types d'environnements. Si ce modèle a été initialement pensé comme un contrôleur fondé sur les perceptions locales, le développement des drones autonomes et leurs moyens de communications permet d'adapter le flocking à ce type d'agents. Les drones sont capables aujourd'hui d'échanger avec leur voisins proches, via un canal de communication haute fréquence, les informations nécessaires au flocking comme leur vitesse et leur position. Ainsi, la plupart des travaux récents [3][4][5][6] considèrent des flockings exploitant les communications. Cependant, ils ne prennent pas en compte les caractéristiques des communications et comment elles sont impactées par la présence d'obstacles. Des hypothèses trop simples sont un frein pour la définition de modèles exploitables en conditions réelles. Les travaux de [7] donnent une bonne idée des effets de communications réalistes sur un comportement émergent. Cependant, les contrôleurs proposés sont trop simples pour refléter comment les modèles de flocking standards sont affectés.

Le but de cet article est de comparer les modèles de flockings et d'examiner comment ils sont impactés par la qualité des communications pour l'échange de données de contrôle, cette qualité variant selon la distribution des obstacles. Dans ce cadre, nous proposons d'étendre deux modèles de référence (Olfati-Saber [3] et Vásárhelyi/

Vicsek [5]) pour améliorer leur robustesse à la présence d'obstacles et leur décentralisation. Les modèles sont évalués à l'aide du simulateur de flockings de Viragh *et al.* [8]. Nous étendons ce simulateur pour à la fois disposer de modèles de communication plus réalistes et d'environnements plus variés contenant des obstacles. En combinant cet outil avec un processus d'optimisation des paramètres nous montrons que les modèles de flocking proposés sont plus robustes dans les environnements très contraints.

En Section 2, nous présentons les modèles de flocking existants avec un focus sur les deux modèles de référence étendus dans cet article. Ces extensions sont présentées en détail en Section 3. La Section 4 présente les modèles de communication réalistes introduits dans les simulations, et comment ils permettent de redéfinir la notion de voisinage. Nous comparons alors ces différents modèles en Section 5 en considérant des environnements variés en terme d'obstacles. Finalement, nous concluons et discutons de la poursuite de ces travaux.

2 Modèles de flocking existants

Les modèles de flocking sont généralement utilisés pour réaliser le déplacement collectif d'une flotte d'agents ou de robots. Ces modèles reposent sur trois interactions fondamentales introduites par Reynolds [2] :

- **La séparation** des voisins pour éviter les collisions
- **L'alignement** avec les voisins pour une direction/vitesse proche des voisins
- **L'attraction** vers les voisins pour maintenir la cohésion.

On note N le nombre d'agents dans le système. Chaque agent i est défini par sa position \mathbf{q}_i et sa vitesse \mathbf{v}_i (les notations en gras représentent des vecteurs). Chaque agent a un ensemble de voisins N_i avec lesquels il interagit :

$$N_i = \{j \in \llbracket 1, \dots, N \rrbracket; j \neq i : \|\mathbf{q}_j - \mathbf{q}_i\| \leq r\} \quad (1)$$

où r est la distance inter-agent maximum sous laquelle deux agents sont considérés voisins.

La plupart des modèles de flocking sont définis sur la base de ces trois interactions, mais lorsqu'elles sont traduites en contrôle bas-niveau plusieurs variantes et extensions existent. Parmi ces modèles de flocking, on trouve dans la littérature les approches leader-follower [9][6], champs de potentiels [10], behavior-based [11], les structures virtuelles [12] ou encore les modèles construits par apprentissage par renforcement [13] [14]. Selon l'approche choisie ce ne sont pas les mêmes caractéristiques qui sont mises en avant et le comportement du flocking peut être différent.

Aujourd'hui le flocking est utilisé pour répondre à différentes problématiques et le modèle de Reynolds est intégré dans des contrôleurs multi-agent plus complexes comme celui d'Olfati-Saber [3] (noté O-S plus loin). Dans son travail, en 2006, l'auteur définit un cadre mathématique approprié afin d'assurer la stabilité de son modèle. Cet aspect est primordial dans la théorie du contrôle et garantit un comportement non chaotique ce qui est indispensable pour des applications basées sur l'utilisation de drones par

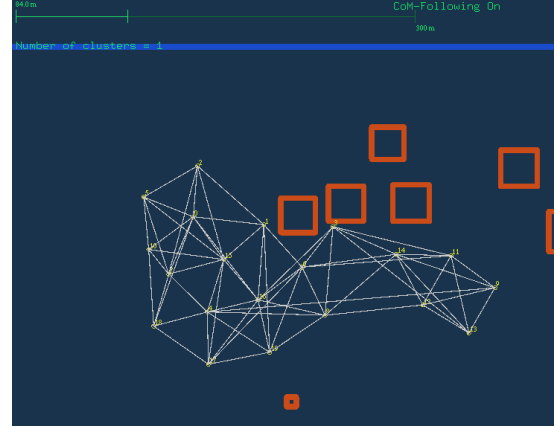


FIGURE 2 – Représentation des agents avec les liens de communication effectifs (utilisés pour interagir) utilisant le modèle d'Olfati-Saber (O-S), obstacles en orange

exemple. Olfati-Saber introduit également des fonctionnalités telles que l'évitement d'obstacles ainsi qu'une "rétroaction de navigation" donnant aux agents une direction globale dans laquelle se diriger. Cette dernière caractéristique est capitale pour éviter la fragmentation de la flotte (*i.e.* la formation de multiple clusters). Pour ces raisons, ce modèle a été largement réutilisé dans la littérature [15] [14] [16] avec quelques modifications afin de servir différents objectifs. La dynamique utilisée par Olfati-Saber est basée sur une régulation en vitesse (double intégrateur) :

$$\begin{cases} \dot{\mathbf{q}}_i = \mathbf{v}_i \\ \dot{\mathbf{v}}_i = \mathbf{u}_i \end{cases}, \quad (2)$$

et l'entrée du contrôle \mathbf{u}_i , appliquée à chaque agent i repose sur trois termes :

$$\mathbf{u}_i = \mathbf{f}_i^g + \mathbf{f}_i^d + \mathbf{f}_i^r, \quad (3)$$

où \mathbf{f}_i^g est un terme basé sur un gradient combinant à la fois la séparation et l'attraction, \mathbf{f}_i^d est le terme d'alignement de vitesse qui agit comme une force de frottement visqueux et \mathbf{f}_i^r est la rétroaction de navigation. Ce dernier terme agit comme si il y avait un leader virtuel dont la position et la vitesse seraient connues par tous les agents. Nous verrons dans la Section 3 que nous utilisons le travail d'O-S comme première couche pour construire un nouveau modèle qui répond à nos attentes. Néanmoins, comme nous considérons uniquement un *pur flocking* dans ce papier, que nous définissons comme un flocking sans direction préférentielle ou but de haut niveau connus des agents, nous ne prendrons pas en compte ce terme de rétroaction. La Fig.2 représente l'évolution des agents utilisant le modèle d'O-S. On se rend compte de la régularité du "maillage" formé par les agents, un des critères qui fait la réputation de ce modèle.

Un autre formalisme majeur du flocking a été introduit par Vicsek [17] en 1995 et considère que les agents sont "auto-propulsés". Cette caractéristique, qui dérive d'observations

des matériaux ferromagnétiques, sera très importante dans notre formulation dans le sens où elle mène à un *flocking pur* (comme défini plus haut). S'appuyant sur cette idée, Vásárhelyi et al. ont développé un framework [5] qui rassemble un simulateur incluant les caractéristiques d'un environnement réaliste telles que les délais de communication, l'inertie, la fréquence de fonctionnement des capteurs, les approximations des capteurs embarqués et les bruits ambiants. Le modèle est défini par une équation d'accélération \mathbf{a}_i pour chaque agent i (voir [8]) :

$$\mathbf{a}_i = \boldsymbol{\eta}_i + \frac{\mathbf{v}_i^d - \mathbf{v}_i - \mathbf{v}_i^s}{|\mathbf{v}_i^d - \mathbf{v}_i - \mathbf{v}_i^s|} \cdot \min \left\{ \frac{\mathbf{v}_i^d - \mathbf{v}_i - \mathbf{v}_i^s}{\tau_{CTRL}}, a_{max} \right\}, \quad (4)$$

où $\boldsymbol{\eta}_i$ représente le bruit extérieur (tel que l'effet du vent), \mathbf{v}_i^s correspond au bruit interne sur la mesure de vitesse (due aux approximations du GPS), τ_{CTRL} est le temps d'établissement du contrôleur en vitesse, a_{max} est l'accélération maximum fixée et \mathbf{v}_i^d est la vitesse désirée :

$$\mathbf{v}_i^d = \frac{\tilde{\mathbf{v}}_i^d}{|\tilde{\mathbf{v}}_i^d|} \min \{ |\tilde{\mathbf{v}}_i^d|, v^{max} \}, \quad (5)$$

où v^{max} est la vitesse maximale autorisée pour chaque agent et $\tilde{\mathbf{v}}_i^d$ est défini de la manière suivante :

$$\tilde{\mathbf{v}}_i^d = \frac{\mathbf{v}_i}{|\mathbf{v}_i|} v^{flock} + \mathbf{v}_i^{rep} + \mathbf{v}_i^{frict} + \sum_s \mathbf{v}_{is}^{wall} + \sum_s \mathbf{v}_{is}^{obstacle} \quad (6)$$

où \mathbf{v}_i est la vitesse réelle de l'agent i , v^{flock} est la vitesse souhaitée, \mathbf{v}_i^{rep} est le terme de répulsion basé sur le modèle du demi ressort, \mathbf{v}_i^{frict} est le terme d'alignement qui synchronise le mouvement au sein de la flotte, \mathbf{v}_{is}^{wall} et $\mathbf{v}_{is}^{obstacle}$ correspondent aux termes utilisés pour l'évitement d'obstacles et de murs. La combinaison de cette loi de contrôle ajoutée aux expériences réelles en extérieur présentées dans l'article font de ce framework une piste très intéressante pour nos travaux et c'est pour cette raison que nous allons nous en inspirer pour proposer une version étendue. La Fig.3 représente l'évolution des agents avec le modèle de Vásárhelyi. Bien que le maillage créé par les agents semble moins régulier que pour le modèle d'O-S (Fig.2) il n'est pas moins robuste comme le montreront les résultats de la Section 5.

Bien que ces différents modèles atteignent de bonnes performances en simulation, en termes de stabilité et de robustesse, de nombreux problèmes surgissent quand il s'agit de passer à des expériences à grande échelle (avec des vrais drones en extérieur). En effet, comme expliqué dans l'article [5], en pratique, de nombreuses pertes de communication surviennent et c'est un inconvénient majeur lorsque l'on travaille avec des drones.

3 Extension des modèles de flocking

Afin d'éviter les désavantages rencontrés par les modèles présentés en section 2 quand ils sont mis dans un environnement plus réaliste, nous proposons de nouveaux modèles de flocking. Ces derniers exploitent les interactions fonda-

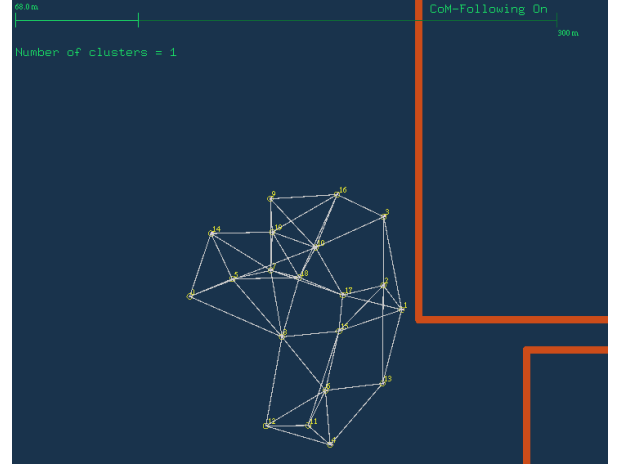


FIGURE 3 – Représentation des agents avec les liens de communication effectifs (utilisés pour interagir) utilisant le modèle de Vásárhelyi.

mentales des modèles précédents [5] et [3] avec de nouvelles fonctionnalités visant à plus de robustesse en présence d'obstacles.

3.1 Modèle VAT : Vásárhelyi + Attraction

De part sa stabilité, le contrôleur en vitesse de Vásárhelyi définit en (6) ne contient pas de terme d'attraction dans sa forme initiale. Cependant, une fois mis dans un environnement fortement contraint (délais et erreurs de communication, approximation de position et multiples obstacles) nous nous sommes rendu compte que la fréquence des coupures de communication augmentait rapidement menant à un accroissement du nombre de fragmentations de la flotte. Nous avons donc décidé d'introduire un terme d'attraction basé sur le modèle du demi-ressort afin d'améliorer la cohésion de la flotte dès que possible :

$$v_{ij}^{att} = \begin{cases} p^{att} (r_0^{att} - q_{ij}) \frac{\mathbf{q}_i - \mathbf{q}_j}{q_{ij}}, & \text{si } q_{ij} > r_0^{att} \\ 0, & \text{sinon} \end{cases} \quad (7)$$

où p^{att} est un gain linéaire, $q_{ij} = |\mathbf{q}_i - \mathbf{q}_j|$ est la distance entre les agents i et j et r_0^{att} est la portée d'attraction en dessous de laquelle les agents arrêtent de s'attirer pour éviter les collisions. Ajouter ce terme à la loi de contrôle doit être fait en accord avec le terme de répulsion afin d'éviter de créer un phénomène oscillatoire instable. Par conséquent, r_0^{att} doit être suffisamment grande pour que les agents aient une marge de déplacement entre l'attraction et la répulsion. Notre première proposition consiste donc en un contrôleur basé vitesse dénoté VAT (ajout de \mathbf{v}_i^{att} à l'eq. (6)) :

$$\tilde{\mathbf{v}}_i^d = \frac{\mathbf{v}_i}{|\mathbf{v}_i|} v^{flock} + \mathbf{v}_i^{rep} + \mathbf{v}_i^{att} + \mathbf{v}_i^{frict} + \sum_s \mathbf{v}_{is}^{wall, obst} \quad (8)$$

Les interactions avec les murs et les obstacles sont concaténées dans le même terme $\sum_s \mathbf{v}_{is}^{wall, obst}$ et l'équation (5) reste valide. Cette vitesse désirée englobe 12 paramètres

agissant sur chaque terme de cette loi et produisant des comportements différents. Nous verrons dans la section 5.2 comment ces paramètres sont optimisées afin d'atteindre un comportement optimal.

3.2 Modèle VOS : Vászrhelyi + Olfati-Saber

Le second modèle que nous proposons, dénoté VOS, est basé sur une combinaison du modèle d'Olfati-Saber [3] pour sa stabilité ainsi que certaines parties du modèle de Vászrhelyi [5] afin d'avoir l'auto propulsion et l'évitement d'obstacles. Pour cela, nous avons développé une structure de contrôle imbriquée alliant les deux contrôleurs :

$$\begin{cases} \mathbf{u}_i = \alpha \mathbf{f}_i^g + \beta \mathbf{f}_i^d & (9a) \\ \tilde{\mathbf{v}}_i^d = \frac{\mathbf{v}_i}{|\mathbf{v}_i|} v^{flock} + \sum_s \mathbf{v}_{is}^{wall,obst} + \int_{T_s} \mathbf{u}_i dt & (9b) \end{cases}$$

\mathbf{u}_i correspond à l'eq. (3) d'O-S sans la rétroaction de navigation \mathbf{f}_i^g . En effet ce terme *global* était introduit pour gérer la fragmentation. Or nous voulons développer un système de flocking totalement décentralisé (α et β sont des coefficients multiplicateurs qui compensent le délai introduit par l'intégration). $\tilde{\mathbf{v}}_i^d$ de l'eq. (2) d'O-S est étendue en introduisant le terme d'auto propulsion de Vászrhelyi, qui permet de gérer localement la fragmentation ($T_s = 1/f_s$ avec f_s la fréquence nominale du contrôleur). Nous ajoutons aussi un terme d'évitement des obstacles de l'environnement, semblable à celui de Vászrhelyi.

Les équations qui en découlent englobent 10 paramètres qui nécessiteront une optimisation (voir 5.2).

4 Communications et voisinage en présence d'obstacles

Comme mentionné dans l'Introduction, la plupart des simulations de flocking considère des communications idéales et ne prennent pas en compte l'impact de l'environnement sur la qualité des communications, provoquant un écart conséquent avec les conditions d'expérimentation réelles. Ainsi, nous proposons un modèle de communication fondé sur l'estimation de la perte radio, intégrant la traversée des obstacles. Cela nous permet de raffiner la définition du voisinage et de mieux estimer avec qui chaque agent interagit.

Généralement, les simulations de flocking utilisent un seuil de distance en dessous duquel la communication est possible et inversement (paramètre r de l'Eq.1). Avec ce type d'approche, les communications sont parfaites [3][15], ou plus réalistes avec la prise en compte de délais et pertes de paquets comme dans [5] [8]. Dans [4], les auteurs utilisent le BER (Bit Error Rate) combiné au SNR (Signal to Noise Ratio) afin de caractériser la qualité de communication et intègrent cela dans un index de performance exprimant une balance entre la distance inter-agent et la qualité de communication. Cependant, il n'y a aucun obstacle dans les environnements simulés par les auteurs. Dans [18], les auteurs développent un système hybride de propagation combinant une étape de prédiction et une étape d'apprentissage de l'affaiblissement de propagation. Ils utilisent l'outil

SPLAT!¹ afin de prédire les fluctuations du champ de propagation en fonction de la topologie d'une zone. Bien que cette méthode donne une bonne estimation des mesures radio, elle ne se soucie pas de l'adaptation du système multi-agent face à ces variations. Sur le même sujet, dans [7], l'auteur explique comment un comportement émergent, tel que le flocking, est impacté lorsque l'on utilise un modèle de propagation relativement réaliste et que l'on modifie les pertes dues aux effets tels que le shadowing, le SNR, le jamming et autres spécificités de la couche physique du moyen de communication. Cependant l'étude sur la dynamique du flocking mis en place est peu approfondie.

Dans la suite, nous introduisons une étape d'estimation de la propagation radio en utilisant le modèle *log distance path loss* (noté LDPL) à la fois dans le vide mais aussi au sein des obstacles et nous regardons comment des environnements hétérogènes affectent la qualité de communication et donc la stratégie de flocking.

4.1 Modèle de perte Log distance en environnements avec obstacles

Au lieu d'utiliser un modèle de communication basé uniquement sur la distance inter-agent, nous souhaitons modéliser de manière plus précise la manière dont évoluent les ondes radio avec l'environnement et par conséquent savoir comment sont affectées les interactions pour un voisinage donné. Pour se faire, nous considérons le modèle LDPL qui prédit la perte de puissance qu'un signal subit en fonction de l'environnement [19]. Il se formule de la façon suivante :

$$PL = P_{Tx_{dBm}} - P_{Rx_{dBm}}, \quad (10)$$

$$PL = \begin{cases} PL_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_g, & \text{if } d \geq d_0 \\ PL_0, & \text{otherwise} \end{cases} \quad (11)$$

où, PL est l'affaiblissement de propagation mesuré en décibels (dB), $P_{Tx_{dBm}}$ est la puissance transmise en dBm de l'émetteur, $P_{Rx_{dBm}}$ est la puissance reçue en dBm par le récepteur, PL_0 est l'affaiblissement de propagation à la distance de référence d_0 calculé en utilisant le modèle de Friis dans le vide, d est la distance entre le récepteur et l'émetteur, γ est le coefficient de l'affaiblissement de propagation qui dépend de l'environnement et X_g relate des variations de l'affaiblissement de propagation causées par les effets de "shadowing" et de chemins multiples. Cette variable suit une distribution normale centrée en zero et un écart type σ dB . Nous référerons à ce modèle sous le nom de **regular**. La courbe verte de la Figure 4 représente l'évolution de $P_{Rx_{dBm}}$ sur un chemin où il n'y a pas d'obstacles et $P_{Tx_{dBm}} = 20dBm$. Nous pouvons voir qu'il faut environ 300 mètres pour atteindre le seuil de puissance $P_{min} = -70dBm$ correspondant au niveau minimum de puissance requis pour établir une communication entre les agents.

Dans la formule précédente, l'environnement est modélisé par, entre autre, le paramètre γ et X_g . Afin de prendre en

1. Informations pour SPLAT! : <https://www.qsl.net/kd2bd/splat.html>

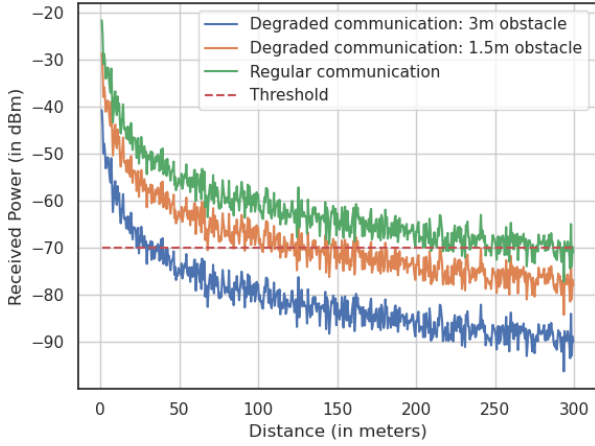


FIGURE 4 – Évolution de la puissance reçue en fonction de la distance inter-agent. La courbe verte représente un chemin sans obstacle, la courbe orange représente un chemin qui intègre un obstacle de 1.5 mètres et la courbe bleue intègre un obstacle de 3 mètres de long.

compte l’impact des différents obstacles sur la qualité de communication, nous ajoutons un modèle de dissipation au sein des obstacles créant des pertes majeures en terme de puissance reçue. L’affaiblissement de propagation résultante s’exprime de la façon suivante :

$$PL = \begin{cases} PL_0 + 10\gamma \log_{10}\left(\frac{d - d_{obst}}{d_0}\right) & \text{if } d - d_{obst} \geq d_0 \\ + X_g + PL_{obst}(d_{obst}), & \\ PL_0 + PL_{obst}(d_{obst}), & \text{sinon} \end{cases} \quad (12)$$

où, d_{obst} est la distance parcourue au sein d’un obstacle et PL_{obst} est l’affaiblissement de propagation en dB due à l’atténuation d’un obstacle. La formulation que nous utilisons est définie de la manière suivante :

$$PL_{obst}(d_{obst}) = 10\gamma_{obst} \log_{10}(d_{obst}) + K \quad (13)$$

où γ_{obst} est le coefficient d’affaiblissement de propagation pour un obstacle et K est une constante. Nous supposons que $d_{obst} \geq 1m$. Nous appellerons ce modèle : **degraded**. Les courbes orange et bleue dans la Figure 4 représentent l’évolution de $P_{Rx_{dBm}}$ sur un chemin où respectivement $d_{obst} = 1.5m$ et $d_{obst} = 3m$. Dans cette configuration, nous avons $\gamma_{obst} = 4$, $P_{Tx_{dBm}} = 20dBm$ et $K = 0$. Nous pouvons voir que ce modèle de communication a un impact très négatif sur la qualité de communication puisqu’il ne faut plus que 30 mètres pour perdre la communication entre deux agents qui ont un obstacle de 3 mètres entre eux.

4.2 Redéfinition du voisinage

En utilisant le nouveau modèle de propagation, nous pouvons redéfinir le voisinage d’interaction N_i par agent i :

$$N_i = \{j \in \llbracket 1, \dots, N \rrbracket; j \neq i : PL_{ij} \leq P_{threshold}\} \quad (14)$$

où PL_{ij} est l’affaiblissement de propagation (Eq. 12) mesuré entre j (émetteur) et i (récepteur) et $P_{threshold} (\neq P_{min})$ est le seuil de puissance en dessous duquel la communication n’est plus possible. Aussi nous introduisons un autre seuil N_{max} , ajustable, qui agit sur la cardinalité de N_i :

$$N_i = \begin{cases} N_i, & \text{if } |N_i| \leq N_{max} \\ N_i^{filtered}, & \text{sinon} \end{cases} \quad (15)$$

où $N_i^{filtered}$ correspond à un sous-ensemble N_i où les agents ayant la puissance reçue la plus faible sont récursivement retiré jusqu’à ce que $|N_i| = N_{max}$. Ce seuil est très important car il permet de limiter le voisinage et donc le coût de calcul au niveau de chaque agent, tout en conservant les voisins les mieux connectés.

Avec cette nouvelle définition du voisinage, basée sur un modèle de communication plus réaliste, nous pouvons étudier l’évolution d’une flotte d’agents dans des environnements fortement contraints par de nombreux obstacles.

5 Comparaison des modèles

5.1 Outil de simulation

Le simulateur que nous utilisons est une extension de l’outil² développé par Viragh et al. [8] illustré par la figure 5. C’est un simulateur multi-agent qui introduit des variables aléatoires pour modéliser les différents bruits ajoutés sur des variables telles que la position GPS ou encore les communications. Cela rajoute une complexité tant pour la résolution des équations différentielles stochastiques (EDS), en utilisant la méthode d’Euler-Maruyama [8], mais aussi pour l’optimisation des paramètres (voir Section 5.2). Nous avons décidé d’utiliser ce simulateur plutôt que les simulateurs plus courants tels que ARGoS ou GAZEBO pour deux raisons principales. La première vient du fait que le modèle initial de Vásárhelyi est déjà implémenté et nous offre donc une comparaison fidèle avec nos modèles. La deuxième est due au fait que nous utilisons ce simulateur principalement sur des serveurs de calculs pour réaliser l’optimisation des paramètres de flocking et donc sans visualisation. Ce simulateur est adapté à ce type d’optimisation, la visualisation n’étant utilisée que pour observer un comportement pour un jeu de paramètre donné.

Les hyper-paramètres utilisés pour une expérience tels que sa durée T ou encore le pas de calcul de la méthode d’Euler sont réglables. La distribution des obstacles est quant à elle définie par plusieurs paramètres tel que sa densité D qui se mesure de la manière suivante :

$$D = 100 \times \frac{\sum_s A_s}{A_{arena}}, \quad (16)$$

où A_s est la surface de l’obstacle s et A_{arena} est la surface de l’arène. La Figure 5(b) montre une vue 3D de l’environnement appelé “Forest” pour sa densité d’obstacles.

De nombreuses fonctionnalités ont été ajoutées à ce simulateur afin de répondre à nos besoins comme la détection

2. Accès au projet initial ici : <https://github.com/csviragh/robotsim>

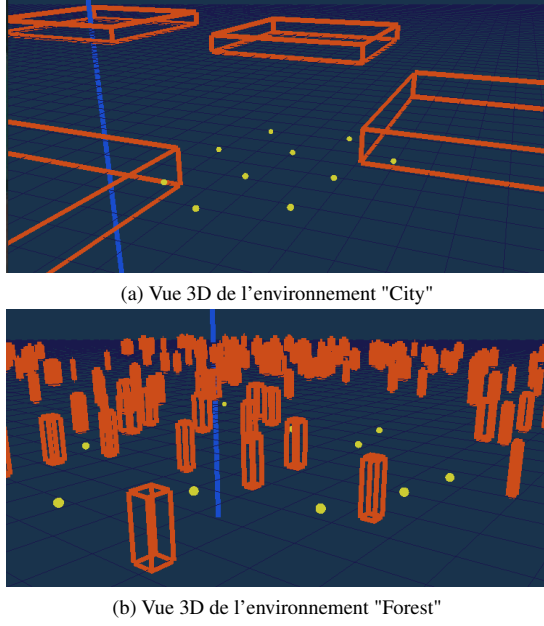


FIGURE 5 – Illustration des environnements tests avec le simulateur [8]

des clusters, en appliquant l'algorithme *Depth First Search* (DFS) sur le graphe des agents connectés, mais aussi la possibilité de changer à la volée le modèle de flocking ou encore le mode de communication (regular/degraded).

5.2 Optimisation des paramètres

Comme nous l'avons mentionné dans la Section 3, les modèles que nous étudions ont de nombreux paramètres qui doivent être optimisés afin d'obtenir le meilleur comportement possible. En s'inspirant du travail de [5], nous avons défini plusieurs métriques permettant d'évaluer les performances de nos modèles de flocking :

— **Velocity** :

$$\Psi_{vel} = \frac{1}{T} \frac{1}{N} \int_0^T \sum_{i=1}^N |\mathbf{v}_i(t)| dt, \quad (17)$$

qui caractérise la vitesse moyenne de la flotte.

— **Collision** (entre agents) :

$$\Psi_{col} = \frac{1}{T} \frac{1}{N(N-1)} \int_0^T \sum_{i=1}^N \sum_{j \neq i}^N \Theta(q_{ij}(t) - r^{coll}) dt, \quad (18)$$

mesure le ratio de collisions, avec Θ la fonction de Heaviside et r^{coll} la distance minimale autorisée entre deux agents.

— **Connected** :

$$\Psi_{clust} = \frac{1}{T} \int_0^T |\overline{O}(t)| dt, \quad (19)$$

$$O(t) = \{i \in [1, \dots, N] : |N_i(t)| = 0\}, \quad (20)$$

$\overline{O}(t)$ représente l'ensemble des agents connectés au temps t , cad. le complémentaire des agents isolés $O(t)$.

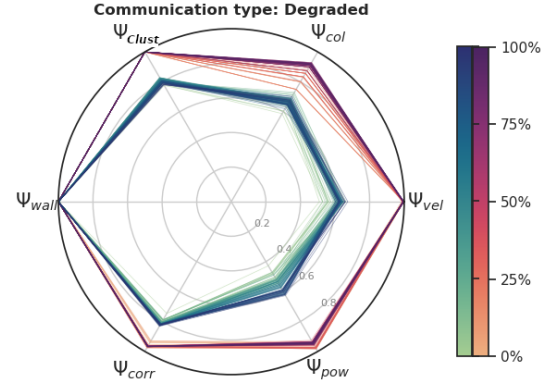


FIGURE 6 – Évolution des métriques durant l'optimisation en utilisant une communication dégradée dans l'environnement générique.

— **Wall** (collision) :

$$\Psi_{wall} = \frac{\int_0^T \sum_{i=1}^N \Theta(\tilde{q}_{is}(t)) \tilde{q}_{is}(t) dt}{\int_0^T \sum_{i=1}^N \Theta(\tilde{q}_{is}(t)) dt}, \quad (21)$$

où \tilde{q}_{is} prend des valeurs positives en dehors de l'arène et à l'intérieur d'un obstacle, et inversement. Ψ_{wall} caractérise la distance moyenne passée à l'intérieur des obstacles.

— **Cluster Correlation** :

$$\Psi_{corr} = \frac{1}{T} \frac{1}{N} \int_0^T \sum_{i=1}^N \frac{1}{|N_i| - 1} \sum_{j \in N_i} \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{|\mathbf{v}_i| |\mathbf{v}_j|} dt, \quad (22)$$

qui donne une mesure de l'alignement moyen par cluster.

— **Cluster Received Power** :

$$\Psi_{pow} = \frac{1}{T} \frac{1}{N} \int_0^T \sum_{i=1}^N \frac{1}{|N_i| - 1} \sum_{j \in N_i} P_{Rx_{ij}}(t) dt, \quad (23)$$

qui mesure la puissance reçue moyenne par cluster.

Ces métriques permettent de définir les conditions nécessaires pour obtenir un comportement optimal : $\Psi_{vel} \rightarrow v^{flock}$, $\Psi_{col} \rightarrow 0$, $\Psi_{clust} \rightarrow 0$, $\Psi_{wall} \rightarrow 0$, $\Psi_{corr} \rightarrow 1$, $\Psi_{pow} \geq P_{min}$. Les résultats obtenus sont passés dans des fonctions non linéaires pour normaliser les valeurs entre 0 et 1 (la fonction sigmoïd par exemple), 0 étant le pire cas et 1 le meilleur, et ce afin de définir un problème d'optimisation multi-objectif.

Dans ce papier nous utilisons un algorithme génétique multi-objectif appelé *NSGA-III* [20][21]. Ce dernier n'est cependant pas adapté aux problèmes stochastiques mais en évaluant la même solution plusieurs fois et en utilisant la variance résultante comme contrainte de l'optimisation nous avons pu palier à ce problème. Cela mène à des solutions plus robustes et par conséquent moins impactées par la caractéristique aléatoire d'une expérience.

La stratégie est ici d'optimiser les modèles VAT et VOS sur un environnement "générique", *i.e.* avec une densité

TABLE 1 – Métriques de VAT et VOS en env. GENERIC

	Vel	Col	Clust	Wall	Corr	Pow
VAT	0.999	0.99	0.94	1	0.86	0.202
VOS	0.612	0.88	0.92	1	0.784	0.26

TABLE 2 – Performances de VA dans [5]

	Vel	Col	Clust	Wall	Corr	Pow
VA	0.938	0.945	1	0.997	0.916	N/A

moyenne d’obstacles et avec un modèle de communication dégradé. Le nombre d’agents utilisé lors de l’optimisation et des simulations, dont les résultats sont donnés en Section 5.3, est toujours le même, $N = 10$. Nous utilisons ensuite ces versions optimisées avec différentes distributions d’obstacles afin de comparer leurs performances. L’environnement générique a $N_{obst} = 90$ obstacles uniformément distribués dans une arène de $1250m^2$. Chaque obstacle est représenté par un carré dont le côté L suit une distribution normale (Gaussienne) $L \sim \mathcal{N}(\mu, \sigma)$, centrée sur $\mu = 24$ mètres et un écart type $\sigma = 12$ mètres. La densité de cet environnement est $D = 3.3\%$.

Le processus d’optimisation est présenté en Fig.6 où l’on peut observer l’évolution de la moyenne des métriques sur le front de Pareto à chaque génération pour les deux modèles (VOS en bleu et VAT en rouge). L’échelle de couleur donne une information sur l’avancement de l’optimisation basé sur le nombre maximum de génération. Bien que VOS semble être bien moins performant que VAT, il existe une raison à cela. C’est notamment parce que son front de Pareto contient plus de solutions avec des variances plus grandes. La Table 1 regroupe les valeurs moyennes des métriques après 100 évaluations des solutions optimisées dans l’environnement *generic* avec une communication *degraded*. On se rend compte ici que les deux modèles ont des performances relativement similaires sauf pour la métrique *velocity* qui est plus basse pour le modèle VOS. Ces solutions et leurs performances vont être comparées en détail dans la prochaine sous-section.

5.3 Résultats et analyse

Notre but ici est de comparer les modèles VAT et VOS dans des environnements différents et avec des stratégies de communication différentes (regular et degraded). La comparaison est basée sur deux environnements différents (voir Fig.5) :

1. **Forest** : $N_{obst} = 500$; $L \sim \mathcal{N}(4, 2)$; $D = 0.7\%$.
2. **City** : $N_{obst} = 15$; $L \sim \mathcal{N}(200, 20)$; $D = 36.5\%$.

Nous allons également faire une comparaison avec le modèle de Vásárhelyi, noté VA, que nous simulerons en utilisant les paramètres optimaux fournis dans [5]. Les tables 3 et 4 regroupent les résultats de 100 évaluations stochastiques des modèles optimaux. Les valeurs moyennes des métriques normalisées entre 0 et 1 (1 étant le meilleur cas, 0

le pire) ainsi que leurs variances sont affichées. La dernière ligne correspond à l’inter-distance moyenne entre agents (en mètre). Afin d’être cohérent dans nos résultats, nous utilisons les mêmes hyper-paramètres à savoir, le temps d’expérience $T = 600s$, la précision de la méthode d’Euler $\Delta t = 0.01s$, le délai introduit dans les communications $t_d = 0.2s$ (représentant un délai d’accès au médium radio, le délai de transmission d’un paquet et le délai de propagation) et la fréquence de communication synchronisée avec la fréquence du GPS $f_{GPS} = 5Hz$. Une présentation de ces modèles est proposée dans la vidéo au lien³. La première observation importante ici est que le modèle VA a de moins bonnes performances que dans [5] (voir Table 2). Cela peut s’expliquer du fait que l’auteur se base sur de petites arènes dans ses essais, menant indirectement à un effet de cohésion (d’où l’absence de force d’attraction). Afin d’éviter ce comportement avantageux, nous utilisons une arène de 1250m de long. Par ailleurs, la distribution des obstacles n’est pas explicitement mentionnée dans [5] bien que celle-ci ait un fort impact sur les valeurs des métriques. Aussi, le fait de considérer notre nouveau modèle de propagation ajoute une nouvelle contrainte qui n’était pas considérée dans l’article initial. Les inconvénients de ne pas ajouter un terme d’attraction sont quant à eux encore plus visibles dans l’environnement *forest* où la métrique *cluster* est relativement basse, ce qui signifie que le phénomène de fragmentation a eu lieu à maintes reprises (Clust=0.637 dans Table 3). Il est également intéressant de remarquer que le fait d’ajouter un terme d’attraction ne crée pas de collisions supplémentaires bien que cela soit contre-intuitif. Ceci s’explique du fait que le terme d’attraction est actif uniquement au dessus d’une certaine valeur de distance inter-agent et avec une inertie résultante bornée pour les agents, ce qui prévient les risques d’oscillations et donc de collisions. Le fait que le modèle VAT soit aussi conçu et optimisé dans un contexte de communication **dégradé**, et donc bénéficie du *neighbour filtering*, participe à expliquer les écarts entre VAT et VA.

5.3.1 Maintien de la connectivité

L’un de nos objectifs dans cet article est d’implémenter un modèle capable d’éviter les fragmentations à répétition. Nous pouvons voir dans les deux tables que cet objectif est globalement atteint. En effet, pour VAT et VOS, la moyenne de la métrique *cluster* est haute, $0.868 \leq \text{Clust} \leq 0.985$. Cela démontre que même si quelques fragmentations se sont produites, les clusters résultant ont réussi à réparer les connexions détruites afin de former un seul et unique groupe. Malgré nos efforts, des fragmentations partielles restent possibles, notamment dans l’environnement *forest* où les nombreux obstacles créent de fréquentes interruptions de communication et donc d’interaction entre agents (voir vidéo).

5.3.2 Puissance reçue et collisions

La distance inter-agent est une composante résultante de toutes les interactions et est principalement modifiée lors

3. vidéo : https://team.inria.fr/chroma/files/2021/03/JFSMA_2021.mp4

TABLE 3 – FOREST

	Regular			Degraded		
	VA	VAT	VOS	VA	VAT	VOS
Vel	0.975	0.946	0.032	0.975	0.955	0.055
σ	0.010	0.03	0.052	0.008	0.023	0.063
Col	0.301	1	0.793	0.286	0.961	0.799
σ	0.429	0.099	0.382	0.429	0.172	0.380
Clust	0.637	0.98	0.923	0.508	0.938	0.868
σ	0.148	0.085	0.13	0.118	0.121	0.163
Wall	1	1	1	1	1	1
σ	0	0	0	0	0	0
Corr	0.133	0.712	0.54	0.128	0.698	0.531
σ	0.068	0.106	0.12	0.06	0.12	0.112
Pow	0.033	0.53	0.517	0	0.045	0.031
σ	0.049	0.11	0.074	0	0.079	0.066
Dist	132.3	37.9	49.9	132.5	42.11	54.37

de l'optimisation de la métrique *received power* ainsi que *collision*. Les agents doivent évoluer dans un intervalle de distance avec les autres qui est pré-défini : pas trop près afin d'éviter les collisions sans pour autant aller trop loin et perdre la connexion avec le groupe. Il est important de comprendre que dans ces tables, une valeur faible de *received power* ne veut pas dire que la connexion est perdue mais qu'elle est faible et donc sujette à des pertes de paquets et autres erreurs. Par exemple dans la Table 4, nous avons $Pow = 0.573$ pour le modèle VOS avec une communication *degraded* mais cela est équivalent à une valeur de $P_R = -59.4dBm$, ce qui signifie qu'en moyenne, la communication était possible. En comparaison, dans la Table 3, pour le modèle VAT avec une communication *degraded*, $Pow = 0.045 \simeq -71.7dBm$ ($-70dBm$ étant la valeur seuil en deça de laquelle la communication n'est plus possible). Cela confirme le fait que, même si la communication est en moyenne impossible, dès qu'ils le peuvent, les agents arrivent à garder une formation de flotte (la métrique de *cluster* étant haute).

5.3.3 Overfitting de l'environnement

Comme mentionné plus haut, les modèles que nous simulons ne sont pas optimisés sur les environnements que nous utilisons pour la comparaison. Par conséquent, il y a un risque d'overfitting sur une distribution d'obstacle donnée. Nous pouvons observer cela pour le modèle VOS dans la Table 3 où la moyenne de la métrique *velocity* est basse, $0.032 \leq Vel \leq 0.055$, ce qui équivaut à environ 60% de la vitesse désirée v^{flock} . Bien que dans la Table 1, $Vel = 0.612$, il y a approximativement 5 fois moins d'obstacles dans cet environnement *generic* où l'optimisation est effectuée. Dans *forest*, il y a énormément d'obstacles proches les uns des autres entraînant à un "effet de cage" et empêchant la flotte de voler à la vitesse désirée. Cela crée aussi de fortes oscillations $0.531 \leq Corr \leq 0.54$ et quelques collisions $0.793 \leq Col \leq 0.799$ (voir vidéo). Le modèle VOS n'est par conséquent pas robuste à la variation

TABLE 4 – CITY

	Regular			Degraded		
	VA	VAT	VOS	VA	VAT	VOS
Vel	0.999	1	0.484	0.999	1	0.501
σ	1.1e-5	0	0.095	4.8e-5	0	0.101
Col	0.71	0.999	0.918	0.654	1	0.86
σ	0.442	0.09	0.264	0.47	0	0.33
Clust	0.671	0.985	0.939	0.54	0.915	0.925
σ	0.18	0.065	0.142	0.197	0.17	0.165
Wall	1	1	1	1	1	1
σ	0	0	0	0	0	0
Corr	0.605	0.912	0.886	0.73	0.945	0.894
σ	0.165	0.067	0.082	0.112	0.04	0.073
Pow	0.512	0.797	0.684	0.335	0.777	0.573
σ	0.205	0.092	0.073	0.286	0.16	0.204
Dist	82.19	29.4	44.3	86.76	33.8	45.55

des distributions d'obstacles.

5.3.4 Communications Regular/Degraded

Les effets résultants de la communication *degraded* peuvent s'observer dans les deux tables. De plus, la dernière ligne qui concerne la distance inter-agent montre que ce type de communication va provoquer l'augmentation de la distance moyenne entre les agents, ce qui cause la plupart du temps de la fragmentation. Cependant, le modèle VAT semble être très résilient et robuste aux pertes introduites par le mode *degraded*, ce qui en fait un modèle particulièrement performant.

6 CONCLUSION

Dans cet article, nous avons proposé deux nouveaux modèles de flocking basés sur l'état de l'art (VAT Vászárhelyi + Attraction et VOS Vászárhelyi + Olfati-Saber) dans le but d'être robuste à des communications dégradées par la présence d'obstacles. Nous avons comparé ces modèles en simulation en proposant un modèle de perte (LDPL) qui prend en compte la propagation radio au sein des obstacles. Cela nous a permis d'analyser comment les modèles de flocking sont affectés ou au contraire robustes à des communications dégradées par les obstacles de l'environnement. Parmi les solutions proposées, et après leur optimisation, nous pouvons affirmer que le modèle VAT donne des résultats plus encourageants que VOS ainsi que le modèle initial de Vászárhelyi.

Bien que l'optimisation ait été réalisée ici uniquement sur l'environnement "generic", il pourrait être intéressant de systématiquement optimiser les modèles dans chacun des environnements utilisés. La suite de notre travail consistera à implémenter des stratégies de flocking adaptatives capables de se comporter de manière optimale dans des environnements hétérogènes avec différents types de communication. Nous prévoyons aussi de mesurer les performances des modèles avec des drones quadrirotor lors d'expérimentation réelles.

Références

- [1] Noury Bouraqadi and Arnaud Doniec. Flocking-based multi-robot exploration. In *4th National Conference on "Control Architectures of Robots"*, Toulouse, France, 2009.
- [2] Craig W. Reynolds. Flocks, herds and schools : A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4) :25–34, 1987.
- [3] Reza Olfati-Saber. Flocking for multi-agent dynamic systems : algorithms and theory. *IEEE Trans. Automat. Contr.*, 51(3) :401–420, 2006.
- [4] Heng Li, Jun Peng, Weirong Liu, Jing Wang, Jiangang Liu, and Zhiwu Huang. Flocking control for multi-agent systems with communication optimization. pages 2056–2061, 06 2013.
- [5] G Vászárhelyi, C Virágh, G Somorjai, T Nepusz, AE Eiben, and T Vicsek. Optimized flocking of autonomous drones in confined environments. *Science Robotics*, 3(20), 2018.
- [6] G. Wen, Z. Duan, H. Su, G. Chen, and W. Yu. A connectivity-preserving flocking algorithm for multi-agent dynamical systems with bounded potential function. *IET Control Theory Applications*, 6(6) :813–821, 2012.
- [7] B. Fraser, R. Hunjet, and C. Szabo. Simulating the effect of degraded wireless communications on emergent behavior. In *2017 Winter Simulation Conference (WSC)*, pages 4081–4092, 2017.
- [8] Csaba Virágh, Gábor Vászárhelyi, Norbert Tarcai, Tamás Szörényi, Gergo Somorjai, Tamás Nepusz, and Tamás Vicsek. Flocking algorithm for autonomous flying robots. *Bioinspiration & Biomimetics*, 9(2) :025012, may 2014.
- [9] G. W. Gamage, G. K. I. Mann, and R. G. Gosine. Leader follower based formation control strategies for nonholonomic mobile robots : Design, implementation and experimental validation. In *Proceedings of the 2010 American Control Conference*, pages 224–229, 2010.
- [10] S. S. Ge and Cheng-Heng Fua. Queues and artificial potential trenches for multirobot formations. *IEEE Transactions on Robotics*, 21(4) :646–656, 2005.
- [11] T. Balch and R. C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6) :926–939, 1998.
- [12] N. H. M. Li and H. H. T. Liu. Formation uav flight control using virtual structure and motion synchronization. In *2008 American Control Conference*, pages 1782–1787, 2008.
- [13] Chao Wang, Jian Wang, Yuan Shen, and Xudong Zhang. Autonomous navigation of uavs in large-scale complex environments : A deep reinforcement learning approach. *Transactions on Vehicular Technology*, 68(3) :2124–2136, 2018.
- [14] Ronny Lim and W. Sheng. Hybrid system of reinforcement learning and flocking control in multi-robot domain. 2016.
- [15] H. M. La and W. Sheng. Flocking control of a mobile sensor network to track and observe a moving target. In *2009 IEEE International Conference on Robotics and Automation*, pages 3129–3134, 2009.
- [16] Guanghui Wen, Zhisheng Duan, Housheng Su, Guanrong Chen, and Wenwu Yu. A connectivity-preserving flocking algorithm for nonlinear multi-agent systems with bounded potential function. *Proceedings of the 30th Chinese Control Conference, CCC 2011*, 01 2011.
- [17] Tamás Vicsek, András Czirok, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particules. *Physical review letters*, 75(6) :1226, 1995.
- [18] Eric Frew Spencer Watza, Ramya Kanlapuli. Hybrid rf propagation model using itm and gaussian processes for communication-aware planning. In *RSS 2017 RCW Workshop*, 2017.
- [19] Theodore Rappaport. *Wireless Communications : Principles and Practice*. Prentice Hall PTR, USA, 2nd edition, 2001.
- [20] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i : Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4) :577–601, 2014.
- [21] J. Blank and K. Deb. Pymoo : Multi-objective optimization in python. *IEEE Access*, 8 :89497–89509, 2020.

Modèle générique inclusif de système ambiant dédié à l'assistance basée sur le paradigme multi-agent

Résumé

L'intelligence ambiante est un paradigme qui vise le déploiement d'environnements intelligents et adaptatifs destinés à accompagner les activités des utilisateurs. Basés sur ce paradigme, les Systèmes ambiants d'assistance permettent de prévenir et d'améliorer la santé et les conditions de vie de ses utilisateurs, qui sont souvent des personnes touchées par un trouble moteur ou cognitif. Cependant, dans la littérature, les solutions proposées adoptent une approche ad hoc et centralisée, ce qui limite, entre autres, leur réutilisabilité, leur interopérabilité ainsi que leur robustesse. Nous proposons un modèle générique pour la conception de systèmes multi-agents ambiants dédié à l'assistance à la personne. Du fait de sa réutilisabilité, ce modèle facilite la mise en opération de tels systèmes, leur interopérabilité, et favorise l'échange des données. L'architecture résultante joue le rôle de « liant » entre les agents logiciels, l'environnement intelligent et les utilisateurs (aidés ou aidants), qui sont placés au centre du modèle.

Mots-clés : *Système multi-agents ambiant, Système d'assistance à l'autonomie, Modèle générique*

Abstract

Ambient intelligence is a paradigm that aims at deploying smart and adaptive

environments that support users' activities. Based on this paradigm, Ambient Assisted Living Systems help prevent and improve the health and living conditions of its users, who are often people affected by motor or cognitive disorder. However, in the literature, the proposed solutions adopt an ad hoc approach, which limits, among other things, their reusability and interoperability. We propose a generic model for the design of ambient multi-agent systems dedicated to personal assistance. Because of its reusability, this model facilitates the implementation of such systems, their interoperability, and data exchange. The resulting architecture acts as a "glue" between the software agents, the intelligent environment and the users (helped or caregivers), who are placed at the center of the model.

Keywords: *Ambient Multi-Agents System, Ambient Assisted Living System, Generic Model*

1 Introduction

Dans l'ère de l'informatique ubiquitaire, la miniaturisation des composants électroniques permet l'intégration des technologies informatiques dans les objets de notre quotidien. Cette ère, dans laquelle nous nous trouvons encore, nous amènera à mettre de côté les périphériques qui ont permis des

interactions explicites avec l'ordinateur (souris, claviers ou écrans, etc.) au profit d'interactions plus « invisibles » avec le système par le biais d'objets connectés [1].

Ces appareils, intégrés dans l'environnement du quotidien, peuvent servir à délivrer des services adaptatifs et personnalisés, ayant pour but de renforcer et supporter les activités humaines. Ce paradigme visant à instrumenter nos environnements quotidiens pour améliorer la qualité de vie de ses utilisateurs correspond à l'Intelligence Ambiante [2], [3]. Parmi les applications possibles de ce paradigme, les systèmes ambiants d'assistance (ou systèmes d'assistance à l'autonomie à domicile) permettent de « prévenir, guérir et améliorer le bien-être et la santé » [2] de ses utilisateurs. Souvent, les systèmes ambiants d'assistance vont accompagner les personnes âgées en prolongeant leur autonomie. Cela peut passer par l'accompagnement sur la prise des médicaments, les rappels de mémoire ou encore la détection de situation à risque.

Pour concevoir et mettre en opération de tels systèmes, le paradigme multi-agent est tout à fait adapté [4]. Car le caractère autonome des prises de décisions et des communications, et l'étroite relation entre l'agent et son environnement sont en phase avec la nature même des systèmes multi-agents [5].

Pourtant, les approches choisies pour mettre en place ces systèmes ambiants d'assistance sont faites sur mesure (approche ad hoc) et donc limitées à un usage précis [6]. En outre, les solutions proposées sont souvent centralisées et ne permettent pas un fonctionnement distribué du système, ce qui limite la flexibilité du système et n'est pas idéal pour la protection de la vie privée. Enfin, malgré la bonne volonté mise en œuvre pour concevoir des systèmes ambiants d'assistance, il apparaît que la prise en compte du facteur humain dans le système n'est pas suffisante [2].

L'apport que nous proposons consiste en un modèle d'architecture générique pour la

conception de systèmes ambiants d'assistance, basé sur le paradigme multi-agents. Dans les sections suivantes, nous débuterons par la description des critères importants pour la conception de tels systèmes et nous verrons les limites des études actuelles par rapport à ces critères. Nous aborderons ensuite la présentation du modèle à travers plusieurs prismes : la modélisation des agents, celle de l'environnement, celui du domaine de l'assistance et finalement la place du facteur humain au sein du système.

2 Contexte

2.1 Critères de conception

Les systèmes ambiants d'assistance se destinent à faciliter la vie quotidienne de ses utilisateurs, notamment en améliorant leur confort et en facilitant leur autonomie dans les tâches du quotidien. À partir des travaux de [6], [7], [2], [8], nous avons dressé un certain nombre de critères qu'il est nécessaire de prendre en compte pour la conception de ce type de système :

- **Adaptation** : le système doit évoluer dynamiquement en fonction des utilisateurs, mais aussi des ressources disponibles ;
- **Décentralisation** : le fonctionnement du système doit faire en sorte que chaque élément soit indépendant. Cette décentralisation permet de renforcer d'autres caractéristiques, dont la flexibilité et la protection de la vie privée ;
- **Personnalisation** : en complétant le critère d'adaptation, le système doit s'adapter en offrant les meilleurs services possible en fonction des préférences et des besoins des utilisateurs ;
- **Transparence** : le fonctionnement du système doit être suffisamment clair

pour que les utilisateurs puissent, consciemment, décider des limites et du degré de soutien que le système leur apporte ;

- **Flexibilité** : la conception du système doit être souple de telle sorte que changer la problématique à résoudre ou le type d'environnement de déploiement ne nécessite pas de tout reconcevoir ;
- **Efficience** : au vu de l'état écologique actuel, nous pensons qu'il est nécessaire de considérer cette dimension lors de la conception du système ;
- **Approche centrée sur l'utilisateur** : le facteur humain est un point central dans ce type de système. Il est donc important d'inclure l'humain, en adoptant une approche de conception de système socio-technique ambiant. En plus des personnes supportées par le système, il est nécessaire d'entrer en ligne de compte le réseau d'aidants associé à ces personnes ;

2.2 Systèmes et modèles existants

Dans cette section, nous analysons des systèmes ambiants d'assistance proches de ce que nous proposons, à travers le prisme des critères énoncés précédemment.

OutCare : Ce projet [9] propose un système d'assistance ambiant pour les personnes touchées par la maladie d'Alzheimer, notamment dans leurs trajets du quotidien. À partir des informations de localisation obtenues via un téléphone portable, des agents logiciels vont fournir des services de supervision d'itinéraires et de rappels mémoires. Les données sont récupérées sur un serveur, qui peut les délivrer aux personnes aidantes.

KopAI : propose des services similaires, en proposant des mécanismes de détections de situations dangereuses ou déviantes au sein du

domicile, et en proposant des rappels de rendez-vous [10]. Les données et les interactions se font à partir d'appareils mobiles, qui renvoient les données sur un serveur central de traitement.

Rosetta : Dans la même lignée, ce projet européen propose un système ambiant d'assistance dédié aux tâches quotidiennes des personnes atteintes de démences [11], [12]. L'environnement de ses utilisateurs est augmenté des capteurs, qui vont permettre de proposer un certain nombre de services, dont de l'assistance à la navigation, la détection de comportements anormaux et la détection de situations d'urgences. Les données transitent via un serveur et les interactions avec le système se font par le biais d'une tablette. Notons qu'un point d'honneur a été mis pour faire participer des utilisateurs réels afin d'améliorer la conception du système.

Les projets qui ont été présentés sont destinés à accompagner les utilisateurs dans leurs tâches du quotidien. Mais ces derniers ne possèdent pas tous les critères de conception énumérés précédemment. Notamment, ces projets (comme la plupart des projets de ce domaine) font l'impasse sur la décentralisation du système. En outre, les solutions proposées sont souvent dépendantes d'un appareil spécifique, ce qui limite la flexibilité du système ainsi que son caractère adaptatif. De plus, ces solutions ne sont pas construites sur un socle générique, ce qui ne facilite pas l'interopérabilité et la réutilisabilité pour un cadre différent de celui abordé initialement.

Pour pallier ce problème, certains chercheurs ont proposé des modèles ou des architectures pour la mise en opération de systèmes ambiants d'assistance.

SOPRANO : est une plateforme qui propose une architecture de système ambiant dépendant d'un middleware qui relie les composants du système sur la base d'une ontologie [13].

OpenAAL : est basé sur les travaux du projet de « Soprano » et offre une architecture qui facilite l'implémentation de services personnalisés et adaptés au contexte [14].

Comme le soulignent les auteurs de l'étude [6], la plupart des projets de systèmes ambiants d'assistance (dont ceux cités précédemment) manquent de standardisation, ce qui bride la flexibilité et l'interopérabilité. En outre, il est possible de remarquer un manque du côté de la formalisation des données, ce qui limite encore une fois l'interopérabilité et la réutilisabilité des données entre les systèmes. Enfin, globalement, plusieurs études [2], [15] s'accordent à dire que la prise en compte du facteur humain reste à améliorer.

3 Proposition

3.1 Présentation du modèle

Le modèle que nous proposons vise à formaliser les différents éléments d'un système d'assistance ambiant et leurs relations. Ces systèmes ambiants d'assistance sont des systèmes complexes et fortement hétérogènes. En effet, notre système doit jouer le rôle de « liant » entre les différentes entités composantes, comme visible sur la figure 1 : d'une part, les *utilisateurs* qui peuvent être des personnes aidées ou aidantes, et qui possèdent leur profil et leurs préférences propres (notamment les limites physiques ou psychiques que le système doit prendre en compte) ; d'autre part, les *objets connectés*, qui instrumentent l'environnement physique des utilisateurs et présentent différentes fonctionnalités et spécificités ; et enfin les *agents*, qui possèdent des objectifs propres et doivent offrir des services d'assistance, malgré leur hétérogénéité en termes de perceptions (à quels appareils et/ou agents ils ont accès) et de buts.

Comme énoncé dans la section 2, nous voulons répondre au manque de

standardisation, tant en termes de modèle générique que de l'architecture et des données. De plus, nous nous sommes évertués à mettre un point d'honneur à l'inclusion du facteur humain dans le système et ainsi dépasser la vision du système purement informatique pour l'envisager comme un système socio-informatique.

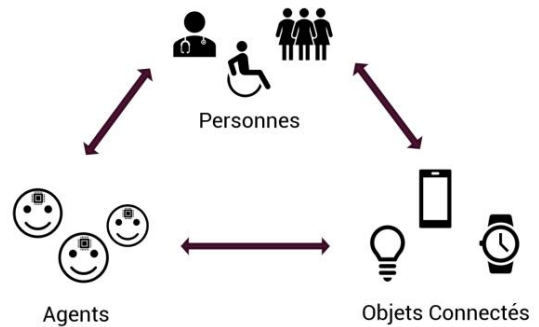


FIG. 1 – Schématisation des différentes entités en interactions dans un système ambiant d'assistance

Ce modèle a été formalisé sous la forme d'une ontologie, décrite dans le langage standard OWL. Cette ontologie présente l'avantage d'être exploitable à plusieurs niveaux. Premièrement, la formalisation du modèle selon une ontologie en OWL permet de valider son appui d'un point de vue sémantique et logique. Deuxièmement, elle permet de structurer les connaissances portées par les agents, ainsi que les communications entre ces derniers. Troisièmement, les données issues du système bénéficient de cet appui sémantique.

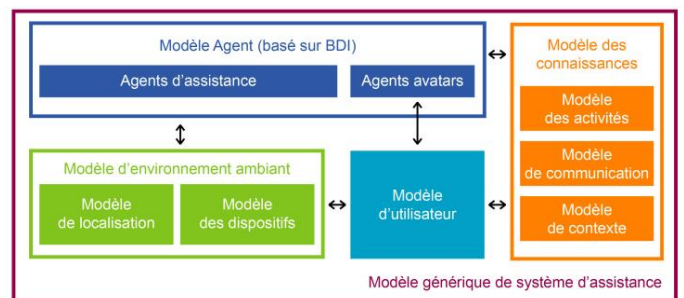


FIG. 2 – Schéma structurel du modèle générique présenté

Dans les sections suivantes, nous aborderons les différentes facettes du modèle présenté sur la figure 2.

Nous commencerons par le modèle agent, suivi de la modélisation de l'environnement. Nous aborderons ensuite les facettes liées au domaine de l'assistance et de la place de l'humain au sein du système.

3.2 Modèle Agent

Dans notre contexte, les agents du système sont des entités logicielles autonomes dont l'objectif est de délivrer des services pour assister directement ou indirectement les utilisateurs. Puisque nous nous plaçons dans un contexte ambiant, les agents peuvent interagir avec l'environnement physique réel, par le biais des capteurs et effecteurs. En outre, ces agents peuvent coopérer et communiquer au sein d'espace d'échanges sociaux, appelés groupes. Comme l'ont proposé les auteurs de [16], ces agents joueront des rôles spécifiques, destinés à délivrer les services d'assistance. Ce tout forme un système cyber-physique où agents, utilisateurs et objets connectés sont en interaction.

Afin de délivrer des services d'assistance qui soient personnalisés et adaptés, les agents devront accomplir leurs objectifs en fonction de leurs connaissances sur l'état du système. Cela englobe leurs connaissances sur l'état de l'environnement physique (par exemple : faut-il allumer la lumière si la personne à assister est à l'extérieur et qu'il fait jour), mais également sur les utilisateurs impliqués (par exemple : si la personne assistée a un déficit visuel, il faudrait que la luminosité de son environnement soit suffisamment élevée). Ainsi, les agents décrits sont cognitifs et notre modélisation des agents s'appuie sur le modèle de référence BDI [17].

Puisque nous proposons la modélisation d'agents cognitifs, la représentation des buts de ces agents doit être formellement décrite. La modélisation des buts que nous proposons s'établit selon les travaux de [18], [19] et

permet d'en déterminer 4 types, comme illustrés sur la figure 3.

L'accomplissement de ces buts est dépendant de l'opportunité d'effectuer certaines actions, ou de l'accomplissement d'autres buts par rapport à la véracité d'un contexte (notion que nous traiterons dans la section 3.4).

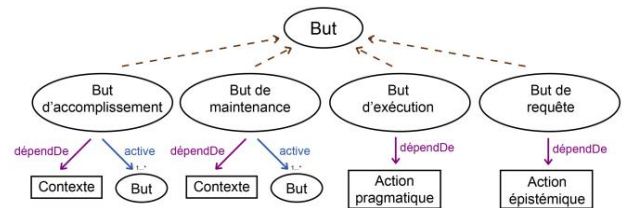


FIG. 3 – Schéma de catégorisation des buts

Par exemple, pour un agent dédié au déclenchement d'alerte suite à la chute d'un utilisateur, celui-ci possède un but d'accomplissement (voir la figure 3) qui déclenchera l'activation d'autres sous-buts dès que le contexte à surveiller (ici, le fait qu'un utilisateur ait chuté) sera vérifié.

Parmi les différentes actions pouvant être effectuées par les agents, certaines dépendent de la communication entre ces derniers. Pour cette communication, notre modélisation permet d'employer le langage FIPA ACL [20], [21]. En outre, ces communications se feront sur la base de l'ontologie du modèle proposée pour structurer les connaissances échangées. Nous apporterons quelques précisions sur les formalités de communication entre les agents dans la section 3.5.

3.3 Modèle d'environnement

Comme énoncé dans la section précédente, notre modèle permet la représentation de systèmes cyber-physiques où des agents interconnectés peuvent coopérer et interagir avec l'environnement physique réel, dans le but d'offrir des services d'assistance.

Dans le cadre des interactions avec l'environnement physique, les agents peuvent

compter sur des capteurs et des effecteurs. Ces capteurs et effecteurs peuvent être sémantiquement reliés au paramètre physique duquel ils dépendent, afin de permettre aux agents de détecter de quels appareils ils vont se servir (pour éclairer, un agent va pouvoir détecter et actionner un appareil qui a été typé comme étant une lumière), mais également de spécifier les éventuelles données exploitées (si un agent doit détecter la luminosité ambiante, il va pouvoir déterminer que les données issues du capteur sont exprimées en lux).

Il est important de préciser qu'un appareil peut être composite et en ce sens, il encapsule d'autres appareils que lui-même. Cela est particulièrement utile pour décrire des appareils comme les smartphones qui, en plus de capacité de calcul, comprennent plusieurs effecteurs et capteurs. Ce mécanisme de décomposition et recomposition, issu de [22], apporte comme avantage de rendre le système plus efficient, en permettant de mutualiser les appareils, et donc de réduire le nombre de dispositifs nécessaires au fonctionnement du système.

3.4 Modèle de l'assistance

Pour délivrer des services d'assistance aux utilisateurs, il faut que les objectifs des agents correspondent à cela. À partir des travaux de [2], [7], le modèle permet de représenter 4 types d'assistance, dont découlent 4 types de rôles :

- *Accompagnement Cognitif* : L'agent va pouvoir adopter certains comportements (par exemple soumettre des rappels ou guider implicitement l'utilisateur) pour permettre à l'utilisateur de prendre les meilleures décisions possible ou pour prolonger son autonomie.
- *Détection des situations dangereuses* : L'agent observe l'état d'une personne si cet état est jugé dangereux, des comportements appropriés doivent

être exécutés (lancement d'alerte ou autre...). Cela correspond à décrire un but d'accomplissement, dont le contexte à surveiller est celui d'une chute, et qui va déclencher des sous-buts pour réagir à cette situation dangereuse.

- *Surveillance de la santé* : Pour délivrer ce service, l'agent doit récupérer les données des dispositifs liés au paramètre de santé à observer (ce qui correspond à des buts de requêtes). Grâce à ces données, l'agent pourra inférer sur ces connaissances pour déterminer et qualifier un état de santé de la personne. Cet état pourra ensuite être communiqué à des utilisateurs (le médecin par exemple) et d'autres agents afin que ces derniers puissent prendre en compte cette nouvelle information.
- *Assistance des activités* : Lorsqu'une activité effectuée par un utilisateur est à assister, l'agent doit déclencher les comportements appropriés tant que l'accomplissement de cette dernière n'est pas vérifié.

Afin de fournir cette assistance de manière personnalisée, le système doit pouvoir enregistrer les changements et dynamiques du système, c'est-à-dire les différentes activités. Ces activités correspondent aux occurrences des différentes actions effectuées au sein du système. Le modèle d'activité dont nous nous servons pour représenter se base sur les travaux de [23]. Ainsi, une activité est composée de : un **sujet** (utilisateur ou agent qui accomplit l'action), un **objet** (l'action entreprise), et un **instrument** (utilisateur ou objet qui permet d'accomplir l'action). Le tout s'inscrit d'un **contexte** associé, comme le montre la figure 4.

En effet, le contexte est un élément important dans les systèmes ambiants d'assistance. C'est la prise en compte de ce contexte qui permet de personnaliser les services à

délivrer. Pour modéliser ce contexte, nous nous appuyons sur plusieurs études dont les travaux de [24]–[26]. Pour le modèle que nous présentons, le contexte correspond à un agrégat d'informations permettant de qualifier une activité ou une situation, en répondant aux questions qui, quoi, où et quand.

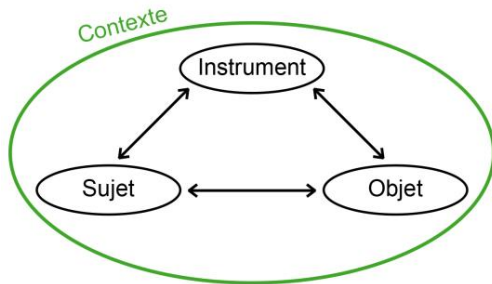


FIG. 4 – Modélisation des activités dans le système, basé sur les travaux de [23]

Pour donner un exemple d'utilisation des concepts qui viennent d'être discutés, si un agent a reçu l'information que Bob a préparé un repas chez lui, cela correspondrait à : l'enregistrement d'une activité, comme montré dans la figure 5, ayant pour *sujet* : Bob, pour *objet* : cuisiner, pour *instrument* : cuisinière_de_bob. Enfin, le contexte associé à cette activité serait composé d'une information sur le lieu : maison_de_bob ; et sur l'instant où cette activité a été effectuée : 12h00.

Sujet	Instrument	Objet	Contexte
Bob	cuisinière_de_bob	Cuisiner	+ maison_de_Bob + 12h00

FIG. 5 – Tableau de données de l'activité décrite dans l'exemple de Bob

3.5 Place de l'humain dans le système

Comme nous l'avons annoncé précédemment, notre approche se veut inclusive de l'utilisateur. Ce dernier possède sa représentation au sein du système. En effet, chaque utilisateur est lié à un profil et à des préférences. Ces informations comportent

également des précisions sur l'état de santé de la personne, afin que le système puisse s'adapter à ces différentes informations. Comme indiqué dans la section 3.4, certaines activités des utilisateurs peuvent être enregistrées, ce qui peut permettre aux agents d'inférer sur l'état des utilisateurs ou de l'environnement.

En outre, il est nécessaire d'ajouter une interface entre le système et l'utilisateur, qui soit garante de l'intégration du facteur humain. Cette interface doit se montrer adaptable pour interagir avec l'utilisateur, en prenant en compte les connaissances sur la situation de la personne. De plus, elle doit également communiquer avec les services proposés pour leur fournir les informations de l'utilisateur nécessaires à leur fonctionnement. Pour aller plus loin dans la prise en compte du facteur humain, cette interface est responsable du respect de la vie privée, en régulant à qui et quelles informations sont partagées. Enfin, cette interface ne doit pas se substituer à la place de l'humain, mais plutôt favoriser les liens sociaux.

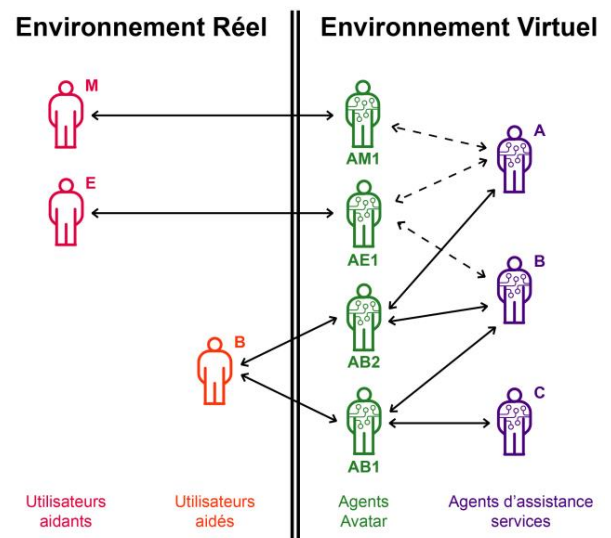


FIG. 6 – Représentation des utilisateurs en interaction avec les agents par le biais d'agents avatar

C'est pourquoi nous proposons que cette interface corresponde à des agents-avatars (AA). Ces AA servent de médiateurs entre le système et les utilisateurs, comme le montre la figure 6. Ils portent les informations du profil et de l'état de l'utilisateur (selon ses préférences). Dans un sens, ils devront communiquer les demandes de l'utilisateur aux agents dont le service est sollicité. Dans l'autre sens, lorsqu'un autre agent souhaite communiquer avec l'utilisateur, l'AA devra se charger d'établir, au moyen des appareils à disposition, la communication avec ce dernier.

Enfin, dans certaines tâches où une implication émotionnelle est jugée nécessaire (par exemple, pour demander à l'utilisateur de se nourrir alors qu'il n'a pas pris en compte les alertes affichées par le système) les agents doivent prendre en compte ce critère pour déléguer cette tâche à une personne avec qui l'utilisateur a de bonnes relations.

En outre, un utilisateur peut désirer partager certaines informations avec certains groupes. À la manière dont nous possédons plusieurs cercles sociaux (cercle d'amis, cercle professionnel, etc.), et adaptons notre posture et ce que nous partageons avec chaque cercle, les agents peuvent former des espaces de communications, comme le montre la figure 6. Notons que l'accès à ces espaces de communication devrait être restreint par un système d'authentification, ou approuvé par la personne assistée.

Ainsi pour reprendre l'exemple des figures 6 et 7, considérons la personne aidée Bob (B). Bob est assisté par un médecin (M) et par une proche : Edith (E).

L'AA de Bob numéro 2 (AB2) représente Bob dans un réseau de communication formel (EC1). Dans cet espace, il est possible d'imaginer que les échanges concernent uniquement les informations médicales de Bob. Pour communiquer sur des données plus personnelles, l'espace de communication (EC2) met en relation l'agent d'assistance (B), l'AA de Edith (AE1) et les deux AA de

Bob (AB1) et (AB2).

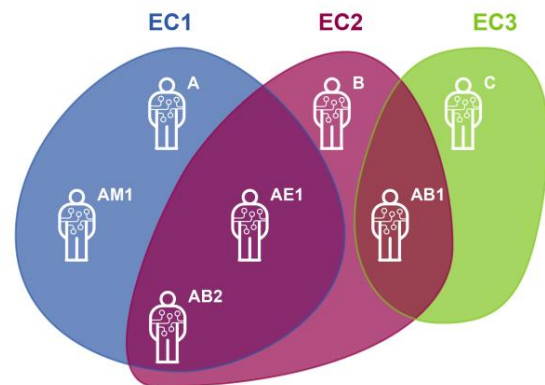


FIG. 7 – Représentation des espaces de communications formés par les échanges affichés sur la figure 6

Dans cet espace (EC2), les échanges peuvent concerner des informations relatives aux activités quotidiennes de John, que ce dernier ne veut partager qu'avec ses proches. Enfin, l'espace (EC3) est réservé aux communications de John avec des agents chargés de délivrer des services personnels, par exemple ceux qui concernent les services de son domicile.

Ces agents avatars et l'organisation de ces espaces de communication permettent de laisser à l'utilisateur le choix de définir ce qu'il partage et avec qui. Ce critère de respect de la vie privée est crucial pour que l'utilisateur puisse avoir confiance et prendre sa place vis-à-vis du système [15], en étant partie prenante. C'est une approche qui demeure cohérente pour la gestion des données personnelles, au vu des questions sociétales actuelles sur la question et de l'existence de travaux similaires et actuels qui sont dans la même visée, telle que le concept de « Pod » présenté dans [27].

3.6 Apports du modèle

Le modèle proposé est un modèle générique qui permet la conception d'architecture de systèmes ambiants dédiés à l'assistance. Ce modèle a été formalisé selon une ontologie, décrite dans le langage OWL, ce qui apporte

plusieurs avantages : le modèle s'appuie sur un socle sémantique et logique de référence ; les connaissances et les communications suivent la même structure uniforme décrite dans le modèle ; les données en sorties bénéficient de cette sémantique pour être plus facilement réexploitables. Ainsi, cela facilite de développement d'un système ambiant d'assistance en n'ayant qu'à produire l'architecture spécifique à la problématique ciblée que l'on souhaite résoudre à partir du modèle générique.

4 Conclusion

Il a été constaté que les solutions actuelles manquent d'un socle standard, ce qui réduit la flexibilité et l'interopérabilité de tels systèmes. De plus, les données qui en résultent peuvent manquer de formalisation. Enfin, il a été mis en lumière le fait que le facteur humain n'était pas suffisamment pris en compte.

Nous avons présenté un modèle générique qui s'inscrit dans le domaine multidisciplinaire des systèmes ambiants d'assistance. À partir de différents modèles issus des domaines des systèmes multi-agents, de l'intelligence ambiante, des activités et de l'assistance, nous avons proposé une structure formalisée réutilisable qui facilite la mise en opération de systèmes ambiants d'assistance. Le fonctionnement du système se veut complètement décentralisé et répond aux critères listés dans la section 2.1, dont l'adaptation, la flexibilité et la personnalisation. Nous avons mis un point d'honneur sur la place de l'humain dans le système en tant qu'acteur, tout en étant transparents pour la préservation de la vie privée.

5 Perspectives

Fondé sur ce modèle, nous implémenterons un système dédié à l'assistance des personnes à mobilité réduite. Pour évaluer les bénéfices de notre modèle, nous pensons qu'une

observation l'impact du système sur les connexions sociales de la personne assistée serait pertinente.

En effet, comme le soulignent les auteurs de [28] et d'autres chercheurs, le bien-être et la santé d'une personne dépend de la qualité de ses relations sociales et du fait qu'elle se sente « socialement connectée » avec les autres. En outre, il est intéressant de montrer que ce type de système permet de réduire intelligemment la charge qui pèse sur les aidants proches, sans chercher à les remplacer, notamment quand une accroche émotionnelle est nécessaire.

Remerciements

Ce travail a reçu le soutien financier du fond européen FEDER et de la Région Réunion.

Références

- [1] M. Weiser, « The Computer for the 21st Century », p. 8, 1991.
- [2] P. Rashidi et A. Mihailidis, « A Survey on Ambient-Assisted Living Tools for Older Adults », *IEEE J. Biomed. Health Inform.*, vol. 17, n° 3, p. 579-590, mai 2013, doi: 10.1109/JBHI.2012.2234129.
- [3] T. G. Stavropoulos, D. Vrakas, D. Vlachava, et N. Bassiliades, « BOnSAI: a smart building ontology for ambient intelligence », in *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics - WIMS '12*, Craiova, Romania, 2012, p. 1, doi: 10.1145/2254129.2254166.
- [4] P. Campillo-Sanchez et J. J. Gómez-Sanz, « Agent Based Simulation for Creating Ambient Assisted Living Solutions », in *Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection*, Cham, 2014, p. 319-322, doi: 10.1007/978-3-319-07551-8_28.
- [5] N. M. Garcia et J. J. P. C. Rodrigues, *Ambient Assisted Living*. 2015.
- [6] A. El murabet, A. Abtoy, A. Touhafi, et A. Tahiri, « Ambient Assisted living system's models and architectures: A survey of the state of the art », *Journal of King Saud University - Computer and Information Sciences*, vol. 32, n° 1, p. 1-10, janv. 2020, doi: 10.1016/j.jksuci.2018.04.009.

- [7] S. Triki, « Système Multi-Agent Ambient pour faciliter l'autonomie et l'accessibilité aux espaces publics des personnes ayant des déficiences cognitives », 2018.
- [8] H. Nakashima, H. Aghajan, et J. C. Augusto, Éd., *Handbook of Ambient Intelligence and Smart Environments*. Boston, MA: Springer US, 2010.
- [9] D. Hutchison *et al.*, « OutCare: Supporting Dementia Patients in Outdoor Scenarios », in *Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 6279, R. Setchi, I. Jordanov, R. J. Howlett, et L. C. Jain, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, p. 365-374.
- [10] S. Fudickar et B. Schnor, « KopAL – A Mobile Orientation System for Dementia Patients », in *Intelligent Interactive Assistance and Mobile Multimedia Computing*, Berlin, Heidelberg, 2009, p. 109-118, doi: 10.1007/978-3-642-10263-9_10.
- [11] ROSETTA. <http://www.aal-europe.eu/projects/rosetta/>.
- [12] F. J. M. Meiland *et al.*, « Participation of end users in the design of assistive technology for people with mild to severe cognitive problems; the European Rosetta project », *Int. Psychogeriatr.*, vol. 26, n° 5, p. 769-779, mai 2014, doi: 10.1017/S1041610214000088.
- [13] M. Klein, A. Schmidt, et R. Lauer, « Ontology-Centred Design of an Ambient Middleware for Assisted Living: The Case of SOPRANO », p. 8, 2007.
- [14] P. Wolf et A. Schmidt, « openAAL - the open source middleware for ambient assisted living (AAL) », p. 5, 2010.
- [15] J. Bohn, V. Coroamă, M. Langheinrich, F. Mattern, et M. Rohs, « Social, Economic, and Ethical Implications of Ambient Intelligence and Ubiquitous Computing », in *Ambient Intelligence*, W. Weber, J. M. Rabaey, et E. Aarts, Éd. Berlin/Heidelberg: Springer-Verlag, 2005, p. 5-29.
- [16] S. Mansour et J. Ferber, « Agent Groupe Rôle et Service : Un modèle organisationnel pour les systèmes multi-agents ouverts », p. 10, 2007.
- [17] A. S. Rao et M. P. Georgeff, *Formal Models and Decision Procedures for Multi-Agent Systems*. 1995.
- [18] L. Braubach, A. Pokahr, D. Moldt, et W. Lamersdorf, « Goal Representation for BDI Agent Systems », in *Programming Multi-Agent Systems*, vol. 3346, R. H. Bordini, M. Dastani, J. Dix, et A. El Fallah Seghrouchni, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, p. 44-65.
- [19] M. B. van Riemsdijk, M. Dastani, et M. Winikoff, « Goals in agent systems: a unifying framework », in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 2*, Estoril, Portugal, mai 2008, p. 713-720, Consulté le: avr. 22, 2020. [En ligne].
- [20] FIPA, « FIPA ACL Message Structure Specification ». 2002.
- [21] P. D. O'Brien et R. C. Nicol, « FIPA — towards a standard for software agents », vol. 16, n° 3, p. 9, 1998.
- [22] R. Fontaine, N. Aky, D. Payet, et R. Courdier, « Vers une utilisation éco responsable des objets connectés par la mutualisation de leurs composants physiques : une approche basée sur le concept d'artefact », 2020, p. 38.
- [23] V. Folcher et P. Rabardel, « 15. Hommes, artefacts, activités : perspective instrumentale », in *Ergonomie*, 1^{re} éd., Presses Universitaires de France, 2004, p. 251.
- [24] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, et P. Steggle, « Towards a Better Understanding of Context and Context-Awareness », in *Handheld and Ubiquitous Computing*, vol. 1707, H.-W. Gellersen, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, p. 304-307.
- [25] K. I.-K. Wang, W. H. Abdulla, et Z. Salcic, « Ambient intelligence platform using multi-agent system and mobile ubiquitous hardware », *Pervasive and Mobile Computing*, vol. 5, n° 5, p. 558-573, oct. 2009, doi: 10.1016/j.pmcj.2009.06.003.
- [26] J. Ye, G. Stevenson, et S. Dobson, « A top-level ontology for smart environments », *Pervasive and Mobile Computing*, vol. 7, n° 3, p. 359-378, juin 2011, doi: 10.1016/j.pmcj.2011.02.002.
- [27] E. Mansour *et al.*, « A Demonstration of the Solid Platform for Social Web Applications », in *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*, Montré#233;al, Qu#233;bec, Canada, 2016, p. 223-226, doi: 10.1145/2872518.2890529.
- [28] J. Holt-Lunstad, T. Robles, et D. A. Sbarra, « Advancing Social Connection as a Public Health Priority in the United States », *Am Psychol*, vol. 72, n° 6, p. 517-530, sept. 2017, doi: 10.1037/amp0000103.

Modèle réutilisable et plus inclusif de système ambiant dédié à l'assistance basée sur le paradigme multi-agent
(présentation courte)

Une stratégie de négociation multi-agents pour réduire la durée moyenne de réalisation

Ellie Beauprez

Ellie.Beauprez@univ-lille.fr

Anne-Cécile Caron

Anne-Cecile.caron@univ-lille.fr

Maxime Morge

Maxime.Morge@univ-lille.fr

Jean-Christophe Routier

Jean-Christophe.Routier@univ-lille.fr

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

Résumé

Nous étudions le problème de la réallocation de tâches pour l'équilibrage de charge dans les modèles distribués de traitement de données massives. Nous proposons une stratégie qui repose sur des agents coopératifs pour optimiser le réordonnement de tâches dans de multiples jobs devant être exécutés le plus tôt possible. Elle permet à un agent de déterminer localement la prochaine tâche à exécuter ou à déléguer grâce à ses connaissances, ses croyances et son modèle des pairs. La nouveauté réside dans la capacité des agents à identifier les opportunités et les agents limitants pour réallouer efficacement les tâches à travers des négociations bilatérales concurrentes. Nos expérimentations montrent que la durée moyenne de réalisation atteinte par notre stratégie reste proche de celle obtenue avec une heuristique classique mais que le temps d'ordonnement est significativement réduit.

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

Abstract

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

Keywords: Distributed Problem Solving, Agent-based Negotiation

1 Introduction

Les sciences des données, qui exploitent de larges volumes de données sur lesquelles des calculs sont effectués en parallèle par différents nœuds, mettent à l'épreuve l'informatique distribuée en ce qui concerne l'allocation de tâches et l'équilibrage de charge. Cet article traite d'une classe d'applications pratiques où : (a) des jobs (c.-à-d. des ensembles de tâches) concurrents doivent être exécutés le plus tôt possible, et (b) les ressources (c.-à-d. les données) requises sont distribuées. Nous considérons particulièrement le modèle de traitement le plus répandu pour traiter des données massives sur une grappe de serveurs, c.-à-d. le patron de conception Map-Reduce [18]. Les jobs y sont composés d'un ensemble de tâches exécutées par les différents nœuds où sont réparties les ressources. Comme plusieurs ressources sont requises pour réaliser une tâche sur un nœud, son exécution nécessite de récupérer des ressources disponibles sur d'autres nœuds, ce qui induit un surcoût.

De nombreux travaux adoptent le paradigme multi-agents pour aborder le problème de la réallocation de tâches et de l'équilibrage de charge dans les systèmes distribués [8]. La plupart d'entre eux adoptent l'approche orientée marché [17, 1, 16] et modélisent le problème comme un jeu non-coopératif. Par contraste, nous supposons comme [2] que les agents sont coopératifs, c.-à-d. ils ont une perception locale et partielle de l'allocation, mais ils partagent le même objectif. Nous allons ici au-delà en considérant plusieurs jobs composés d'ensembles de tâches, chacune pouvant être exécutée par un seul des agents, tous compétents. Les agents souhaitent minimiser la durée moyenne de réalisation des jobs, c.-à-d. le *flowtime* moyen. La principale difficulté provient du fait que la réaffectation d'une tâche ne modifie pas uniquement la charge de travail des agents impliqués mais également leur propre ordonnancement.

Nous proposons une stratégie de délégation des tâches, qui décide quelle transaction est suggérée ou acceptée. Elle repose sur un modèle des pairs et détermine le comportement de l'agent à chaque point de choix dans le protocole de négociation. La stratégie d'offre sélectionne une délégation potentielle, c'est-à-dire une tâche et un receveur. La règle d'acceptabilité détermine si l'agent accepte ou refuse une délégation.

Nos contributions sont les suivantes ¹.

(a) Nous formalisons le problème d'allocation multi-agents de jobs composés de tâches situées dont les coûts diffèrent selon la localisation des ressources.

(b) Nous proposons une stratégie qui identifie en continu les agents limitants et les opportunités au sein de répartitions déséquilibrées pour déclencher des négociations concurrentes et bilatérales afin de déléguer des tâches.

(c) Nous avons conduit des expériences approfondies qui montrent que notre méthode atteint un *flowtime* proche de celui obtenu par l'heuristique classique et réduit significativement le temps de réordonnement.

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

2 Travaux connexes

Les problèmes d'ordonnement classiques sont des problèmes d'optimisation sous contraintes qui peuvent être approximés par différentes heuristiques telles que la méthode de montée en gradient ou le recuit simulé (voir [12] pour une synthèse). Ces approches ne sont pas toujours adaptées à la réallocation de tâches dans des systèmes distribués où décentralisation et adaptativité sont nécessaires. En effet, d'une part, un contrôle global constitue un goulot d'étranglement en matière de performance, car il doit en permanence collecter des informations sur l'état du système. À l'opposé, nos agents prennent des décisions locales sur une allocation existante dans le but d'améliorer l'équilibre de charges. De plus, les problèmes d'or-

donnement classiques sont statiques. L'estimation inexacte du temps d'exécution des tâches, aggravée par des perturbations (consommation de tâches, libération de jobs, ralentissement des nœuds, etc.) peut nécessiter d'importantes modifications de l'allocation existante pour qu'elle reste optimale. Qui plus est, les agents peuvent agir dans des environnements dynamiques qui évoluent au cours du temps.

Pour ces raisons l'ordonnement multi-agents a reçu une attention particulière pour les problèmes d'équilibrage de charge dans des systèmes distribués (voir [8] pour une synthèse récente). Banerjee et Hecker proposent dans [3] un protocole général distribué d'allocation de ressources pour l'équilibrage de charge de jobs à gros grains sur un système massivement distribué. Ils soulignent que les interactions locales entre agents font émerger des propriétés globales. En revanche, Selvitopi et al. se penchent dans [15] sur l'ordonnement simultané de tâches de granularité plus fine, dans le but d'améliorer la localité des données et d'équilibrer les charges de travail. Leur approche est basée sur des modèles de graphes et d'hypergraphes qui utilisent des connaissances spécifiques à l'application. Notre étude vise à lier les deux niveaux de granularité par la réallocation de tâches indépendantes au sein de jobs multiples sans disposer d'information a priori sur ces jobs et tâches.

Schaerf et al. se penchent dans [14] sur le comportement adaptatif d'agents pour un équilibrage de charge efficace utilisant l'apprentissage multi-agent par renforcement. Turner et al. combinent dans [16] l'apprentissage par classification supervisée avec un processus interne de prise de décision pour l'assignation de tâche. À l'inverse, nous ne considérons aucun modèle préalable, ni des données, ni de l'environnement, car cela ne serait pas pertinent par rapport à la classe d'applications pratiques qui nous concerne.

Dans [9], les auteurs distinguent deux types de proximité : la localité des ressources qui sont requises pour l'exécution des tâches et le voisinage des agents. Sur ce dernier point, nous supposons un réseau social complètement connecté avec des coûts de communications uniformes.

D'une part, la plupart des travaux adoptent l'approche orientée marché et modélisent le problème comme un jeu non-coopératif. Par exemple, An et al. proposent dans [1] un mécanisme distribué de négociation où des agents égoïstes négocient sur des ressources à la fois le prix contractuel et la pénalité de dégroupement.

1. Cet article est une synthèse (60 %) en français de [5].

D'autre part, la plupart des algorithmes d'optimisation sous contraintes distribués (voir [11] pour une synthèse récente) tentent d'optimiser une fonction mono-objectif utilitaire (une somme des coûts à minimiser). Plus récemment, [2] vise un objectif égalitaire qui est la minimisation du temps nécessaire à la réalisation de l'ensemble des jobs (c.-à-d. le *makespan*). Nous considérons ici le problème de la coordination des décisions entre agents pour trouver une solution globalement optimale pour des fonctions multi-objectifs. Les agents tentent de minimiser la durée moyenne de réalisation de plusieurs jobs concurrents.

3 Tâches situées

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

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

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

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

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

Pour plus de simplicité, nous faisons l'hypothèse ici qu'il y a exactement un agent par nœud et que toutes les ressources sont accessibles pour tous les agents.

L'exécution d'un job (sans date butoir) consiste à exécuter un ensemble de tâches indépendantes nécessitant des ressources pour produire un résultat.

Définition 2 (Job/Tâche). *Soient \mathcal{D} un système distribué et Res l'espace des résultats. On considère un ensemble de ℓ jobs $\mathcal{J} = \{J_1, \dots, J_\ell\}$. Chaque job J_i associé à la date de libération $t_{J_i}^0$ est un ensemble de k_i tâches $J_i = \{\tau_1, \dots, \tau_{k_i}\}$ où chaque tâche τ est une fonction $\tau : 2^{\mathcal{R}} \mapsto Res$.*

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

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

Définition 3 (Coût). *Soient \mathcal{D} un système distribué et \mathcal{T} un ensemble de tâches. La fonction de coût $c : \mathcal{T} \times \mathcal{N} \mapsto \mathbb{R}_+^*$ est telle que :*

$$c(\tau, v_i) \leq c(\tau, v_j) \Leftrightarrow \sum_{\rho \in \mathcal{R}_\tau, v_i \in l(\rho)} |\rho| > \sum_{\rho \in \mathcal{R}_\tau, v_j \in l(\rho)} |\rho| \quad (2)$$

Comme la collecte de ressources distantes représente un surcoût, une tâche est moins coûteuse si les ressources nécessaires sont « plus locales » (cf. section 6).

Nous considérons le problème d'allocation multi-agents de jobs composés de tâches situées.

Définition 4 (MASTA+). *Un problème d'allocation multi-agents de jobs est un quadruplet $MASTA+ = \langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ où :*

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

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

Définition 5 (Allocation). *Une allocation pour un problème MASTA+ est un vecteur de m lots de tâches ordonnées $\vec{A} = ((B_1, \prec_1), \dots, (B_m, \prec_m))$ où chaque lot (B_i, \prec_i) est l'ensemble des tâches $(B_i \subseteq \mathcal{T})$ affectées au nœud v_i associé à un ordre total strict $(\prec_i \subseteq \mathcal{T} \times \mathcal{T})$. $\tau_j \prec_i \tau_k$ signifie que si $\tau_j, \tau_k \in B_i$ alors τ_j est exécutée avant τ_k par v_i . L'allocation \vec{A} vérifie :*

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

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

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

- $\vec{B}_i = (B_i, \prec_i)$, le lot trié de v_i ;

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

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

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

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

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

Pour évaluer la qualité d'une allocation de tâches, nous considérons le *flowtime* moyen, qui mesure le temps moyen écoulé entre la date de libération des jobs et leur date d'achèvement, et le *makespan* qui est le temps nécessaire à la réalisation de l'ensemble des jobs.

Définition 6 (*Flowtime/Makespan*). Soient *MASTA+* un problème d'allocation de tâches et \vec{A} une allocation. On définit :

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

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

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

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

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

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

- le taux de disponibilité locale de \vec{A} ,

$$L(\vec{A}) = \sum_{\tau \in \mathcal{T}} \frac{\sum_{\rho \in \mathcal{R}_\tau, v(\tau, \vec{A}) \in l(\rho)} |\rho|}{\sum_{\rho \in \mathcal{R}_\tau} |\rho|} \quad (9)$$

Contrairement au *makespan*, le *flowtime* dépend de l'ordre d'exécution des tâches sur chacun des nœuds. Le taux de disponibilité locale mesure la proportion des ressources traitées localement (Eq. 9).

Exemple 1 (*MASTA+*). À partir de $\mathcal{D} = \langle \mathcal{N}, \mathcal{E}, \mathcal{R} \rangle$ avec $\mathcal{N} = \{v_1, v_2, v_3\}$, $\mathcal{E} = \{(v_1, v_2), (v_1, v_3), (v_2, v_3)\}$ et $\mathcal{R} = \{\rho_1, \dots, \rho_9\}$ où les ressources sont répliquées sur 2 nœuds (cf. Fig. 1a), nous considérons *MASTA+* = $\langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ avec $\mathcal{T} = \{\tau_1, \dots, \tau_9\}$ où chaque tâche τ_i nécessite la ressource ρ_i , $\mathcal{J} = \{J_1, J_2, J_3\}$ tel que $J_1 = \{\tau_1, \tau_2, \tau_3\}$, $J_2 = \{\tau_4, \tau_5, \tau_6\}$ et $J_3 = \{\tau_7, \tau_8, \tau_9\}$ et la fonction de coût donnée dans la table 1. Nous supposons que le coût d'une tâche est proportionnel à la taille des ressources et qu'il est deux fois plus important si la ressource est distante. L'allocation représentée sur la figure 1b est telle que $\vec{B}_1 = (\tau_1, \tau_4, \tau_7)$, $\vec{B}_2 = (\tau_5, \tau_8, \tau_2)$ et $\vec{B}_3 = (\tau_3, \tau_9, \tau_6)$. Le *makespan* et le *flowtime* moyen sont $C_{max}(\vec{A}) = 10$ et $C(\vec{A}) = 8.33$.

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

4 Consommation et délégation

Nous décrivons ici les opérations de consommation/délégation et le protocole de négociation.

L'ajout ou la suppression d'une tâche τ dans le lot \vec{B}_i , qui modifie l'ensemble des tâches, peuvent changer l'ordre d'exécution des tâches.

Si $\tau \notin B_i$ alors $\vec{B}_i \oplus \tau$ désigne le lot qui contient l'ensemble des tâches $B_i \cup \{\tau\}$ trié selon \prec_i .

Si $\tau \in B_i$ alors $\vec{B}_i \ominus \tau$ désigne le lot qui contient $B_i \setminus \{\tau\}$ trié selon \prec_i .

Ces opérations impliquent un réordonnancement du lot.

La stratégie de consommation d'un agent spécifie l'ordonnancement des tâches dont il a la charge. Son étude va au-delà de la portée de cet article [4]. Comme nous voulons minimiser le *flowtime* moyen des jobs, nous considérons ici une stratégie orientée job qui trie le lot d'abord par job puis par tâche au sein d'un même job (les tâches d'un même job sont consécutives dans le lot). En particulier, les tâches des jobs les moins coûteux sont prioritaires sur celles des jobs les plus coûteux pour minimiser localement le délai de réalisation des jobs. Par la suite, $J_1 \triangleleft_i J_2$ signifie que les tâches de J_1 sont prioritaires sur celles de J_2 et $\tau_1 \triangleleft_i \tau_2$ que la tâche τ_1 est prioritaire sur la tâche τ_2 .

La délégation d'une tâche est un événement disruptif qui modifie l'allocation courante ; c'est

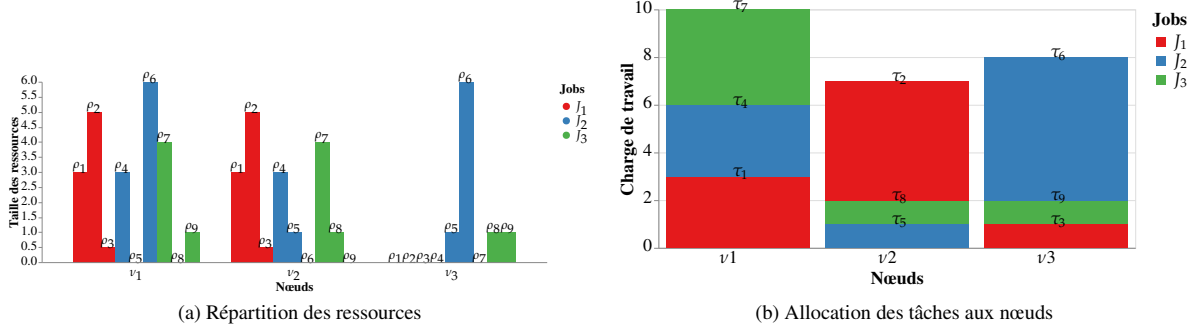


FIGURE 1 – Distribution des ressources et des tâches pour notre exemple fil rouge

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9
$c(\tau, v_1)$	3	5	0.5	3	2	6	4	2	1
$c(\tau, v_2)$	3	5	0.5	3	1	12	4	1	2
$c(\tau, v_3)$	6	10	1	6	1	6	8	1	1

TABLE 1 – Le coût des tâches pour chacun des nœuds

une réallocation.

Définition 7 (Délégation). Soient $MASTA+ = \langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ un problème d'allocation et $\vec{A} = (\vec{B}_1, \dots, \vec{B}_m)$ une allocation. Si la tâche τ est allouée au donneur v_i ($\tau \in B_i$), alors la délégation au receveur v_j aboutit à l'allocation $\delta(\tau, v_i, v_j, \vec{A})$ avec les m lots $\delta(\tau, v_i, v_j, \vec{B}_k)$ définis tels que :

$$\delta(\tau, v_i, v_j, \vec{B}_k) = \begin{cases} \overrightarrow{B_i \ominus \tau} & \text{si } k = i \\ \overrightarrow{B_j \oplus \tau} & \text{si } k = j \\ \vec{B}_k & \text{sinon} \end{cases} \quad (10)$$

Une délégation de tâche doit réduire le *makespan* local et le *flowtime* local.

Définition 8 (Délégation socialement rationnelle). Soient $MASTA+ = \langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ un problème d'allocation, \vec{A} une allocation et $\delta(\tau, v_i, v_j, \vec{A})$ l'allocation après la délégation de τ par v_i à v_j . Cette délégation est socialement rationnelle :

— vis-à-vis du *makespan* ssi le *makespan* local décroît,

$$w_j(\vec{A}) + c(\tau, v_j) < w_i(\vec{A}) \quad (11)$$

— vis-à-vis du *flowtime* ssi le *flowtime* local dé-

croît,

$$\sum_{J \in \text{jobs}(B_i \cup B_j)} \max(C_J(\overrightarrow{B_i \ominus \tau}), C_J(\overrightarrow{B_j \oplus \tau})) < \sum_{J \in \text{jobs}(B_i \cup B_j)} \max(C_J(\vec{B}_i), C_J(\vec{B}_j)) \quad (12)$$

Une allocation est stable s'il n'existe aucune délégation socialement rationnelle.

Dans une allocation stable vis-à-vis du *makespan*, les agents ne peuvent plus améliorer localement le *makespan*.

Propriété 1 (Terminaison). Soient $MASTA+ = \langle \mathcal{D}, \mathcal{T}, \mathcal{J}, c \rangle$ un problème d'allocation et \vec{A} une allocation qui n'est pas stable vis-à-vis du *makespan*. Il existe un chemin fini de délégations socialement rationnelles vis-à-vis de ce critère qui mène à une allocation stable pour le *makespan*.

Cette propriété découle du théorème 7 dans [7]. En revanche, une séquence de délégations socialement rationnelles vis-à-vis du *flowtime* ne mène pas nécessairement à une allocation stable pour ce critère.

Pour réaliser des délégations de tâches, les agents réalisent de multiples négociations bilatérales à un tour. Chaque négociation, reposant sur un protocole d'offres alternées [13], inclut trois étapes de décision : (a) l'offre de stratégie du

proposant qui sélectionne une délégation potentielle, c.-à-d. une tâche dans son lot et un receveur, (b) une règle d'acceptabilité qui permet au répondant de déterminer s'il décline ou accepte la délégation, et (c) dans ce dernier cas, la délégation est confirmée ou annulée par le proposant selon les consommations qui ont eu lieu entre-temps.

5 Stratégie de délégation

Nous décrivons ici les différentes parties de la stratégie de délégation et nous esquissons le comportement de l'agent.

Le **modèle des pairs** est construit à partir des informations échangées entre agents via des messages. En particulier, avant le processus de négociation et après chaque délégation dans laquelle il est impliqué, l'agent v_i informe ses pairs que chaque job J lui coûte $c(J, v_i)$. Comme le nombre de jobs est négligeable par rapport au nombre de tâches, la taille de ces messages est dérisoire par rapport à la description des lots. Le modèle de la cible v_j par le sujet v_i repose sur :

(a) la base de croyances du sujet, éventuellement partielle ou obsolète, qui contient les croyances concernant les coûts des jobs pour v_j ($c^i(J, v_j)$, $\forall J \in \mathcal{J}$) et les croyances concernant la charge de travail de v_j ($w_j^i(\vec{A}) = \sum_{J \in \mathcal{J}} c^i(J, v_j)$);
 (b) la stratégie de consommation de la cible supposée par le sujet, notée $(\mathcal{J}, \triangleleft_j^i)$.

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

Le sujet peut alors déduire la durée de réalisation du job J pour la cible, éventuellement après l'ajout/la suppression de τ ($C_J^i(\vec{B}_j)$, $C_J^i(\vec{B}_j \oplus \vec{\tau})$ et $C_J^i(\vec{B}_j \ominus \vec{\tau})$) ainsi que la durée de réalisation d'un job pour l'allocation, $C_J^i(\vec{A}) = \max_{v_j \in \mathcal{N}} C_J^i(\vec{B}_j)$ où $C_J^i(\vec{B}_i) = C_J(\vec{B}_i)$.

Le sujet considère que la cible est agent limitant pour le job J , noté $v_j = v_{\max}^i(\vec{A}, J)$, si la durée de réalisation de ce job pour la cible est la durée de réalisation maximale, c.-à-d. $C_J^i(\vec{A}) = C_J^i(\vec{B}_j)$.

La **règle d'acceptabilité** est une décision locale prise par le receveur d'une délégation, qui est basée sur ses connaissances et le modèle de ses pairs, pour accepter ou décliner une délégation.

Définition 9 (Acceptabilité). *La délégation*

$\delta(\tau, v_i, v_j, \vec{A})$ de la tâche τ du donneur v_i au receveur v_j dans \vec{A} est acceptable par le receveur :
 — vis-à-vis du *makespan* ssi le receveur croit que le *makespan* local décroît,

$$w_j(\vec{A}) + c(\tau, v_j) < w_j^i(\vec{A}) \quad (13)$$

— vis-à-vis du *flowtime* ssi le receveur croit que le *flowtime* local décroît,

$$\begin{aligned} \sum_{J \in \mathcal{J}} \max(C_J^j(\vec{B}_i \ominus \vec{\tau}), C_J(\vec{B}_j \oplus \vec{\tau})) < \\ \sum_{J \in \mathcal{J}} \max(C_J^j(\vec{B}_i), C_J(\vec{B}_j)) \end{aligned} \quad (14)$$

Alors que le premier critère s'appuie sur la croyance concernant la charge de travail du donneur (Eq. 13), le second repose sur la connaissance des durées de réalisation des jobs pour le receveur avant et après la délégation ainsi que ses croyances concernant les durées de réalisation pour le donneur avant et après (Eq. 14).

Pour réduire le *flowtime* moyen, la règle d'acceptabilité vérifie non seulement l'acceptabilité vis-à-vis du *flowtime* mais aussi l'acceptabilité vis-à-vis du *makespan* afin de garantir la convergence des négociations (Prop. 1).

La **stratégie d'offre** d'un donneur v_i , qui repose sur ses connaissances, ses croyances et son modèle des pairs, sélectionne une délégation en quatre étapes.

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

$$\forall \mathcal{J}' \subseteq \mathcal{J}, \quad (15)$$

$$J_* = \min_{\triangleleft_i} \{J \in \text{jobs}(\mathcal{B}_i) \cap \mathcal{J}' \mid v_i = v_{\max}^i(\vec{A}, J)\}$$

2. Sélection d'un receveur. Les jobs d'un receveur impactés par la délégation sont ceux placés après J_* selon \triangleleft_j^i . Afin de ne pas augmenter la durée de réalisation de ces jobs, notre heuristique sélectionne le receveur v_* pour qui la somme des différences entre la durée de réalisation pour l'allocation et celle pour l'agent est la plus grande,

$$\forall \mathcal{N}' \subseteq \mathcal{N}, \quad (16)$$

$$v_* = \min_{\triangleleft} \left\{ \operatorname{argmax}_{v_j \in \mathcal{N}'} \sum_{J_* \triangleleft_j^i J} (C_J^i(\vec{A}) - C_J^i(\vec{B}_j)) \right\}$$

où $<$ désigne l'ordre naturel sur les identifiants des nœuds.

3. Sélection d'une tâche. Afin de réduire les durées de réalisation, le donneur sélectionne une tâche distante dont la délégation réduira son coût puisqu'elle sera exécutée localement. Notre heuristique sélectionne donc la tâche du job J_* ou des jobs qui le précèdent dans \vec{B}_i avec le meilleur gain. En cas d'égalité, c'est la tâche prioritaire du lot qui est choisie,

$$\forall \mathcal{J}' \subseteq \mathcal{J}, \tau_* = \min_{\triangleleft_i} \left\{ \operatorname{argmax}_{\tau \in \mathcal{J}' \cap B_i \cap \{J | J = J_* \vee (J \triangleleft_i J_*)\}} c(\tau, v_i) - c(\tau, v_*) \right\} \quad (17)$$

4. Validation. Par symétrie avec le critère d'acceptabilité (cf. Def. 9), le critère de déclenchement est une décision locale prise par le donneur pour déterminer si la délégation est perçue comme étant socialement rationnelle. Dans le but de garantir la convergence du processus de négociation, la règle de déclenchement est une conjonction du critère de déclenchement vis-à-vis du *makespan* et de celui vis-à-vis du *flowtime*. Si la règle de déclenchement n'est pas vérifiée, une autre tâche ($\mathcal{J}' = \mathcal{J}' \setminus \{\tau_*\}$ dans l'étape 3) ou un autre receveur ($\mathcal{N}' = \mathcal{N}' \setminus \{v_*\}$ dans l'étape 2), éventuellement un autre job ($\mathcal{J}' = \mathcal{J}' \setminus \{J_*\}$ dans l'étape 1) est choisi. En cas d'échec, aucune délégation n'est proposée et l'agent passe en état de pause jusqu'à ce que sa base de croyances soit mise à jour et qu'une nouvelle opportunité (i.e. une délégation) soit trouvée.

Dans notre approche, une réallocation de tâche est le résultat de négociations entre agents qui adoptent tous le même **comportement** : ils alternent entre les rôles de proposant, de répondant et de contractant. Les agents exécutent leur comportement selon leurs connaissances et croyances. Le comportement des agents est spécifié dans [6] par un automate fini déterministe². Afin d'éviter les interblocages, les propositions sont associées à des dates butoirs.

La réception de message venant des pairs met à jour la base de croyances de l'agent et aucune proposition n'est envoyée tant que l'agent croit que l'allocation est stable.

Exemple 2 (Stratégie de délégation). *Considérons le problème MASTA+ et l'allocation \vec{A} de l'exemple 1. Nous supposons que les agents ont des croyances à jour et qu'ils savent qu'ils*

² <https://gitlab.univ-lille.fr/maxime.morge/smastaplus/-/tree/master/doc/specification>

adoptent tous la même stratégie de consommation. Contrairement à v_1 et v_2 , l'agent v_3 peut faire une proposition en sélectionnant :

1. le job pour lequel il est agent limitant (Eq. 15), $J_* = J_2$;
2. l'agent le moins limitant pour les jobs impactés (Eq. 16) que sont le job J_3 pour v_1 et les jobs J_1 et J_3 pour v_2 :

$$\begin{aligned} \sum_{J_* \triangleleft_1^3 J} (C_J^3(\vec{A}) - C_J^3(\vec{B}_1)) &= 0 \\ \sum_{J_* \triangleleft_2^3 J} (C_J^3(\vec{A}) - C_J^3(\vec{B}_2)) &= 8 \end{aligned} \quad (18)$$

Ainsi, $v_* = v_2$;

3. la tâche avec le gain le plus important (Eq. 17) dans J_2 ou dans les jobs antérieurs, i.e. J_1 et J_3 :

$$\begin{aligned} c(\tau_6, v_3) - c(\tau_6, v_2) &= 6 - 12 = -6 \\ c(\tau_9, v_3) - c(\tau_9, v_2) &= 1 - 2 = -1 \\ c(\tau_3, v_3) - c(\tau_3, v_2) &= 1 - 0.5 = 0.5 \end{aligned} \quad (19)$$

Ainsi, $\tau_* = \tau_3$;

4. la délégation est déclenchable puisque le critère de déclenchement vis-à-vis du *flowtime* est vérifié,

$$\begin{aligned} \Sigma_{J \in \mathcal{J}} \max(C_J(\vec{B}_3 \ominus \vec{\tau}_3), C_J^3(\vec{B}_2 \oplus \vec{\tau}_3)) &= 16.5 \\ < \Sigma_{J \in \mathcal{J}} \max(C_J(\vec{B}_3), C_J^3(\vec{B}_2)) &= 17.0 \end{aligned} \quad (20)$$

et le critère de déclenchement vis-à-vis du *makespan* est vérifié,

$$w_2^3(\vec{A}) + c(\tau, v_2) = 7.5 < w_3(\vec{A}) = 8.0 \quad (21)$$

Le donneur v_3 délègue la tâche τ_3 à l'agent v_2 pour atteindre l'allocation $\vec{A}' = \delta(\tau_3, v_3, v_2, \vec{A})$ telle que $\vec{B}'_1 = \{\tau_1, \tau_4, \tau_7\}$, $\vec{B}'_2 = \{\tau_5, \tau_8, \tau_3, \tau_2\}$ et $\vec{B}'_3 = \{\tau_9, \tau_6\}$. L'allocation obtenue est stable.

6 Évaluation empirique

L'application pratique que nous considérons est le déploiement distribué du patron de conception MapReduce pour le traitement de jeux de données massives sur une grappe de serveurs, comme avec Spark [18]. Nous nous focalisons ici sur la phase *reduce* des jobs MapReduce. Cela peut être formalisé par un problème MASTA+ où plusieurs jobs sont soumis de façon concurrente et la fonction de coût est telle que :

$$\begin{aligned} c_i(\tau, v_j) &= \Sigma_{\rho_j \in \mathcal{R}_\tau} c_i(\rho_j, v_j) \\ \text{avec } c_i(\rho_j, v_i) &= \begin{cases} |\rho_j| & \text{si } v_i \in l(\rho_j) \\ \kappa \times |\rho_j| & \text{sinon} \end{cases} \end{aligned} \quad (22)$$

où nous avons fixé empiriquement $\kappa = 2$ comme une valeur réaliste.

Notre prototype [6] est implémenté avec le langage de programmation Scala et la bibliothèque Akka [10] adaptée aux applications orientées messages, fortement concurrentes, distribuées et robustes. Nous supposons que : (a) le délai de transmission des messages est arbitraire mais non négligeable, (b) l'ordre des messages par pair émetteur-récepteur est préservé, (c) la distribution des messages est garantie. Les expériences ont été réalisées sur une lame munie de 20 CPUs avec 512Go de RAM.

Ce travail est une première étape dans l'évaluation de nos stratégies, puisque nous comparons ici le calcul d'une réallocation, c.-à-d. la résolution d'un problème MASTA+, sans les itérations induites par les consommations de tâches, même si la stratégie de consommation est nécessaire pour trier les lots de tâches de chaque agent.

Nous considérons trois métriques : (1) le *flowtime* moyen (Eq. 7), (2) le taux de disponibilité locale (Eq. 9), et (3) le temps d'ordonnement. Nous cherchons à (a) comparer les durées moyennes de réalisation des jobs atteintes par notre SMA avec celles atteintes par les approches classiques, et (b) évaluer l'accélération grâce à la décentralisation.

Les résultats atteints par le solveur polyvalent IBM® ILOG® CPLEX® lorsqu'il résout le problème sous-jacent d'optimisation mathématique discrète non linéaire sont médiocres. C'est la raison pour laquelle nous comparons notre stratégie de négociation avec un algorithme de montée en gradient qui débute avec la même allocation initiale générée aléatoirement. À chaque étape, l'algorithme de montée en gradient sélectionne parmi toutes les délégations possibles celle qui minimise le *flowtime* moyen.

Les instances de MASTA+ que nous considérons sont telles que $m \in [2;12]$ nœuds/agents, $\ell \in [2;5]$ jobs et $n = 3 \times \ell \times m$ tâches. Il y a une ressource par tâche. Chaque ressource ρ_i est répliquée 3 fois et $|\rho_i| \in [0;100]$. Nous générons 10 instances de MASTA+, et pour chacune nous générons aléatoirement 10 allocations initiales. Les hypothèses que nous voulons tester sont : (1) le *flowtime* par notre stratégie est proche de celui obtenu par l'approche classique et (2) la décentralisation réduit significativement le temps d'ordonnement.

Les figures 2a, 2b et 3 montrent les médianes et les déviations standards des métriques en fonc-

tion du nombre de nœuds avec $\ell = 4$ jobs. Il est à noter que l'algorithme de montée en gradient n'a été évalué que sur des instances MASTA+ de petites tailles en raison de son temps de réordonnement rédhibitoire. Comme à chaque étape, l'algorithme de montée en gradient considère toutes les allocations possibles, le *flowtime* atteint est meilleur que celui de notre stratégie. Puisque le surcoût par rapport au *flowtime* de notre stratégie est de 25% vis-à-vis du gradient, notre stratégie de délégation semble être efficace même si le critère d'acceptabilité vis-à-vis du *makespan* nécessaire pour la convergence peut conduire à écarter certaines délégations susceptibles de réduire le *flowtime* (cf. section 4). Cela est dû au fait que la stratégie de délégation sélectionne les tâches distantes dont la délégation réduit le coût dans le but d'améliorer le taux de disponibilité locale qui est légèrement meilleur que celui atteint par l'algorithme de montée en gradient. De plus, étant donné que ce dernier évalue à chaque itération toutes les délégations possibles, son temps d'ordonnement est bien supérieur à celui de notre stratégie de négociation. Par exemple, il est six fois plus élevé pour 9 agents et 4 jobs.

Il est intéressant de noter que l'écart entre les temps de réordonnement des deux méthodes croît exponentiellement avec le nombre d'agents, tandis que l'écart pour le *flowtime* est globalement constant. On peut s'attendre à avoir un temps de réordonnement plus grand avec une méthode de recherche locale telle que le recuit simulé sans pour autant avoir de garanties sur la qualité du résultat.

Par conséquent, même si le nombre d'agents est faible, le gain réalisé sur le *flowtime* par l'algorithme de montée en gradient sera pénalisé et annulé par le surcoût du temps de réordonnement. Ce surcoût pénalise l'équilibrage en continu des charges dans un système distribué qui devrait s'adapter aux phénomènes perturbateurs (consommation de tâches, libération de jobs, ralentissement des nœuds).

Enfin, on constate que l'efficacité de l'exécution de notre stratégie sur plusieurs cœurs croît avec le nombre d'agents et le nombre de jobs. Par exemple, avec un *flowtime* équivalent, la version décentralisée est 3 fois plus rapide pour 12 agents et 4 jobs que la version centralisée.

7 Conclusion

Dans cet article, nous avons proposé un système multi-agents pour la réallocation de tâches sur

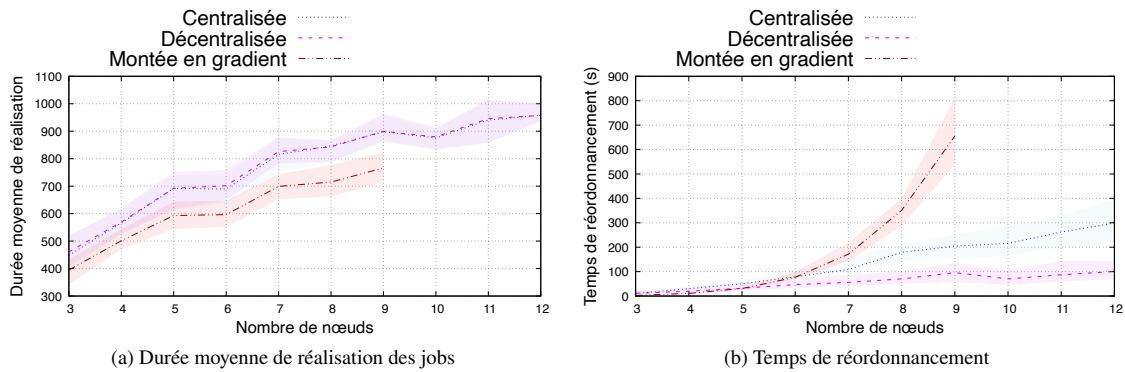


FIGURE 2 – *Flowtime* moyen et temps d’ordonnancement pour notre stratégie de négociation (centralisée ou décentralisée) et l’algorithme de montée en gradient.

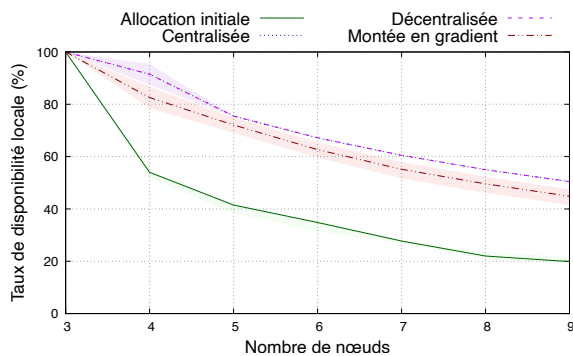


FIGURE 3 – Taux de disponibilité locale pour notre stratégie de négociation (centralisée et décentralisée), l’algorithme de montée en gradient et l’allocation initiale.

des nœuds en fonction de la localisation des ressources nécessaires afin de réduire la durée moyenne de réalisation de jobs concurrents. Nos expériences montrent que la durée moyenne de réalisation atteinte par notre stratégie est proche de celle atteinte par l’approche heuristique classique et qu’elle réduit considérablement le temps de réordonnancement. Cela est dû au fait que le processus de négociation adapte en continu l’allocation afin d’améliorer l’équilibre des charges en réduisant les durées de réalisation des jobs pour les nœuds qui sont limitants. D’une part, la stratégie de consommation consiste à exécuter d’abord les tâches des jobs les moins coûteux avant celles des jobs les plus coûteux. D’autre part, la stratégie de délégation consiste à sélectionner un job qui permet de réduire les durées de réalisation du donneur en choisissant un receveur qui n’est pas limitant pour les jobs impactés et en choisissant une tâche dont la délégation réduit le coût puisqu’elle sera exécutée locale-

ment. Une analyse de sensibilité pour étudier l’influence du facteur de réplication va au-delà de la portée de cet article, mais mériterait une étude approfondie. Évidemment, notre approche passe l’échelle, car elle gère un grand nombre de tâches grâce à des décisions locales des agents à propos de la prochaine tâche à déléguer/exécuter. En outre, le surcoût de la négociation est négligeable par rapport au bénéfice de l’équilibrage des charges, car aucune négociation n’est déclenchée lorsque les agents estiment que l’allocation est stable.

Une étude comparative de notre stratégie avec différentes méthodes de résolution distribuées dépasse le cadre de cet article mais serait intéressante pour le futur. Un certain nombre de nos expériences suggèrent que nous devons étendre notre cadre de négociation pour envisager (a) des échanges de tâches afin d’améliorer la durée moyenne de réalisation des allocations stables, et (b) une règle d’acceptabilité moins restrictive qui écarte actuellement certaines délégations qui pourraient réduire le *flowtime* moyen. En général, les travaux futurs doivent étendre le processus de réallocation des tâches vers un processus itératif, dynamique et continu, qui se déroule de manière concurrente à l’exécution des tâches, pour permettre au système distribué de s’adapter aux phénomènes perturbateurs (consommation de tâches, libération de jobs, ralentissement des nœuds).

Références

- [1] Bo An, Victor Lesser, David Irwin, and Michael Zink. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In *Proc. of 9th International Conference on Autonomous*

- Agents and Multiagent Systems (AAMAS)*, pages 981–988, 2010.
- [2] Quentin Baert, Anne-Cécile Caron, Maxime Morge, Jean-Christophe Routier, and Kostas Stathis. Stratégie situationnelle pour l'équilibrage de charge. In *Actes des 27ièmes journées francophones sur les systèmes multi-agents (JFSMA)*, pages 9–18. Cépaduès, 2019.
- [3] Soumya Banerjee and Joshua P. Hecker. A Multi-agent System Approach to Load-Balancing and Resource Allocation for Distributed Computing. In *Proc. of the 1st Complex Systems Digital Campus World E-Conference 2015*, pages 41–54. Springer International Publishing, 2017.
- [4] Ellie Beauprez, Anne-Cécile Caron, Maxime Morge, and Jean-Christophe Routier. Stratégie multi-agents de négociation pour la réduction du flowtime. <https://tinyurl.com/MASTAPlus>, 2020. Rapport de recherche. Université de Lille.
- [5] Ellie Beauprez, Anne-Cécile Caron, Maxime Morge, and Jean-Christophe Routier. A Multi-Agent Negotiation Strategy for Reducing the Flowtime. In *Proc. of 13th International Conference on Agents and Artificial Intelligence (ICAART)*, volume 1, pages 58–68, February 2021.
- [6] Ellie Beauprez and Maxime Morge. Scala implementation of the Extended Multi-agents Situated Task Allocation. <https://gitlab.univ-lille.fr/maxime.morge/smastaplus>, 2020.
- [7] Ulle Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Negotiating Socially Optimal Allocations of Resources. *Journal of Artificial Intelligence Research*, 25 :315 – 348, 2006.
- [8] Yichuan Jiang. A survey of task allocation and load balancing in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(2) :585–599, 2016.
- [9] Yichuan Jiang and Zhaofeng Li. Locality-sensitive task allocation and load balancing in networked multiagent systems : Talent versus centrality. *Journal of Parallel and Distributed Computing*, 71(6) :822–836, 2011.
- [10] Lightbend. Akka is the implementation of the actor model on the JVM. <http://akka.io>, 2020.
- [11] Gauthier Picard. Optimisation sous contraintes distribuée : une introduction au domaine. In *Actes des 26ièmes journées francophones sur les systèmes multi-agents, (JFSMA)*, pages 43–52. Cépaduès, 2018.
- [12] Michael L. Pinedo. *Scheduling. Theory, Algorithms, and Systems. Third Edition*. Springer, 2008.
- [13] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1) :97–102, 1 1982.
- [14] Andrea Schaerf, Y. Shoham, and Moshe Tennenholtz. Adaptive load balancing : A study in multi-agent learning. *Journal of Artificial Intelligence Research*, 2 :475–500, 1995.
- [15] Oguz Selvitopi, Gunduz Vehbi Demirci, Ata Turk, and Cevdet Aykanat. Locality-aware and load-balanced static task scheduling for MapReduce. *Future Generation Computer Systems*, 90 :49–61, 2019.
- [16] Joanna Turner, Qinggang Meng, Gerald Schaefer, and Andrea Soltoggio. Distributed Strategy Adaptation with a Prediction Function in Multi-Agent Task Allocation. In *Proc. of 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 739–747, 2018.
- [17] Michael P Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1 :1–23, 1993.
- [18] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauly, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets : A fault-tolerant abstraction for in-memory cluster computing. In *Proc. of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*; San Jose, CA, USA, pages 15–28, 2012.

Enchères pour l'Amélioration des Communications lors de l'Allocation de Tâches

F. Quinton^a C. Grand^a C. Lesire^a
 felix.quinton@onera.fr christophe.grand@onera.fr charles.lesire@onera.fr

^aONERA/DTIS, Université de Toulouse, France

Résumé

Nous proposons une formule d'évaluation des offres pour résoudre le problème d'allocation de tâches aux systèmes multi-robot. Nous nous concentrons sur l'amélioration des communications entre les robots et avons introduits un nouveau terme spécialisé pour l'évaluation des enchères. Nous avons évalué notre méthode dans un scénario de patrouille, et avons démontré que la préservation des communications améliorerait la robustesse du système multi-robots, en particulier lorsque les robots sont sujets à des pannes.

Mots-clés : *Systèmes multi-robots, Allocation de tâches, Approche basées enchères, Mission de surveillance*

Abstract

We propose a novel bid valuation formula to solve the multi-robot task allocation problem. We focused on monitoring communications among the robots, and improved the communication network's connectivity through the introduction of a dedicated term in the bid valuation formula. We tested our method in a simulated patrolling scenario, and demonstrated that preserving communications improved the robustness of the multi-robot system, especially when robots are prone to failure.

Keywords: *Multi-robots systems, Task allocation, Auction-based methods, Surveillance mission*

1 Introduction

Les systèmes multi-robots (SRM) sont un domaine de recherche actif depuis longtemps, et leur application aux missions de surveillance a été largement étudiée [1]. Cependant, une stratégie de coordination fiable est nécessaire pour qu'un SMR fonctionne de manière sûre et efficace. Une étape importante pour atteindre la coordination d'un SMR est la ré-

solution du problème d'attribution des tâches multi-robots (ATMR), qui consiste à attribuer un ensemble de tâches aux agents du SMR [2]. L'ATMR étant un problème NP, la plupart des recherches visant à le résoudre dans des scénarios en temps réel se concentrent sur les méthodes approchées.

Parmi les nombreuses méthodes proposées pour l'ATMR, les approches basées marché (ABM) ont fait l'objet de nombreux travaux de recherche. Dans ces méthodes, les robots qui composent la SMR sont en concurrence pour obtenir les tâches, qui sont attribuées par le biais d'un mécanisme de marché imitant les transactions économiques du monde réel. En particulier, une sous-catégorie de ABM ayant reçu beaucoup d'attention est celle des approches basées enchères (ABE), dans lesquelles les tâches sont vendues par un système d'enchères, dont les commissaires priseurs sont les robots [3]. Les ABA excellent dans la répartition efficace et robuste des tâches simples [3]. Toutefois, les ABE nécessitent un réseau de communication (RC) fiable pour que le système de marché fonctionne correctement. Par exemple, si un robot est déconnecté du RC, il peut être incapable de communiquer ses offres. Par conséquent, une ABE pourrait ne pas attribuer de tâche à ce robot. Ainsi, une connectivité du RC entraîne une augmentation des inefficacités et des vulnérabilités. Pourtant, peu de travaux sont consacrés à l'amélioration du RC dans la SMR.

Dans cet article, nous présentons une méthode d'évaluation des enchères qui aborde cette question en incitant à la préservation du RC. Les missions de patrouille impliquent généralement des modifications dynamiques du RC, qui peuvent être causées, par exemple, par l'augmentation de la distance entre les robots. Notre méthode consiste à pénaliser les attributions qui divisent le RC en sous-RC plus petits reliant des sous-équipes ne com-

Démonstration

muniquant pas, tout en récompensant les attributions qui fusionnent des sous-RC en RC plus grands reliant des robots par des communications multi-hop. Nous appliquons cette méthode de tarification à une mission de patrouille dans laquelle les robots rencontrent des pannes qui les empêchent de surveiller certaines zones. Nos résultats montrent que la prise en compte de la connectivité du RC dans le processus d'attribution des tâches améliore la robustesse du SMR face à ces défaillances.

Après une revue de la littérature pertinente, nous exposerons la mission étudiée et décrivons le schéma ABE que nous avons utilisé. Ensuite, nous présenterons notre méthode d'évaluation des enchères, qui inclue notre terme de préservation de la communication. Puis, nous détaillerons nos expériences numériques et la façon dont nous avons modélisé les défaillances des robots. Les résultats expérimentaux en résultant sont ensuite discutés. Nous concluons notre article en résumant nos principaux résultats et en proposant des orientations pour nos futures recherches.

2 Revue de Littérature

L'utilisation des ABM pour résoudre l'ATMR a fait l'objet de nombreuses recherches [3], et des améliorations ont été apportées aux ABE pour les adapter aux besoins spécifiques des missions de patrouille multi-robots, tels que des robots capables de rejoindre ou de quitter l'équipe à tout moment [4], ou l'inclusion d'un superviseur robotique redistribuant les tâches des robots peu performants [5].

Cependant, les ABE dépendent fortement de la fiabilité des communications, car la qualité de l'allocation obtenue fluctue fortement en fonction des messages d'enchère qui sont perdus. En général, les performances obtenues par les ABE se détériorent à mesure que le RC devient moins fiable [6]. [7] ont mené une étude comparative incluant de nombreuses variantes d'ABE, dans un contexte de communications détériorées. Plus précisément, les messages partagés pour l'application des ABE avaient une probabilité de succès indiquant la probabilité que le message soit reçu par ses destinataires. Les auteurs ont montré que toutes les variantes voient leurs performances se détériorer lorsque la probabilité de perte du message augmente. Lorsque l'objectif de la mission est de minimiser la

longueur du trajet du robot sur une carte générée de manière aléatoire, les expériences numériques des auteurs ont produit des résultats mitigés, car aucune variante de l'ABE n'était meilleure que les autres sur l'ensemble des probabilités de perte de messages. À la lumière de ces résultats, il est clair que des communications peu fiables sont très préjudiciables à la qualité des solutions au problème d'ATMR produites par les ABE, ce qui motive notre étude. Pour résoudre ces problèmes, les auteurs se sont concentrés sur l'amélioration des protocoles de communication entre les robots. [8] ont proposé une méthode pour réduire les exigences de communication en utilisant un système d'enchères single item (ESI). Les commissaires-priseurs n'envoient les messages d'annonce qu'aux enchérisseurs avec lesquels ils partagent immédiatement un lien de communication dans le réseau. La méthode proposée dans [9], basée sur un système ESI, permet aux agents mandataires de recueillir et de filtrer les offres des agents voisins. [10] permettent aux robots recevant une annonce de décider s'ils doivent la relayer à leurs coéquipiers en fonction d'une estimation de leurs offres.

Une alternative pour assurer la stabilité du RC serait de concevoir une formule d'évaluation des offres qui tienne compte de la qualité du réseau, et d'inciter les robots à privilégier les tâches qui leur permettraient de maintenir un RC bien connecté. [11] ont défini une mesure appelée mesure de proximité, dont le but est d'évaluer la proximité d'un robot par rapport à ses voisins. Cette mesure est utilisée dans la formule de calcul des enchères pour tenir compte des modifications apportées au RC pendant le processus de répartition des tâches. Des simulations numériques ont permis d'établir que l'inclusion de la mesure de proximité dans l'évaluation des offres améliore la qualité des solutions produites. Cependant, cette méthode utilise la distance entre les robots comme seule mesure de la qualité de la communication, sans tenir compte de l'éventuelle présence d'obstacles empêchant les communications, et n'intègre donc pas de mesure réelle de la qualité de la communication dans l'évaluation des enchères.

Notre contribution se distingue de cette étude car notre terme de préservation de la communication est basé sur la connectivité du RC,

Démonstration

et en particulier sur le nombre de robots dans les sous-RC, plutôt que sur la distance entre les robots. En outre, nous abordons un scénario différent, la mission de surveillance, et considérons la possibilité que les robots subissent des pannes, afin d'évaluer l'effet de la surveillance de la connectivité du RC sur la robustesse de notre méthode face aux pannes.

3 Formulation du Problème

3.1 Mission de surveillance

Dans cet article, nous adopterons le cadre des missions de surveillance multi-robots, et considérerons l'oisiveté instantanée des noeuds comme la mesure permettant de déterminer l'efficacité des algorithmes proposés. L'oisiveté instantanée des noeuds $w \in \mathcal{W}$ est définie par l'équation (1).

$$I_w = t - t_{last}^w \quad (1)$$

Où $t \in \mathbb{R}_+$ est la date actuelle, et $t_{last}^w \leq t$ est la date de la dernière visite du noeud w . De nombreux critères permettent d'évaluer les performances en termes d'oisiveté, et nous nous concentrerons sur l'oisiveté moyenne et l'oisiveté maximale, notées \bar{I} et I_{max} . Pour résoudre ce problème, nous utilisons une équipe de robots notée \mathcal{R} . Chaque robot $r \in \mathcal{R}$, est décrit par sa vitesse v_r et sa portée de communication c_r . Nous supposons que les robots sont capables de déterminer leur position avec précision. Bien que nous envisagions des robots homogènes pour nos expériences numériques, notre méthode pourrait également être utilisée pour résoudre l'ATMR pour des équipes de robots hétérogènes. Notre algorithme de prise de décision étant distribué, les robots doivent partager (resp. recevoir) des informations avec (resp. de) leurs coéquipiers afin d'entretenir une connaissance locale de leur environnement et du comportement de leurs coéquipiers. En particulier, les données partagées incluent les enchères, l'attribution actuelle des noeuds aux robots, et les dates de dernière visite des noeuds. Ces deux derniers éléments sont essentiels au calcul d'enchères pertinentes. De plus, les enchères elles-mêmes doivent être partagées entre le plus grand nombre possible de robots pour que l'enchère aboutisse à la meilleure décision. Pour ces raisons, nous souhaitons nous assurer que le réseau de com-

munication qui relie les robots est hautement connecté.

3.2 Schéma d'enchères séquentielles single item

Pour résoudre le problème décrit dans la section 3.1, nous utilisons un système d'enchères séquentielles single item (ESSI) [12]. Les ESSI sont un outil simple permettant de prendre en compte les synergies entre les tâches dans une ABE. Dans le cadre des ESSI, les tâches des commissaires-priseurs sont attribuées l'une après l'autre, lors de cycles d'enchères consécutifs. Les tâches sont mises aux enchères une par une, et les robots envoient leurs offres une à une, en tenant compte de leur plan actuel pour évaluer la tâche mise aux enchères. Le commissaire-priseur clôture un tour après avoir attendu les offres pendant une durée déterminée, afin de tenir compte des délais de communications.

Une représentation schématique d'un SMR résolvant l'ATMR par le biais d'un schéma ESSI est affichée dans la Fig. 1. Les listes s correspondent à la liste des robots des articles à mettre aux enchères, et les listes $exec$ à la liste des robots des tâches à exécuter. La vente aux enchères se déroule en plusieurs tours, chacun correspondant à un robot qui vend ses tâches. Lors de la vente de ses tâches, un robot assume le rôle de commissaire-priseur, ce qui signifie qu'il est chargé de l'annonce de l'enchère, ce qui correspond à la diffusion des spécifications de la tâche au plus grand nombre de coéquipiers possible. Le commissaire-priseur rassemble également les offres de ses coéquipiers et les évalue afin de déterminer quels coéquipiers doivent recevoir la tâche. Dans la Fig. 1a, le robot 1 est le commissaire-priseur, et doit mettre aux enchères les tâches t_1 et t_2 . Comme dans les enchères réelles, le commissaire-priseur annonce un objet disponible à la vente, c'est-à-dire le mandat d'exécuter la tâche t_1 , par un message d'annonce noté (A1). Chaque robot calcule son offre selon un processus d'évaluation des offres, noté (A2), qui tient compte de la distance entre la position du robot et la position de la tâche, et de l'impact de l'attribution sur le futur CN, et sera détaillé dans la section 4. Ensuite, les robots envoient au commissaire-priseur un message d'offre (A3) contenant leur offre pour la tâche t_1 . Notez que le commissaire-priseur fait égale-

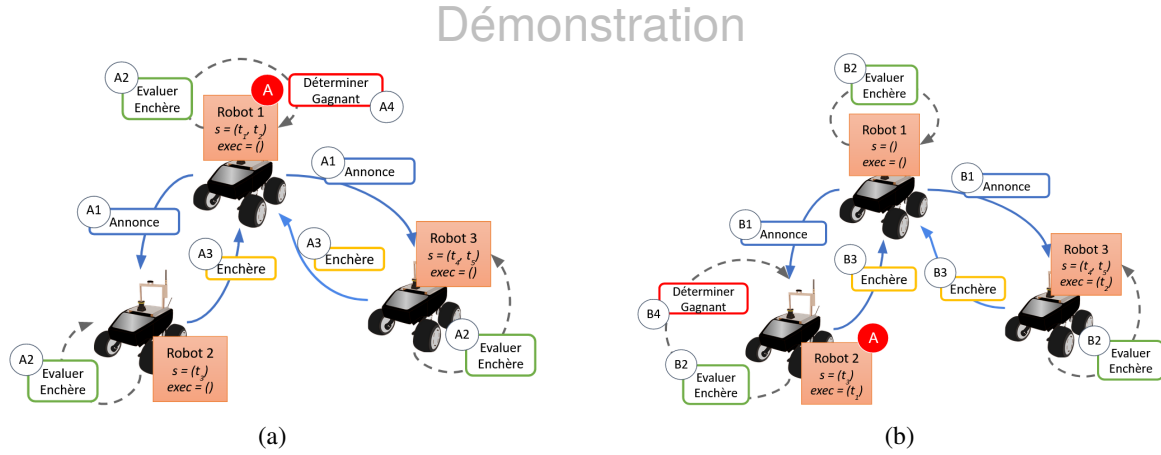


FIGURE 1 – Représentation du processus d'allocation de tâche avec un système d'enchères séquentielles single item. Les flèches pleines représentent les échanges de messages, et les flèches pointillées représentent les calculs. Les chiffres indiquent l'ordre dans lequel les processus se déroulent. Le jeton rouge "A" indique le commissaire-priseur de l'enchère en cours. Les listes s correspondent aux listes d'articles à mettre aux enchères de chacun des robots, et les listes $exec$ correspondent aux listes de tâches à effectuer de chacun des robots. Dans la figure 1a, le robot 1 est le commissaire-priseur et vend la tâche t_1 . Dans la figure 1b, le robot 1 a vendu les tâches t_1 et t_2 , et a passé le rôle de commissaire-priseur au robot 2, qui mettra aux enchères la tâche t_3 .

ment des offres dans les enchères qu'il mène. Après avoir attendu un temps limité, le robot 1 détermine que le gagnant de l'enchère est le robot 2 (A4), et lui donne le mandat d'exécuter la tâche t_1 . Ensuite, le robot 1 répète le même processus pour vendre la tâche t_2 au robot 3. Le robot 1 ayant vendu toutes ses tâches, il passe le rôle de commissaire-priseur au robot 2, qui répète le processus d'enchère de la Figure. 1b afin de vendre aux enchères la tâche t_3 et de se l'attribuer à lui-même (B4). Ensuite, le robot 3 devient le commissaire-priseur et met aux enchères les tâches t_4 et t_5 . Les rondes d'enchères sont lancées à intervalles réguliers, et un robot conserve le rôle de commissaire-priseur tant qu'il a des tâches dans sa liste de tâches à vendre. Notons qu'à l'initialisation, un robot détient toutes les tâches.

4 Enchère Préservant les Communications

La fonction d'évaluation des enchères est l'un des éléments les plus importants d'une ABE. Elle indique l'intérêt qu'un agent accorde à la tâche en délibération, et donc détermine quel robot remportera une enchère, puisque le gagnant est le robot ayant déposé l'offre la plus élevée. Notre fonction d'évaluation des enchères est composée de deux termes conçus

pour prendre en compte les principaux aspects du scénario. Nous utilisons l'oisiveté instantanée du noeud mis aux enchères, w , évaluée par le robot enchérisseur r au moment t de la réception du message d'annonce, notée $I_r(t, w)$, ainsi que le temps de déplacement du robot r jusqu'au noeud w , pour évaluer l'impact de l'attribution de w à r sur les performances de l'allocation par rapport à la métrique de l'oisiveté. La formulation donnée dans l'équation (2) fait écho au terme d'oisiveté proposé par [13], mais nous allouons chaque point exactement une fois.

$$I_r(t, w) = 1 + \frac{t - t_{last}^w}{d(r, w)/v_r} \quad (2)$$

Où $t \in \mathbb{R}_+$ est l'heure à laquelle le calcul a lieu, $t_{last}^w \leq t$ est la date de la dernière visite du noeud w , $v_r \in \mathbb{R}_+$ est la vitesse du robot r , et $d(r, w) \in \mathbb{R}_+$ est la longueur du chemin le plus court entre la position de r et w .

Pour préserver un RC fortement connecté, nous introduisons un terme de communication $K_r(t, w)$ décrit dans l'équation (3).

$$K_r(t, w) = \frac{|C_{\mathcal{N}(t)}(r)|}{|C_{\mathcal{N}(t + \frac{d(r, w)}{v_r})}(r)|} - 1 \quad (3)$$

Où $\mathcal{N}(t) = (\mathcal{R}, \mathcal{F})$ est un graphe décrivant le réseau de communication au moment t , avec

Démonstration

\mathcal{F} les paires de robots capables de communiquer entre eux, et $C_\Gamma(r)$ est la composante connexe du graphe Γ qui contient le robot r . Notez que chaque robot maintient une version locale de $\mathcal{N}(t)$, de sorte que le calcul de $K_r(t, w)$ est effectué en utilisant uniquement les connaissances locales de r . Ce terme est le rapport entre le nombre de robots dans le même sous-RC que le robot enchérisseur r à la date de début de l'enchère t , et le nombre de robots dans le même sous-RC que r à la date future $t + \frac{d(r,w)}{v_r}$ à laquelle il atteindra le noeud mis aux enchères, w . Pour calculer ce deuxième nombre, nous devons prévoir les positions des coéquipiers de r dans le futur, ce qui nécessite une connaissance locale régulièrement mise à jour des destinations et des dates d'arrivée des coéquipiers. Formellement, nous définissons le nombre de robots dans le même sous-RC que r comme le cardinal de la composante connexe qui contient r dans la représentation topologique du RC. Notez que nous soustrayons 1 au ratio pour nous assurer que $K_r(t, w)$ est négatif si l'affectation de w à r diminue le nombre de robots dans son sous-NC.

En pratique, il est très courant que $I_r(t, w)$ soit beaucoup plus grand que $K_r(t, w)$, ce qui supprime en grande partie l'impact de $K_r(t, w)$ sur le résultat de l'enchère. Pour résoudre ce problème, la formule finale d'évaluation des offres est donnée par l'équation (4), qui garantit que le terme d'oisiveté fluctue dans $[0, 1[$. Il convient de noter que des valeurs plus élevées de β aplatissent le terme d'oisiveté, et que α pourrait être utilisé pour moduler l'impact du terme de communication sur l'offre globale. Nous avons utilisé $\beta = 1$ et $\alpha = 1$ pour produire nos résultats numériques.

$$B_r(t, w) = \frac{I_r(t, w)}{\beta + I_r(t, w)} - \alpha K_r(t, w) \quad (4)$$

5 Simulation numériques

L'objectif des expériences décrites dans cette section est d'évaluer la performance de l'ESSI proposée dans un scénario réaliste qui reproduit la surveillance de deux immeubles de bureaux similaires reliés par un couloir. Dans cette simulation, la capacité d'une paire de robots à communiquer dépend de leur portée de communication. De plus, la commu-

nication est parfaite lorsque les robots sont à portée. Nous nous sommes concentrés sur l'évaluation de l'impact des pannes subies par les robots sur les performances du SMR. Plus précisément, nous avons développé un environnement de simulation dans lequel les communications sont utiles pour adapter l'allocation des tâches aux pannes, qui se produisent pendant l'exécution de la mission. Cela nous a permis d'évaluer l'effet de la prise en compte de la connectivité du RC dans l'évaluation des offres sur la robustesse de la méthode d'attribution des tâches aux pannes subies par les robots.

5.1 Pannes subies par les robots

Outre le cadre des missions de surveillance, les robots sont également susceptibles de subir des pannes qui les empêchent de se rendre sur certains noeuds. Pour simuler ces pannes, on définit pour chaque robot et chaque noeud w , un seuil $\overline{n_{r,w}}$, qui indique le nombre maximum d'observation qu'il peut effectuer pour chaque noeud, avant de rencontrer une panne. Les robots maintiennent une liste contenant le nombre d'observation effectuées sur chaque noeud, notés $n_{r,w}$, et les robots ne peuvent pas surveiller les noeuds tels que $n_{r,w} \geq \overline{n_{r,w}}$. Pour les noeuds qui ne provoquent pas de pannes pour un robot r , nous fixons $\overline{n_{r,w}} = +\infty$. Notons que les robots ignorent délibérément les seuils $\overline{n_{r,w}}$ lorsqu'ils procèdent à l'attribution des tâches, de sorte que la défaillance se produit de manière inattendue, pour mieux reproduire une situation réelle.

Ce processus reproduit deux problèmes réels qui se posent lors des missions de surveillance. Premièrement, la surveillance de certains noeuds peut nécessiter des capteurs spécifiques, de sorte qu'une défaillance de ces capteurs sur un robot pourrait l'empêcher de surveiller ces noeuds. Deuxièmement, en surveillant une zone, les robots peuvent détecter des éléments qui nécessitent un équipement spécifique pour être correctement évalués. Dans ce cas, si le robot qui a effectué la détection initiale n'est pas équipé pour effectuer la tâche de surveillance complète, il doit faire appel à ses coéquipiers.

5.2 Paramètres de la simulation

Pour produire des résultats numériques, nous avons conçu la carte présentée dans la Fi-

Démonstration

gure 2, et avons mener des missions de surveillance avec 5 robots ayant une portée de communication égale à 75 unités de distance, et avec 10 robots ayant une portée de communication égale à 50 unités de distance. Pour chaque robot, 16 noeuds avaient un $\overline{n_{r,w}}$ fini, ce qui signifie qu'ils étaient susceptibles de provoquer une panne, tandis que $\overline{n_{r,w}}$ était fixé à $+\infty$ pour les 24 autres noeuds. Tous les robots avaient une vitesse de déplacement de 20 unités de distance par unité de temps, et commençaient au noeud indiqué par un carré rouge. Les délais de mise aux enchères ont été fixés à 0,1 secondes après l'annonce de l'enchère, ce qui était plus que suffisant pour que les robots puissent calculer leurs offres. Nous avons exploré les paramètres relatifs au taux d'échec des robots, c'est-à-dire le nombre de visites $\overline{n_{r,w}}$ avant que les noeuds ne provoquent effectivement des pannes, ainsi que la taille de l'équipe $|\mathcal{R}|$, et la portée de communication c_r , qui est la même pour tous les robots.

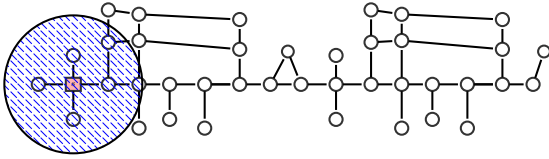


FIGURE 2 – Représentation métrique de la carte de la mission. La largeur totale de la carte est égale à 580 unités de distance, et sa hauteur totale à 135 unités de distance. Le cercle en pointillé autour de s correspond à la portée de communication des robots avec $\text{range} = 75$. Les robots commencent la mission au niveau du noeud s .

Nous avons résolu l'ATMR en utilisant un système ESSI avec et sans notre terme de communication présenté dans l'Equation (3), c'est-à-dire avec $\alpha = 0$ et $\alpha = 1$ dans l'Equation (4).

5.3 Protocole d'enchères

Pour résoudre l'ATMR avec un système ESSI, les robots doivent échanger des messages, comme indiqué dans la Figure 1. Dans ce paragraphe, nous allons décrire le protocole d'enchères, en détaillant l'algorithme du robot commissaire-priseur (Algorithme 1) et

l'algorithme des robots enchérisseurs (Algorithme 2).

Algorithme 1: Algorithme du commissaire-priseur

Data : Les S_r des noeuds à enchérir,
La durée \bar{t} d'une enchère.

```

1 for  $w \in S_r$  do
2   SendAnnouncement ( $w, t + \bar{t}$ );
3   while time left do
4     if ReceiveBid ( $B_{r'}(w)$ ),  $r' \neq r$ 
5       then
6          $B_w \leftarrow B_w \cup B_{r'}(w)$ ;
7     if  $n_{r,w} < \overline{n_{r,w}}$  then
8        $B_w \leftarrow B_w \cup \text{ComputeBid}(w)$ ;
9     if  $B_w \neq \emptyset$  then
10       $r^* \leftarrow B_w$ ;
11      if  $r^* = r$  then
12        Plan ( $w$ );
13      else
14        SendAward ( $r^*, w$ );
15    else
16       $S_r \leftarrow S_r \cup w$ ;
```

Le commissaire-priseur met aux enchères ses noeuds séquentiellement (ligne 1). Pour lancer une enchère, il annonce un noeud à l'aide d'un message d'annonce, qui contient l'emplacement de la tâche et la date limite de l'enchère (ligne 2). Ensuite, le commissaire-priseur attend de recevoir des enchères en réponse à son annonce, jusqu'à ce que la date limite de l'enchère soit atteinte (ligne 3). A la réception d'un message d'enchère, le commissaire-priseur enregistre le couple composé de l'identifiant du robot enchérisseur et de la valeur de son enchère (lignes 4 et 5). Le commissaire-priseur enchérit également pour la tâche qu'il met aux enchères, si cette tâche ne fait pas partie de ses tâches interdites (lignes 6 et 7). Enfin, le commissaire-priseur peut déterminer le gagnant de l'enchère, c'est-à-dire le robot ayant déposé l'offre avec la valeur la plus élevée (ligne 9), et lui attribuer la tâche (lignes 10 à 13). En fonction des offres reçues, le commissaire-priseur peut soit s'attribuer la tâche et la planifier afin de l'exécuter, soit l'attribuer à un coéquipier, via la publication d'un message d'attribution. Si aucune offre n'a été reçue, alors la tâche figure sur la liste noire du commissaire-priseur, et elle est donc remise dans la liste des tâches à mettre aux enchères afin d'être remise en

Démonstration

délibération ultérieurement (ligne 15).

Algorithm 2: Algorithme des enchérisseurs

```

1 if ReceiveAnnouncement ( $w, \bar{t}$ ) then
2   | if  $t < \bar{t}$  et  $n_{r,w} < \overline{n_{r,w}}$  then
3   |   | SendBid (ComputeBid ( $w$ ));
4 if ReceiveAward ( $r^*, w$ ) then
5   | if  $r^* = r$  then
6   |   | Plan ( $w$ );

```

Les robots enchérisseurs attendent les messages du commissaire-priseur. Lorsqu'ils reçoivent un message d'annonce, ils vérifient que la tâche correspondante ne figure pas dans leur liste de tâches interdites et que l'enchère n'est pas expirée, avant de calculer leur offre pour la tâche mise aux enchères et de l'envoyer au commissaire-priseur (lignes 1 à 3). Lorsqu'ils reçoivent un message d'attribution, ils planifient la tâche s'ils ont gagné l'enchère (lignes 4 à 6).

Un autre protocole important à détailler décrit le comportement des robots après qu'ils aient terminé l'exécution d'une tâche, c'est-à-dire après avoir surveillé un noeud. Ce comportement est décrit par l'algorithme 3. Les robots doivent vérifier si la tâche qu'ils ont surveillée a provoqué une panne et doit être ajoutée à leur liste de tâches interdites, en fonction du nombre de fois qu'ils ont surveillé le noeud considéré. Si le nombre de visites dépasse le seuil, noté $\overline{n_{r,w}}$, alors la visite n'améliore pas l'oisiveté du noeud et la tâche doit être à nouveau mise aux enchères (lignes 1 et 2). Sinon, l'oisiveté du noeud est réinitialisée à la date présente (ligne 5), et le robot notifie l'équipe (ligne 6). Notons que les tâches qui sont exécutées avec succès sont également remises aux enchères (ligne 7) afin de garantir que les tâches continuent à changer de mains si cela peut améliorer les performances de l'allocation. De plus, cette procédure, ainsi que la procédure de mise aux enchères présentée dans l'Algorithm 1, garantit qu'à tout moment, chaque noeud est soit attribué, soit en attente d'attribution, et n'est jamais dupliqué.

5.4 Données partagées

Afin d'utiliser les algorithmes présentés dans la section 5.3 et d'effectuer les calculs présentés dans la section 4, les robots doivent

Algorithm 3: Comportement des robots après avoir observé un noeud.

Data : Le noeud surveillé, w ,
 Le nb. de visite de w par r , $n_{r,w}$,
 Le nb. max. de visite de r sur w , $\overline{n_{r,w}}$,
 La liste S_r des noeuds à enchérir,
 La liste T_{last}^r des dernière visites.

```

1 if  $n_{r,w} \geq \overline{n_{r,w}}$  then
2   |  $S_r \leftarrow S_r \cup w$ ;
3 else
4   |  $n_{r,w} \leftarrow n_{r,w} + 1$ ;
5   |  $T_{last}^r[w] \leftarrow t$ ;
6   | SendLastVisitDates ( $T_{last}$ );
7   |  $S_r \leftarrow S_r \cup w$ ;

```

partager plusieurs informations sur l'état actuel de la mission. Ils partagent leurs connaissances locales au mieux de leurs capacités, c'est-à-dire à chaque coéquipier qui se trouve dans le même sous-RC. Pour évaluer l'oisiveté des noeuds, les robots doivent partager leurs connaissances des dates de dernières visites pour chaque noeud, T_{last} . Chaque robot $r \in \mathcal{R}$ maintient une version locale de T_{last} , dénommée T_{last}^r , qui est mise à jour lorsqu'il reçoit la version d'un autre robot. Ces mises à jour de T_{last}^r s'effectuent lorsqu'un robot surveille un noeud (Algorithm 3, ligne 8). Les robots mettent à jour leur liste des dernières dates de visite en conservant le minimum entre les éléments de la liste reçue et ceux de leur liste locale. Les robots ont également besoin des chemins prévus par leurs coéquipiers afin de prévoir leurs positions futures et de calculer le coût de communication présenté dans l'équation (3). Nous voyons que le partage des données est nécessaire pour évaluer avec précision l'oisiveté des noeuds. Par conséquent, il est d'autant plus pertinent de préserver la connectivité du RC qu'elle permet aux données partagées d'être plus précises et plus souvent actualisées, ce qui améliore la pertinence des enchères.

5.5 Résultats

Pour évaluer les performances de notre approche, nous avons pris en compte le maximum et la moyenne de l'oisiveté de tous les noeuds, ainsi qu'une mesure de la connec-

Démonstration

tivité du RC, le nombre de composantes connexes du RC. Toutes ces mesures doivent être minimisées. La figure 3 montre les résultats associés à l'oisiveté maximale et moyenne, pour des instances avec 5 robots, une portée de communication $c_r = 75$, et $\overline{n_{r,w}} = N$ avec $N \in \{2, 3, 4\}$. En d'autres termes, tous les $\overline{n_{r,w}}$ finis ont la même valeur N , qui va de 4 dans les figures 3a et 3d à 2 dans les figures 3c et 3f. Les données sont extraites de 20 simulations d'une durée de 1200 secondes chacune, pour chaque valeur de N , avec et sans inclusion du terme de communication dans la formule d'évaluation des offres. Pour produire les résultats finaux, nous avons calculer la moyenne des résultats de ces simulations. Ces moyennes sont entourées de bandes indiquant l'écart type qui leur est associé. La figure 4 présente les mêmes métriques, pour les résultats obtenus lorsque la mission a été effectuée par 10 robots avec une portée de communication $c_r = 50$.

Le nombre moyen de composantes connexes est inférieur lorsque l'on inclut le terme de communication dans le calcul des enchères. Avec 5 robots, l'allocation effectuée avec notre terme de communication a produit en moyenne 2,95 composantes connexes, tandis que les allocations effectuées sans le terme de communication ont produit en moyenne 3,22 composantes connexes. Avec 10 robots, l'allocation effectuée avec notre terme de communication a produit en moyenne 5,66 composantes connexes, et les allocations effectuées sans terme de communication ont produit en moyenne 6,38 composantes connexes.

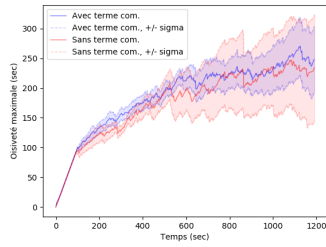
6 Discussion

Lorsque la mission est réalisée par 5 robots avec un portée de communication de $c_r = 75$, l'inclusion du terme de préservation de la communication améliore l'allocation de deux façons. Tout d'abord, on constate que lorsque les pannes surviennent plus tôt dans le scénario, elle permet d'atteindre de meilleure performance, puisque les oisivetés maximum (Figure 3c) et moyenne (Figure 3f) produites par notre méthode se démarque clairement par rapport à celle obtenues sans le terme de communication. Deuxièmement, la variance des résultats est considérablement réduite lorsque l'on utilise le terme de communication : les canaux d'écart type sont plus

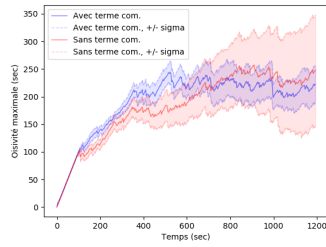
étroits autour des courbes représentant notre méthode, et ce sur tous les graphiques de la Figure 3. Lorsque les pannes se produisent plus tard au cours des missions, deux phénomènes réduisent les effets de l'amélioration de la connectivité du RC. Premièrement, puisque les pannes sont moins probables, le nombre de pannes déclenchées est réduit. De plus, cela diminue la probabilité qu'un seul robot rencontre plusieurs pannes. Les robots ayant rencontrer plusieurs pannes sont particulièrement préjudiciables aux performances de l'allocation, car s'ils se trouvent près des bords de la carte, ils peuvent conserver ces tâches indéfiniment.

Lorsque la mission est réalisée par 10 robots avec un portée de communication de $c_r = 50$, l'inclusion du terme de communication a amélioré l'oisiveté de manière différente. On constate une variance beaucoup plus importante autour des résultats de notre méthode, ce qui est caractérisé par des canaux d'écart-type autour de la courbe de notre méthode beaucoup plus larges dans les graphiques de la Figure 4 que dans la Figure 3. Nous postulons que cela est dû à la réduction de c_r , qui rend les communications très rares, ce qui perturbe la régularité de l'effet du terme de communication. Outre l'augmentation de la variance, les instances avec $\overline{n_{r,w}} = 4$ affichées dans les Figures 4a et 4d ont produit des résultats similaires à ceux des instances avec 5 robots et $\overline{n_{r,w}} = 2$ (Figures 3c et 3f). D'autre part, l'oisiveté maximale dans les instances avec 10 robots et $\overline{n_{r,w}} = 3$ ou 2 présentent une variance beaucoup plus élevée lorsque l'on utilise notre terme de communication lorsque l'on ne l'utilise pas, comme le montrent les Figures 4b, 4c. Cela est dû à la coexistence d'instances très performantes et d'instances moins performantes. En revanche, les mêmes graphiques montrent que lorsqu'on ne tient pas compte du terme de communication, l'oisiveté maximale est constamment élevée, ce qui indique que l'allocation produit beaucoup de mauvaises performances. Ces allocations se caractérisent par le fait que certains noeuds ne sont pas remis aux enchères après qu'un robot a subi une panne en essayant de les surveiller. Ce problème est causé par la très faible valeur de c_r , qui crée de nombreuses positions isolées, c'est-à-dire des positions à partir desquelles il n'est pas possible d'atteindre d'autres robots en raison de la taille

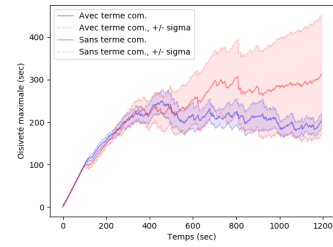
Démonstration



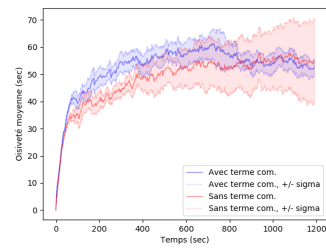
(a) Oisiveté Maximale, $\overline{n_{r,w}} = 4$



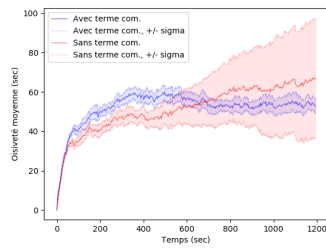
(b) Oisiveté Maximale, $\overline{n_{r,w}} = 3$



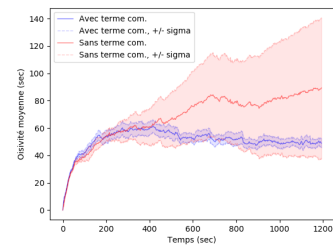
(c) Oisiveté Maximale, $\overline{n_{r,w}} = 2$



(d) Oisiveté Moyenne, $\overline{n_{r,w}} = 4$

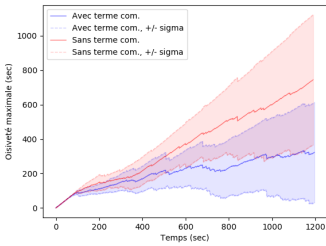


(e) Oisiveté Moyenne, $\overline{n_{r,w}} = 3$

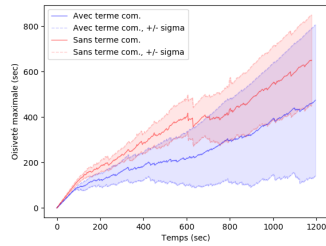


(f) Oisiveté Moyenne, $\overline{n_{r,w}} = 2$

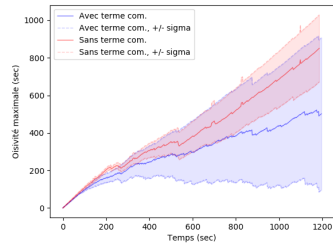
FIGURE 3 – Résultats pour l’oisiveté maximale et moyenne, avec 5 robots et $c_r = 70$. $\overline{n_{r,w}}$ varie de 4 à gauche, 3 au milieu, et 2 à droite. 20 runs ont été moyennés pour obtenir chaque courbe.



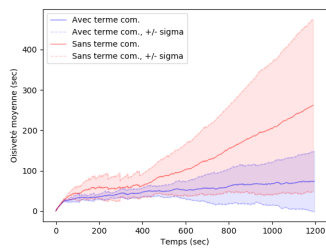
(a) Oisiveté Maximale, $\overline{n_{r,w}} = 4$



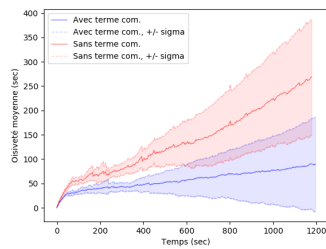
(b) Oisiveté Maximale, $\overline{n_{r,w}} = 3$



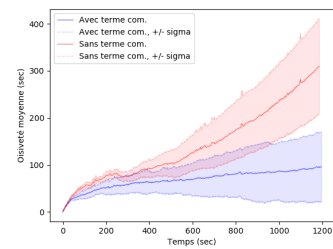
(c) Oisiveté Maximale, $\overline{n_{r,w}} = 2$



(d) Oisiveté Moyenne, $\overline{n_{r,w}} = 4$



(e) Oisiveté Moyenne, $\overline{n_{r,w}} = 3$



(f) Oisiveté Moyenne, $\overline{n_{r,w}} = 2$

FIGURE 4 – Résultats pour l’oisiveté maximale et moyenne, avec 10 robots et $c_r = 50$. $\overline{n_{r,w}}$ varie de 4 à gauche, 3 au milieu, et 2 à droite. 20 runs ont été moyennés pour obtenir chaque courbe.

de la carte. Si un robot lance une enchère pour une de ses tâches interdites à partir d’une position isolée, il ne pourra pas la vendre et sera dans l’incapacité de s’en débarrasser.

L’ensemble de ces résultats peuvent s’expli-

quer par l’amélioration de la connectivité du RC, qui permet aux robots qui rencontrent une panne de revendre rapidement la tâche ayant échoué à un coéquipier. Lorsque l’on ne contrôle pas les communications, il peut arriver qu’une tâche ayant échoué ne soit ja-

Démonstration

mais réattribuée, surtout lorsqu'elle se trouve sur un bord de la carte. Dans ce cas, le robot peut être isolé et donc incapable de mettre la tâche aux enchères. Ce scénario pourrait également se produire lorsque l'évaluation des enchères inclut un terme de communication, mais cela est moins probable, ce qui explique les meilleures performances des allocations produites avec le terme de communication par rapport à celles produites sans le terme de communication.

7 Conclusion

Dans cette étude, nous avons proposé une formule d'évaluation des enchères conçue pour récompenser les allocations améliorant la connectivité du RC d'un SMR résolvant l'ATMR avec un système ESSI. Nous avons produit des simulations numériques démontrant l'utilité de prendre en compte la connectivité du RC, puisque les performances de l'allocation et sa robustesse aux pannes, ont été meilleures en tenant compte de la connectivité du RC.

Notre formule d'évaluation des offres pourrait être encore plus efficace si les robots étaient autorisés à intercaler des tâches dans leur chemin. Cela pourrait permettre aux robots d'adopter des stratégies de coordination d'équipe améliorant le RC, telles que des déplacements groupés, des rendez-vous synchronisés. Pour ce faire, il faudrait concevoir un processus d'évaluation des offres permettant de déterminer rapidement où placer la tâche en délibération, et de produire des évaluations qui tiennent compte de la modification du chemin résultant de ce placement.

Les travaux futurs incluent le développement d'un tel processus d'évaluation des enchères, et la mise en œuvre d'autres ABM que les ESSI, telles que les enchères parallèles single item, et les méthode de consensus. De plus, notre méthode doit être testée dans des scénarios incluant des obstacles et des robots hétérogènes.

Références

- [1] L. E. Parker, "Adaptive heterogeneous multi-robot teams," *Neurocomputing*, 1999.
- [2] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation : A review

of the state-of-the-art," *Cooperative Robots and Sensor Networks*, 2015.

- [3] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination : A survey and analysis," *Proceedings of the IEEE*, 2006.
- [4] C. Poulet, V. Corruble, and A. E. F. Seghrouchni, "Auction-based strategies for the open-system patrolling task," in *International Conference on Principles and Practice of Multi-Agent Systems*, 2012.
- [5] C. Pippin, H. Christensen, and L. Weiss, "Performance based task assignment in multi-robot patrolling," in *Symposium on applied computing*, 2013.
- [6] N. Kalra and A. Martinoli, "Comparative study of market-based and threshold-based task allocation," in *Distributed autonomous robotic systems 7*, 2006.
- [7] M. Otte, M. J. Kuhlman, and D. Sofge, "Auctions for multi-robot task allocation in communication limited environments," *Autonomous Robots*, 2020.
- [8] G. Ferri, A. Munafo, A. Tesei, and K. LePage, "A market-based task allocation framework for autonomous underwater surveillance networks," in *OCEANS*, 2017.
- [9] M. Madhyastha, S. C. Reddy, and S. Rao, "Online scheduling of a fleet of autonomous vehicles using agent-based procurement auctions," in *SOLI*, 2017.
- [10] I. Mezei, V. Malbasa, and I. Stojmenovic, "Auction aggregation protocols for wireless robot-robot coordination," in *International Conference on Ad-Hoc Networks and Wireless*, 2009.
- [11] W. Sheng, Q. Yang, J. Tan, and N. Xi, "Distributed multi-robot coordination in area exploration," *Robotics and Autonomous Systems*, 2006.
- [12] P. Sujit and R. Beard, "Distributed sequential auctions for multiple uav task allocation," in *American Control Conference*, 2007.
- [13] C. Yan and T. Zhang, "Multi-robot patrol : A distributed algorithm based on expected idleness," *International Journal of Advanced Robotic Systems*, 2016.

Réaffectation de tâches de la théorie à la pratique : état de l'art et retour d'expérience

Ellie Beauprez Luc Bigand Anne-Cécile Caron
ellie.beauprez@univ-lille.fr luc.bigand.etu@univ-lille.fr anne-cecile.caron@univ-lille.fr
Maxime Morge Jean-Christophe Routier
maxime.morge@univ-lille.fr jean-christophe.routier@univ-lille.fr

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

Résumé

La problématique de l'affectation efficace de tâches parmi des entités exécutantes est commune à de nombreuses applications réelles. Comme les problèmes d'allocation de ressources ou les problèmes d'appariement, l'affectation de tâches à des exécutants est un problème d'anti-coordination où des agents doivent adopter des plans d'action distincts. Dans cet article, nous proposons un état de l'art des méthodes multi-agents pour l'ordonnancement dynamique de tâches réalisées en parallèle par plusieurs exécutants ainsi que notre retour d'expérience sur leur mise en œuvre.

Mots-clés : Résolution collective de problèmes

Abstract

The problem of efficient task assignment is common to many real-world applications. Like multi-agent resource allocations or matching problems, assigning tasks to workers is an anti-coordination problem. where agents choose distinct plans of action. In this paper, we review the multi-agents dynamic scheduling methods for the simultaneous performance of tasks by several workers and we provide our feedback on their implementation.

Keywords: Distributed Problem Solving

1 Introduction

La problématique de l'affectation efficace de tâches parmi des entités exécutantes est commune à de nombreuses applications réelles pour le calcul parallèle [21, 20], le traitement de données massives [2], l'informatique en nuage [34], la logistique [22, 35, 26], la robotique collective [13, 9, 38, 41] ou les plateformes de *crowd-sourcing* [12]. Depuis les travaux fondateurs de Kuhn [22], il y a plus de 65 ans, de nombreuses variantes du problème d'affectation de tâches ont été étudiées, pour lesquelles des algorithmes

d'approximation et des heuristiques ont été proposés. Pour l'exemple, l'algorithme polynomial de Kuhn-Munkres, appelé méthode hongroise, minimise le coût total de l'affectation de n tâches à n exécutants .

Les systèmes multi-agents (SMA), en tant que paradigme pour la modélisation et l'implémentation de systèmes informatiques visent à résoudre des problèmes difficiles ou impossibles à résoudre pour un système monolithique. Dans les travaux fondateurs tels le *contract net protocol* [11], ou le *distributed monitoring vehicle testbed* [25], l'idée d'une distribution des ressources et des tâches est renforcée par le fait que les problèmes sont eux-mêmes de nature distribuée. Comme les problèmes d'allocation de ressources [8] ou les problèmes d'appariement [28], l'affectation de tâches à des entités est un problème d'anti-coordination où des agents doivent adopter des plans d'action distincts.

Le but de cet article est d'établir un état de l'art des travaux qui traitent d'affectation de tâches en se focalisant sur les problèmes d'ordonnancement où les tâches sont réalisées en parallèle par plusieurs exécutants et sur les méthodes de réaffectation qui adoptent une approche centrée individu. Nous fournissons une grille d'analyse inspirée par [14, 16, 35, 37, 20]. Cette grille nous permet de classer les travaux existants en fonction des problèmes abordés —leurs ingrédients (ressources, exécutants, tâches) et les performances visées. Nous proposons une typologie des méthodes de (ré)affectation en fonction de leurs caractéristiques (décentralisation, dynamisme, modèle, etc.) ainsi que notre retour d'expérience sur leur mise en œuvre dans nos différents prototypes [1, 32, 33, 3, 4].

L'article est structuré comme suit. Tout d'abord, nous évoquons une application à grand succès (cf. section 2). Ensuite, nous présentons les ingrédients des problèmes d'affectation de tâches

1. Pour une implémentation récente, voir [32].

(cf. section 3) en les illustrant à l'aide d'un cas d'étude jouet qui s'inspire de cette application. Nous passons en revue les principales méthodes qui ont été proposées pour cette famille de problème : les méthodes de référence (cf. section 4) et celles qui adoptent une approche centrée individu (cf. section 5). La section 6 synthétise notre grille d'analyse et dresse quelques perspectives.

2 Application à grand succès

Kiva [13] est un SMA utilisé pour la préparation de commande dans les centres de distribution Amazon®. Il automatise la collecte de produits pour des multiples commandes en supprimant les tâches de tri manuel et de réaménagement des entrepôts. Ce système complexe est composé de véhicules autonomes (*drive unit* —DU), qui collecte des *pods* d'inventaire sur lesquels les produits sont stockés. Dans un entrepôt, les *pods* sont initialement placés dans la zone de réserve autour de laquelle se situent des stations d'inventaire (*inventory station* —IS). Les différents produits d'une même commande sont collectés en parallèle par les DU.

Pour répondre aux commandes en temps réel, la planification de la collecte, du réapprovisionnement et du rangement des produits n'est pas le résultat d'une optimisation globale mais obtenu par une heuristique mise en œuvre par des agents DU et des agents IS qui communiquent entre eux et prennent des décisions locales à partir de connaissances limitées. Afin d'assigner les tâches de collecte des produits, l'heuristique sélectionne le DU et le *pod* en fonction de leur localisation et du nombre de produits disponibles sur le *pod*.

Il est important de noter que l'heuristique n'est pas effectivement distribuée mais l'approche centrée individu constitue un cadre conceptuel pour la représentation modulaire du problème de planification.

3 Problème d'affectation de tâches

Nous présentons ici les ingrédients d'un problème d'affectation de tâches à l'aide d'un cas d'étude qui parodie Kiva, c.-à-d. la collecte de colis par des livreurs.

Notre environnement est une grille de cellules, chacune contenant au plus une entité, passive (objet) ou active. Les objets sont des colis de poids différents dispersés et la destination où ils doivent être déposés. Les entités actives sont des

livreurs ou des équipes de livreurs dont la taille détermine le poids maximal des colis qu'elles peuvent transporter. Une entité active peut : (a) se déplacer vers l'une des cases adjacentes ; (b) charger un colis situé dans une cellule adjacente si elle en a la capacité ; (c) déposer un colis si elle est dans une cellule adjacente à la destination.

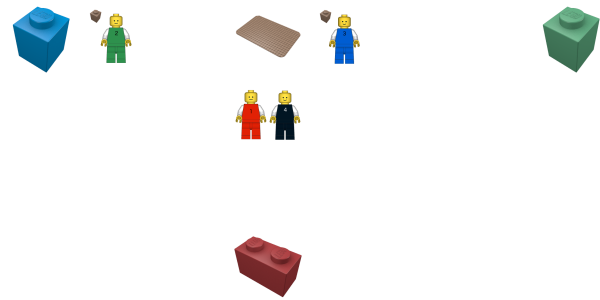


FIGURE 1 – Un colis ciblé par une entité est coloré [31]

Au delà de notre exemple, un problème d'affectation de tâches est caractérisé par les ressources utilisées, le type de tâches et les exécutants à qui elles sont affectées.

Ressources. Afin d'être exécutée, une tâche nécessite généralement l'utilisation de ressources. Leurs propriétés ont des impacts drastiques sur la nature du problème [8]. Ces ressources sont des données, des capacités de calculs, de la mémoire, des objets physiques ou de l'énergie. Elles sont présentes en quantité limitée dans le système, éventuellement munie d'une taille, en un unique exemplaire ou répliquées en plusieurs exemplaires indistinguables. Alors qu'une information est discrète, une ressource énergétique est continue et donc divisible selon une quantité. Les ressources périssables ont une valeur/taille décroissante dans le temps. Lorsqu'elle est utilisée, une ressource est soit consommée par l'exécutant et disparaît ; soit réemployée pour une autre tâche, elle est dite statique. Lorsqu'une ressource est partageable, elle peut être attribuée à plusieurs agents. Une ressource est transférable si elle est utilisable par un autre exécutant que celui qui la possède initialement. Selon la typologie du système, une ressource est locale à un exécutant s'il y accède directement. Alors qu'une ressource non locale et non partageable est inaccessible, Une ressource non locale mais partageable est dite distante.

Dans notre exemple, les colis sont des exemplaires uniques muni d'un poids. Ces ressources sont discrètes, indivisibles, non périssables et consommables. Si une entité ne peut pas trans-

mettre un colis à une autre entité et que l'on considère une équipe comme une seule et même entité alors les ressources ne sont ni partageables ni transférables. Bien que toutes accessibles, la localité des ressources est variable pour les différents exécutants.

Tâches. Qu'elles consistent à explorer un environnement ou à traiter des données, les propriétés des n tâches, qui sont des éléments centraux des problèmes d'affectation, ont également un impact radical sur le problème indépendamment de la méthode de résolution [37]. Qu'elle requiert ou pas des ressources, la réalisation d'une tâche après sa libération vise à produire un résultat, éventuellement avant une date butoir. Comme les ressources, les tâches sont uniques dans le système ou des instances de différents types. Alors que posséder une ressource représente un avantage (c.-à-d. une valeur positive), devoir accomplir une tâche, forcément consommable, représente une charge (c.-à-d. une valeur négative). L'estimation du coût d'une tâche, c.-à-d. son temps d'exécution, est un défi soumis à des incertitudes, des changements et des aléas. Lorsque des conditions préalables doivent être validées pour qu'elles soient réalisables, les tâches sont soumises à des contraintes de précedence induites comme cela peut être le cas dans un procédé de production. Une tâche est préemptive si son exécution peut être suspendue pour être reprise plus tard sans en altérer le résultat, éventuellement avec un surcoût voire par un autre exécutant. Un job désigne généralement une tâche divisible en plusieurs sous-tâches atomiques.

Dans notre exemple, chaque tâche consiste à collecter un colis : atteindre une case adjacente à ce colis, le charger, atteindre une case adjacente à la destination et l'y déposer. Comme certaines actions peuvent échouer (e.g. un déplacement vers une case occupée) ou être non sollicitées (e.g. une entité refoulée par une autre), le coût d'une tâche n'est qu'une estimation du nombre de pas de simulations nécessaires pour son accomplissement. Attendu que les entités peuvent ne porter qu'un colis à la fois et ne le déposer que sur la destination, les tâches sont atomiques, non-préemptives et indépendantes.

Exécutants. Qu'ils soient des robots, des machines ou des nœuds de calculs, les m exécutants doivent être compétents pour tout ou partie des tâches. Même si chaque exécutant a les compétences pour réaliser n'importe quelle tâche, leurs efficacités dans la réalisation des tâches peuvent être homogènes (noté P_m) ou hétérogènes (noté

R_m). Le coût d'une tâche n'est pas nécessairement une propriété intrinsèque, mais elle varie d'un exécutant à un autre si les ressources requises pour une tâche sont inégalement réparties parmi les exécutants. De plus, la topologie est essentielle pour déterminer la faisabilité et la qualité d'une affectation. Le réseau d'accointances (qu'il soit complet, une étoile, une ligne, un anneau, une grille, un arbre ou invariant d'échelle) et les distances entre les exécutants peuvent limiter voire interdire la circulation des ressources et des tâches. D'après la taxonomie de [14], les tâches sont mono-exécutant si chacune d'elles est affectée à un seul exécutant ou multi-exécutants sinon. Réciproquement, les exécutants sont mono-tâche si chacun est affecté à une seule tâche ou multi-tâches si chacun réalise séquentiellement plusieurs tâches.

Dans notre exemple, les entités actives ne sont pas omnipotentes mais une tâche de collecte est réalisable par une entité active ayant la taille requise. Le coût d'une collecte, qui dépend de l'entité exécutante (c.-à-d. de sa position), varie au cours de la simulation. Comme elles peuvent enchaîner les collectes, les entités actives sont multi-tâches. Nous considérons qu'une équipe de livreurs est une seule et même entité et donc, comme dans la suite de cette section, les tâches sont considérées comme mono-exécutant.

En résumé, un problème d'affectation de tâches consiste à les répartir parmi les exécutants en fonction de leur coût.

Définition 1 (Problème). *Un problème d'affectation de tâches à des exécutants est un triplet $\mathcal{P} = \langle \mathcal{S}, \mathcal{T}, c \rangle$ où :*

- $\mathcal{S} = \langle \Omega, \mathcal{V}, \mathcal{R}, \sigma \rangle$ est un système avec m exécutants $\Omega = \{\omega_1, \dots, \omega_m\}$ dans un graphe d'accointances $\mathcal{V} \subset \Omega \times \Omega$ et munis de ressources \mathcal{R} qui leur sont assignées selon une fonction $\sigma : \mathcal{R} \rightarrow 2^\Omega$;
- $\mathcal{T} = \{\tau_1, \dots, \tau_n\}$ est un ensemble de n tâches;
- $c : \mathcal{T} \times \mathcal{R} \mapsto \mathbb{R}_+^*$ est la fonction de coût des tâches.

Le coût d'une tâche pour un exécutant (éventuellement ∞ si elle n'est pas réalisable) se déduit de la localisation des ressources. Même si la plupart des travaux s'abstraient de la répartition des ressources qui est implicite dans la fonction de coût, cette répartition peut guider les heuristiques de (ré)affectation (e.g. [21, 2]).

Affectation. Une affectation de tâches est une répartition parmi les exécutants des tâches selon des lots ordonnés où toutes les tâches sont affectées, ici à un seul exécutant. Formellement,

Définition 2 (Affectation). Soit \mathcal{P} un problème d'affectation de tâches. Une affectation est un vecteur de m lots de tâches $\vec{A} = (\vec{B}_1, \dots, \vec{B}_m)$ où $\vec{B}_i = (B_i, \prec_i)$ est un ensemble de tâches ($B_i \subseteq \mathcal{T}$) affectées à l'exécutant $\omega_i \in \Omega$ ordonné selon un ordre total strict ($\prec_i \subseteq \mathcal{T} \times \mathcal{T}$) tel que $\tau_j \prec_i \tau_k$ signifie que si $\tau_j, \tau_k \in B_i$ alors la tâche τ_j est exécutée avant la tâche τ_k par ω_i . Une allocation \vec{A} vérifie que :

$$\forall \tau \in \mathcal{T} \exists \omega_i \in \Omega, \tau \in B_i \quad (1)$$

$$\forall \omega_i \in \Omega, \forall \omega_j \in \Omega \setminus \{\omega_i\}, B_i \cap B_j = \emptyset \quad (2)$$

Il est important de noter que réduire ce problème à un problème d'allocation de ressources associées à des valeurs négatives consiste à ignorer l'ordonnancement des tâches.

Définition 3 (Ordonnancement). Soit τ la tâche affectée à l'exécutant ω_i à la date de libération (release date) t_τ^0 dans l'affectation de tâches \vec{A} pour un problème \mathcal{P} . Son ordonnancement se caractérise par :

— un **délai d'attente** (delay),

$$\delta(\tau, \vec{A}) = \sum_{\tau' \in B_i | \tau' \prec_i \tau} c(\tau', \omega_i) \quad (3)$$

— une **durée de réalisation** (response time),

$$C_\tau(\vec{A}) = \delta(\tau, \vec{A}) + c(\tau, \omega_i) \quad (4)$$

— une **date d'achèvement** (completion date)

$$t_\tau^E(\vec{A}) = t_\tau^0 + C_\tau(\vec{A}) \quad (5)$$

Le délai d'attente d'une tâche correspond aux temps d'exécution estimés (coûts) des tâches qui la précèdent dans le lot correspondant (Eq. 3). On fait ici l'hypothèse que les exécutants ne sont jamais inactifs. La durée de réalisation de la tâche correspond au temps d'attente avant que la tâche soit entamée plus l'estimation de son temps d'exécution (Eq. 4). Elle dépend donc de la charge de l'exécutant. Contrairement au coût de la tâche, le délai d'attente et donc la durée de réalisation dépendent de l'ordre d'exécution. La date d'achèvement d'une tâche est l'instant auquel elle est terminée (Eq. 5).

Objectifs. Pour évaluer la qualité d'une affectation de tâches, au delà de la fiabilité du système [20] ou de la qualité de service [34], notre étude se focalise sur les mesures de performances envisageables.

Définition 4 (Métriques). Une affectation de tâches \vec{A} pour un problème \mathcal{P} se caractérise par :

- la charge de travail (workload) des exécutants,

$$w_i(\vec{A}) = \sum_{\tau \in B_i} c(\tau, \omega_i), \forall \omega_i \in \Omega \quad (6)$$

- le débit de réalisation (throughput),

$$W(\vec{A}) = \sum_{\omega_i \in \Omega} w_i(\vec{A}) \quad (7)$$

- la durée globale de réalisation (makespan),

$$C_{max}(\vec{A}) = \max_{\tau \in \mathcal{T}} t_\tau^E(\vec{A}) \quad (8)$$

- la durée moyenne de réalisation (flowtime),

$$C(\vec{A}) = \frac{1}{m} \sum_{\tau \in \mathcal{T}} C_\tau(\vec{A}) \quad (9)$$

Tandis que la charge de travail d'un exécutant est une mesure individuelle, les autres métriques sont macroscopiques. Le débit de réalisation est équivalent au nombre de tâches réalisées par unité de temps (Eq. 7). La durée moyenne de réalisation mesure le temps écoulé en moyenne entre la date de libération d'une tâche et sa date d'achèvement (Eq. 9). La durée globale de réalisation représente le temps nécessaire à la réalisation de l'ensemble des tâches (Eq. 8). Elle correspond donc à la charge maximale des exécutants, c.-à-d. celle de l'agent limitant (*bottleneck agent*) :

$$C_{max}(\vec{A}) = \max_{\omega_i \in \Omega} \{w_i(\vec{A})\} \quad (10)$$

Les objectifs [35, 20] qui consistent à minimiser ces métriques sont divers et variés :

- le débit de réalisation (Eq. 7) est une mesure utilitaire de la performance du système du point de vue des exécutants qui est agnostique vis-à-vis de l'ordonnancement ;

- la charge maximale des exécutants (Eq. 8-10) est une mesure égalitaire, du point de vue des exécutants, de l'équilibrage des charges qui est également agnostique vis-à-vis de l'ordonnancement. Cet objectif peut être raffiné en considérant un ordre lexicographique sur les charges, la différence entre la charge maximale et la charge minimale ou la différence entre la charge maximale et la charge moyenne ;

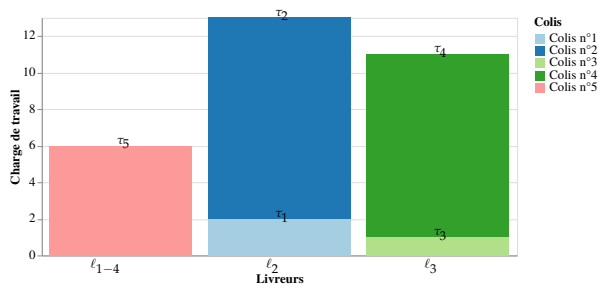


FIGURE 2 – Ordonnancement des tâches pour notre exemple

— la durée moyenne (ou totale) de réalisation est une mesure utilitaire de la performance du système du point de vue des tâches et donc de leur commanditaire. Cet objectif qui consiste à réduire les temps de réponse, c.-à-d. les délais d’attente, peut être raffiné en distinguant les différents types de tâches.

Ces objectifs parfois antagonistes peuvent également être combinés à travers des fonctions multi-objectifs.

Dans notre exemple, les métriques sont $W(\vec{A}) = 30$, $C_{max}(\vec{A}) = 13$ et $C(\vec{A}) = \frac{33}{5}$. L’échange de cible entre les livreurs bleu et vert fait décroître le *makespan* qui approxime (en négligeant l’ordre des collectes) le nombre de pas de simulation c.-à-d. le nombre de pas effectués par l’entité la plus sollicitée au cours de la simulation.

4 Méthodes d’affectation

La théorie de l’ordonnancement [7] propose des méthodes d’affectation hors-ligne. Nous présentons ici les méthodes qui servent de référence (*baseline*) sur les problèmes les plus classiques .

La règle du « plus court processus en premier » – *shortest processing time first* (SPT) – est une méthode très simple qui minimise le débit de réalisation pour un problème avec un seul exécutant multi-tâches et n tâches mono-exécutant [10]. Ce résultat se généralise au problème avec m exécutants multi-tâches si le coût d’une tâche est identique d’un exécutant à l’autre (P_m). Selon cet ordonnanceur, la plus petite tâche est affectée au premier exécutant, la seconde plus petite tâche au second exécutant, . . . , la $(m + 1)^{ime}$ plus petite tâche suit la plus petite tâche dans le lot du premier exécutant, la $(m + 2)^{ime}$ plus petite tâche suit la seconde plus petite tâche dans le lot du deuxième exécutant, etc.

2. Pour une implémentation récente de ces ordonnanceurs, voir [32].

Le problème d’ordonnancement qui consiste à minimiser la durée totale de réalisation (*flow-time*) avec m exécutants multi-tâches et n tâches mono-exécutant dont les coûts dépendent de l’entité exécutante (R_m) peut être formalisé par programmation linéaire —*Linear Programming* (LP). Ce problème se réduit à un problème d’appariement pondéré dans un graphe biparti avec n tâches et $n \times m$ positions. Ce problème est polynomial [17]. Reposant sur l’algorithme de Ford-Fulkerson, la complexité de l’algorithme décrit par [6] est $\mathcal{O}(\max(mn^2, n^3))$.

Le problème d’ordonnancement qui consiste à minimiser la durée globale de réalisation (*makespan*) avec m exécutants multi-tâches et n tâches mono-exécutant dont les coûts dépendent de l’entité exécutante est NP-difficile [18]. Les algorithmes pseudo-polynomiaux connus pour ce problème sont : une heuristique de réaffectation (*earliest completion time* – ECT) [19], des heuristiques en deux phases reposant sur la programmation linéaire [23], des méthodes de recherche locale [15], ou l’algorithme classique de séparation et évaluation [29].

D’une manière générale, les problèmes d’affectation sont des problèmes d’optimisation sous contraintes pour lesquels les résultats obtenus par des solveurs polyvalents tel que IBM® ILOG® CPLEX® sont médiocres. Toutefois, ces problèmes peuvent être résolus de manière approchée par différentes heuristiques comme la montée en gradient, le recuit simulé ou les algorithmes génétiques [30]. L’affectation obtenue avec ces algorithmes d’approximation est acceptable mais les temps d’ordonnancement sont rédhibitoires avec un grand nombre de tâches (plus de 100 000).

Les limites des méthodes d’affectation sont les suivantes.

La réactivité. Les méthodes classiques d’ordonnancement sont majoritairement des processus d’affectation instantanés (*instantaneous assignment* – IA) et non pas prolongés dans le temps (*time-extended assignment* – TA) [14]. Les modèles d’affectation tâches-exécutants pour maximiser le gain des exécutants et des commanditaires sur les plateformes *crowd-sourcing* où les tâches comme les exécutants arrivent en continu (OTA-TSA-*Online Task Assignment with Two-Sided Arrival*) suppose un motif d’arrivée des types de tâches et des types d’exécutants dont les distributions de probabilités sont indépendantes et prédites à partir de l’historique [12]. Toutefois, comme indiqué dans la section 3, l’estimation inexacte des temps d’exécution, agrégée

par des perturbations (consommation/libération de tâches, ralentissement des exécutants, etc.), peuvent nécessiter d'importantes modifications de l'affectation existante pour qu'elle reste optimale. Plutôt que de recalculer en continu une affectation optimale, quelques modifications locales au cours de l'exécution des tâches peuvent améliorer la performance de l'affectation.

Le passage à l'échelle. Un contrôleur global constitue un goulot d'étranglement en matière de performance, car il doit collecter les informations d'état de l'ensemble du système en temps réel. De plus, les problèmes d'affectation sont souvent intractables à cause de la combinatoire des ordonnancements.

5 Méthodes de réaffectation

Le paradigme multi-agents est approprié pour la conception et l'implémentation de mécanismes distribués et adaptatifs de réaffectation dynamique de tâches-exécutants, en particulier les modèles SMA dont la structure est dynamique et dénuée de relation d'autorité [16]. Ces modèles se distinguent de par la nature des tâches et des agents, qu'ils représentent les exécutants ou les commanditaires des tâches (cf. tableau 1).

Modèle	Agents	Tâches
Coalition	coopératifs	multi-exécutants
Équipe	coopératifs	divisibles
Marché	égoïstes	atomiques

TABLE 1 – Modèles pour la réaffectation

5.1 Coalition

La formation d'une coalition se justifie si la réalisation d'une tâche nécessite plus d'un exécutant comme le suggère notre exemple (cf. figure 1) ou si son coût diminue avec le nombre d'exécutants affectés. Le problème de formation des coalitions pour l'affectation de tâches est équivalent à un problème de couverture ou de partitionnement par un ensemble de coût minimal. Ces problèmes sont NP-difficiles, car le nombre de coalitions potentielles est exponentiel.

En particulier, Shehory et Kraus proposent des algorithmes décentralisés gloutons et *anytime* pour l'assignation de tâches via la formation de coalitions [40]. Les auteurs supposent que les tâches sont multi-exécutants et soumises à des contraintes de précédence. Les capacités/efficacités des exécutants sont hétérogènes. Chaque agent est membre d'une seule coalition

et une coalition est mono-tâche. Les algorithmes proposés procèdent en deux étapes :

1. le calcul distribué du coût des coalitions. L'algorithme permet à un agent de choisir l'ensemble des coalitions pour lesquelles il calcule le coût et en informe ses pairs ;
2. une procédure gloutonne, itérative et distribuée où les agents choisissent les coalitions qu'ils préfèrent et les forment. Afin de choisir les coalitions avec les coûts par agent minimaux, chaque agent cherche parmi les coalitions dont il a calculé les coûts celle dont le coût par agent est minimal et l'annonce à ses pairs, puis le groupe sélectionne une coalition. Comme les coalitions doivent être disjointes, les agents impliqués sont exclus de la liste des coalitions potentielles à recalculer.

Le nombre de coalitions envisagées est réduit en limitant la taille maximale des coalitions, car les petites coalitions, moins coûteuses en communications, sont privilégiées.

Il est important de noter que la plupart des algorithmes multi-agents ne sont pas opérationnels. Ils ne sont pas implémentables en l'état mais ils nécessitent d'être reformulés en termes de comportement pour être mis en œuvre [3].

5.2 Équipe

Une équipe est composée d'agents coopératifs travaillant ensemble à la réalisation d'un objectif commun. Comme pour une coalition, une équipe vise à maximiser une fonction objective globale plutôt que la satisfaction individuelle des membres mais les tâches sont mono-exécutant. Chacun muni de compétences qui lui sont propres correspondant aux types de tâches qu'il peut accomplir, les agents communiquent avec leurs voisins dans le graphe d'accointances et jouent un rôle dans la réalisation de l'ensemble des tâches et sous-tâches. Les agents ont accès à une représentation explicite des informations sur les objectifs à accomplir, les autres membres et le plan d'actions commun afin de guider leur comportement et de s'adapter en cas d'évènements imprévus.

En particulier, Lesser et al. proposent un modèle hiérarchique de représentation des tâches pour la coordination qui est indépendant du domaine : affecter les ressources, assigner les tâches et les ordonnancer [24]. L'objectif premier de *Generalized Partial Global Planning* (GPGP) est de maximiser l'utilité combinée de l'ensemble des agents par la réalisation de ses objectifs de plus

haut niveau. GPGP est un modèle de coordination basée sur la planification des activités. La coordination des agents consiste en une recherche coopérative sur un arbre d'activité qui évolue de façon dynamique. Les activités (tâches ou objectifs) et leur relation de dépendance sont représentées dans un arbre de but de type ET/OU. De plus, les relations de dépendance entre les ressources (consommables ou pas) nécessaires à la réalisation des activités sont représentées par un graphe biparti. Le modèle GPGP vise à réduire les comportements incohérents des agents : le déséquilibre de charges, la réalisation de sous-tâches inutiles ou redondantes, voire l'inactivité. Un agent arbitre entre son choix individuel et celui des autres agents quand il a une évaluation biaisée de la solution optimale, car il ne dispose que d'une information partielle. Le modèle GPGP permet de résoudre des problèmes en temps réel nécessitant une gestion des ressources et impliquant des interdépendances complexes entre tâches devant être réalisées par différents agents.

Cette approche suppose que l'effort nécessaire à la coordination (raisonnement et communication) est négligeable par rapport aux temps d'exécution des tâches à coordonner. Dans le cas contraire, l'ordonnancement des tâches est déterminé par le système (distribué) sous-jacent.

5.3 Place de marché

S'inspirant de théorie économique, la « programmation orientée marché » aborde les problèmes de planification distribuée à travers la recherche d'un équilibre pour un jeu non-coopératif [42]. Les agents délèguent des (lots de) tâches voire les échantillons Ces transactions, effectuées via des carnets d'ordre ou des protocoles de négociation comme le *contract net protocol* [11] visent à réaffecter les tâches de façon efficace.

Il est important de noter que, contrairement à une équipe, une place de marché suppose que les contraintes et les objectifs sont complètement distribués. Leur communication est considérée comme une manipulation stratégique qui menace l'équilibre du marché. La « programmation orientée marché » est un patron de conception pour l'ordonnancement distribué où le concepteur définit une configuration de marché, c.-à-d. les ressources, les tâches et les agents (leurs objectifs et leurs comportements) [13, 38]. Par exemple, MASTA est un système multi-agents qui exploite la localité de ressources transfé-

rables pour réaffecter en continu des tâches afin de minimiser la durée globale de réalisation (*makespan*) [2]. Parmi les méthodes multi-agents de réaffectation, on distingue trois familles.

CBBA. L'algorithme à base de consensus (CBBA – *Consensus Based Bundle Algorithm*) [9] est une méthode multi-agents d'affectation en deux phases qui consiste à : (a) sélectionner les tâches à négocier ; (b) déterminer l'agent qui remporte ces négociations. Dans la continuité, Turner et al. étudient l'affectation en continu de tâches à une flotte de robots pour maximiser le débit de réalisation (Eq.7) avec des ressources en carburant limitées [41]. Grâce à l'apprentissage automatique supervisé à partir des exécutions précédentes, les robots choisissent dynamiquement et de manière décentralisée la meilleure heuristique de sélection de tâche.

DCOP. Les problèmes de réaffectation peuvent être représentés sous la forme d'un problème d'optimisation sous contraintes distribué – *Distributed Constraint Optimization Problems* (DCOP). De nombreuses méthodes ont été développées pour la recherche d'une solution optimale à un DCOP qui est un problème NP-difficile (voir [36] pour une synthèse récente). Les principales difficultés dans la mise en œuvre de ces méthodes pour la réaffectation de tâches résident dans :

1. la représentation d'un problème réaliste sous la forme d'un DCOP, voire de plusieurs sous-problèmes COP, nécessite une expertise de la méthode de résolution [26] ;
2. la mise en correspondance de la fonction objectif avec une mesure de performance autre que le débit de réalisation (cf. définition 4). Par exemple, MGM2 suppose que cette fonction est monotone [27].

MARL. Les problèmes de réaffectation peuvent également être modélisés via des processus de décision markoviens [5], en particulier des processus de décision markoviens partiellement observables décentralisés – *Decentralized Partially Observed Markov Decision Process* (Dec-POMDP). L'optimisation d'un Dec-POMDP à horizon fini est un problème NEXPTIME. Les méthodes de résolution approchée ne peuvent être appliquées que sur de très petites instances de problèmes, elles ne passent pas à l'échelle. Au-delà de ces méthodes de planification hors-ligne, l'apprentissage multi-agents par renforcement (MARL-*Multi-Agent Reinforcement Learning*) nécessite une connaissance parfaite de l'environnement et requiert une phase d'apprentissage [39].

6 Synthèse

Dans cet article, nous avons proposé un état de l'art des travaux qui traitent d'affectation de tâches en nous focalisant sur les méthodes multi-agents pour l'ordonnancement dynamique de tâches réalisées en parallèle par plusieurs exécutants. Le tableau 2 synthétise l'ensemble des travaux évoqués ici selon notre grille d'analyse. La partie gauche du tableau discerne les différents problèmes abordés : leurs ingrédients (ressources, tâches, exécutants) et les objectifs visés. La partie droite révèle les caractéristiques de ces méthodes ainsi que les techniques et modèles sous-jacents. La partie supérieure du tableau contient les méthodes d'affectation alors que la partie inférieure les méthodes de réaffectation où l'assignation est dynamique et continue.

Parmi les avantages du paradigme multi-agents, nous avons mentionné la distribution d'heuristiques pour des problèmes intractables à cause de la combinatoire des ordonnancements pour permettre le passage à l'échelle. De plus, l'approche centrée individu constitue un cadre conceptuel autorisant une représentation modulaire des problèmes d'affectation combinant les objectifs antagonistes des exécutants et des commanditaires (notés $C(\vec{A}) \oplus W(\vec{A})$). Les SMAs permettent de prendre en compte la topologie du réseau et la localisation des ressources. Peuvent être abordés des problèmes d'affectation où la réalisation d'une tâche nécessite plus d'un exécutant ainsi que les problèmes complexes où les tâches sont soumises à des contraintes de précédence ou à des ressources localisées qui nécessitent de la coordination. Intrinsèquement réactifs, 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.) sans pour autant reposer sur un historique comme les méthodes d'apprentissage.

Pour aborder les applications réelles, la difficulté réside dans la formulation de systèmes complexes d'assignation tâches-exécutants décentralisés et adaptatifs qui sont divers de par leurs ingrédients et leurs objectifs. La conception des comportements individuels joués de manière asynchrone par les exécutants et les commanditaires doit aboutir à l'émergence d'affectations faisables qui combinent les objectifs des exécutants avec ceux des commanditaires.

Références

- [1] Quentin Baert, Anne-Cécile Caron, Maxime Morge, and Jean-Christophe Routier. MAS4Data - Multiagent systems for analyzing very large data sets. <https://github.com/cristal-smac/mas4data>, 2019.
- [2] Quentin Baert, Anne-Cécile Caron, Maxime Morge, Jean-Christophe Routier, and Kostas Stathis. Stratégie situationnelle pour l'équilibrage de charge. In *Actes des 27ièmes journées francophones sur les systèmes multi-agents (JFSMA)*, pages 9–18. Cépaudès, 2019.
- [3] Ellie Beauprez, Luc Bigand, and Maxime Morge. Musketeers : Scala library of algorithms for coalition formation. <https://gitlab.univ-lille.fr/maxime.morge/musketeers>, 2020.
- [4] Ellie Beauprez and Maxime Morge. SMASTA+ - Scala implementation of the Extended Multi-agents Situated Task Allocation. <https://gitlab.univ-lille.fr/maxime.morge/smastaplus>, 2020.
- [5] Aurélie Beynier, François Charpillet, Daniel Szer, and Abdel-Ilhah Mouaddib. DEC-MDP / DEC-POMDP. In Olivier Sigaud Olivier Buffet, editor, *Markov Decision Processes in Artificial Intelligence*, pages 277–313. Wiley-ISTE, 2010.
- [6] J. Bruno, E. G. Coffman, Jr., and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Commun. ACM*, 17(7) :382–387, July 1974.
- [7] Bo Chen, Chris N. Potts, and Gerhard J. Woeginger. *Handbook of combinatorial optimization*, chapter A review of machine scheduling : Complexity, algorithms and approximability, pages 1493–1641. Springer, 1998.
- [8] Yann Chevaleyre, Paul E Dunne, Ulle Endriss, Jerome Lang, Michel Lemaître, Nicolas Maudet, Julian Padget, Steve Phelps, Juan A Rodriguez-Aguilar, and Paulo Sousa. Issues in multiagent resource allocation. *Informatica*, 30 :3–31, 2006.
- [9] Han-Lim Choi, Luc Brunet, and Jonathan P How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4) :912–926, 2009.
- [10] R.W. Conway, W.L. Maxwell, and L.W. Miller. *Theory of Scheduling*. Addison-Wesley, Reading, MA, 1967.

	Ressources	Tâches	Exécutants	Objectif	Dynamique	Décentralisé	Technique/ Modèle
Kuhn-Munkres [22]	—	n mono-exécutant	R_n mono-tâche	$W(\vec{A})$	✗	✗	LP
SPT [10]	—	n mono-exécutant	P_m multi-tâches	$W(\vec{A})$	✗	✗	heuristique
Bruno et al. [6]	—	n mono-exécutant	R_m multi-tâches	$C(\vec{A})$	✗	✗	LP
ECT [19]	—	n mono-exécutant	R_m multi-tâches	$C_{max}(\vec{A})$	✗	✗	heuristique
OTA-TSA [12]	—	n mono-exécutant	R_m multi-tâches	$C(\vec{A})$ $\oplus W(\vec{A})$	✓	✗	LP + variable aléatoire
Shehory et Krause [40]	—	n multi-exécutant	R_m mono-tâche	$W(\vec{A})$	✓	✓	coalition
GPGP [24]	consommables ou pas	n mono-exécutant	R_m multi-tâches	$W(\vec{A})$	✓	✓	équipe
Kiva [13]	consommables répliquées	n mono-exécutant	R_m multi-tâches	$C(\vec{A})$ $\oplus W(\vec{A})$	✓	✓	marché
MASTA [2]	transférables duplicables	n mono-exécutant	R_m multi-tâches	$C_{max}(\vec{A})$	✓	✓	marché
Turner et al. [41]	limitées	n mono-exécutant	R_m en ligne multi-tâches	$W(\vec{A})$	✓	✓	CBBA + classification
Li et al. [26]	—	n mono-exécutant	P_m multi-tâches	$C(\vec{A})$ $\oplus W(\vec{A})$	✓	✓	DCOP
Schaerf et al. [39]	—	n mono-exécutant	P_m multi-tâches	$W(\vec{A})$	✓	✓	MARL

TABLE 2 – Grille d’analyse des méthodes de (ré)affecation de tâches-exécutants

- [11] Randall Davis and Reid G. Smith. Negotiation as a metaphor for distributed problem solving. *Artif. Intell.*, 20(1) :63–109, 1983.
- [12] John P. Dickerson, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Assigning tasks to workers based on historical data : Online task assignment with two-sided arrivals. In *Proc. of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 318–326. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.
- [13] John Enright and Peter R. Wurman. Optimization and coordinated autonomy in mobile fulfillment systems. In *Automated Action Planning for Autonomous Mobile Robots, Papers from the 2011 AAAI Workshop, San Francisco, California, USA, August 7, 2011*, volume WS-11-09 of AAAI Workshops. AAAI, 2011.
- [14] Brian P. Gerkey and Maja J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The international journal of robotics research*, 23(9) :939–954, 2004.
- [15] A. M. A. Hariri and N. Potts, Chris. Heuristics for scheduling unrelated parallel machines. *Computers & operations research*, 18(3) :323–331, 1991.
- [16] Bryan Horling and Victor Lesser. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19(4) :281–316, 2004.
- [17] WA Horn. Minimizing average flow time with parallel machines. *Operations Research*, 21(3) :846–847, 1973.
- [18] Ellis Horowitz and Sartaj Sahni. Exact and approximate algorithms for scheduling non-identical processors. *Journal of the ACM*, 23(2) :317–327, 1976.
- [19] Oscar H. Ibarra and Chul E. Kim. Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors. *Journal of ACM*, 24(2) :280–289, 1977.
- [20] Yichuan Jiang. A survey of task allocation and load balancing in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(2) :585–599, 2016.
- [21] Yichuan Jiang and Zhaofeng Li. Locality-sensitive task allocation and load balancing in networked multiagent systems : Talent versus centrality. *J. Parallel Distrib. Comput.*, 71(6) :822–836, 2011.
- [22] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2) :83–97, 1955.

- [23] Jan Karel Lenstra, David B. Shmoys, and Eva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46(1-3) :259–271, 1990.
- [24] V. Lesser, K. Decker, T. Wagner, N. Carver, A. Garvey, B. Horling, D. Neiman, R. Podorozhny, M. N Prasad, A. Raja, R. Vincent, and X. Q. Xuan, P. Zhang. Evolution of the GPGP/TAEMS domain-independent coordination framework. *Autonomous agents and multi-agent systems*, 9(1-2) :87–143, 2004.
- [25] Victor R. Lesser and Daniel D. Corkill. The distributed vehicle monitoring testbed : A tool for investigating distributed problem solving networks. *AI Mag.*, 4(3) :15–33, 1983.
- [26] Shijie Li, Rudy R. Negenborn, and Gabriel Lodewijks. A Distributed Constraint Optimization Approach for Vessel Rotation Planning. In *Computational Logistics*, pages 61–80. Springer, 2014.
- [27] Rajiv T Maheswaran, Jonathan P Pearce, and Milind Tambe. Distributed algorithms for dcop : A graphical-game-based approach. In *Proc. of the 17th International Conference on Parallel and Distributed Computing Systems (ISCA)*, pages 432–439, 2004.
- [28] David F. Manlove. *Algorithmics of Matching Under Preferences*. World Scientific, 2014.
- [29] Silvano Martello, François Soumis, and Paolo Toth. Exact and approximation algorithms for makespan minimization on unrelated parallel machines. *Discrete applied mathematics*, 75(2) :169–188, 1997.
- [30] Ethel Mokotoff. Parallel machine scheduling problems : A survey. *Asia-Pacific Journal of Operational Research*, 18(2) :193, 2001.
- [31] Maxime Morge. Répartition des tâches pour la collecte de colis : démonstration. Actes des 27ièmes journées francophones sur les systèmes multi-agents, July 2019. Poster.
- [32] Maxime Morge. ScaMATA : Scala implementation of Multi-agents Task Allocation. <https://github.com/cristal-smac/ScaMATA>, 2019.
- [33] Maxime Morge. ScaSMATA : Scalable Situated Multi-Agent Task Allocation. <https://github.com/cristal-smac/ScaMATA>, 2019.
- [34] Amro Najjar, Olivier Boissier, and Gauthier Picard. Négociation adaptative pour l’acceptabilité des services d’un fournisseur saas. In *Actes des ièmes journées francophones sur les systèmes multi-agents*, pages 85–94. Cépaduès, 2017.
- [35] David W. Pentico. Assignment problems : A golden anniversary survey. *European Journal of Operational Research*, 176(2) :774 – 793, 2007.
- [36] Gauthier Picard. Optimisation sous contraintes distribuée : une introduction au domaine. In *Actes des 26ièmes journées francophones sur les systèmes multi-agents, (JFSMA)*, pages 43–52. Cépaduès, 2018.
- [37] Michael L. Pinedo. *Scheduling. Theory, Algorithms, and Systems. Third Edition*. Springer, 2008.
- [38] Cyril Poulet, Vincent Corruble, and Amal El Fallah Seghrouchni. Travailler en équipe : le choix social appliqué au problème de la patrouille multi-agents. In *Actes des 20ièmes journées francophones sur les systèmes multi-agents*, pages 65–74. Cepadues Editions, 2012.
- [39] Andrea Schaerf, Y. Shoham, and Moshe Tennenholtz. Adaptive load balancing : A study in multi-agent learning. *Journal of Artificial Intelligence Research*, 2 :475–500, 1995.
- [40] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2) :165–200, 1998.
- [41] Joanna Turner, Qinggang Meng, Gerald Schaefer, and Andrea Soltoggio. Distributed strategy adaptation with a prediction function in multi-agent task allocation. In *Proc. of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 739–747, 2018.
- [42] Michael P Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1 :1–23, 1993.

COMOKIT : un environnement générique et modulaire pour analyser les impacts des politiques d'intervention contre l'épidémie de COVID-19

K. Chapuis^a
kevin.chapuis@gmail.com

P. Taillandier^{b,c,d}
patrick.taillandier@inrae.fr

B. Gaudou^e
benoit.gaudou@ut-capitole.fr

A. Brugière^{c,d}
arthur.brugière@ird.fr

A. Drogoul^{c,d}
alexis.drogoul@ird.fr

^aUMR 228 ESPACE-DEV,
IRD, Université de Montpellier, France

^bUR 0875 MIAT,
INRAE, France

^cUMI 209 UMMISCO,
IRD, Sorbonne Université, France

^dUniversité Thuyloi, Hanoi, Vietnam

^eUMR 5505 IRIT,
Université Toulouse 1 Capitole, CNRS, France

Résumé

Depuis le début de la pandémie de la COVID-19, de nombreux travaux ont cherché à modéliser et comparer les impacts des politiques d'intervention non pharmaceutique. Cette volonté s'est cependant heurtée à la difficulté à pouvoir facilement prendre en compte la variété des cultures, des comportements et des contextes dans lesquels se déploient à la fois le virus et les politiques de santé publique. Initié pour soutenir les autorités vietnamiennes dans le déploiement d'interventions à l'échelle locale, l'environnement logiciel COMOKIT, qui repose sur un modèle à base d'agents, a été conçu dès le départ pour être modulaire et facilement déployé sur différents cas d'étude, quelles que soient la disponibilité et la qualité des données démographiques, géographiques et épidémiologiques. Intégrant des outils de construction de l'environnement géographique et de génération de populations synthétiques, COMOKIT permet une modélisation détaillée des individus et de leurs activités à une temporalité fine. Cet article présente son adaptation à la comparaison de l'impact de mesures de confinement dans le centre-ville de Nice.

Mots-clés : simulation à base d'agent, COVID-19, plate-forme GAMA

Abstract

Since the beginning of the COVID-19 pandemic, much work has sought to model and compare the impacts of intervention policies. However,

this willingness has been hampered by the difficulty of being able to easily consider the variety of cultures, behaviours, and contexts in which both the virus and public health policies are deployed. Initiated to support the Vietnamese authorities in the deployment of interventions on a local scale, the COMOKIT software environment, which is based on an agent-based model, was designed initially to be modular and easily deployed on different case studies, regardless of the availability and quality of demographic, geographic and epidemiological data. COMOKIT integrates tools for constructing the geographic environment and generating synthetic populations, and allows detailed modelling of individuals and their activities at a fine temporality. This paper presents its adaptation to the comparison of the impact of containment measures in the city centre of Nice.

Keywords: agent-based simulation, COVID-19, GAMA platform

1 Introduction

L'apparition de la COVID-19 en décembre 2019 a profondément impacté nos sociétés. Face à cette maladie inconnue, les autorités de santé publique ont été contraintes d'expérimenter, en peu de temps, diverses politiques d'intervention, à différentes échelles, sans pouvoir systématiquement anticiper et évaluer leurs impacts. Face à cette double difficulté, la modélisation est très vite apparue comme un outil puissant permet-

tant d'explorer *in silico* une gamme de stratégies d'intervention avant leur éventuelle mise en œuvre sur le terrain. De nombreux modèles ont ainsi été développés et utilisés très vite après l'apparition des premiers cas. Parmi ceux-ci, une majorité repose sur de la modélisation mathématique (modèles SEIR à compartiments comme [2]), et certains ont eu un impact très important sur les choix de politiques d'interventions [1]. Si ces modèles sont particulièrement utiles pour apporter une réponse rapide et globale, ils ne permettent pas de prendre en compte ce qui apparaît aujourd'hui comme un aspect fondamental de l'épidémie, à savoir l'hétérogénéité des individus en termes de relations sociales, de comportements et d'attitudes vis-à-vis de la maladie [9]. Nous affirmons dans cet article que la modélisation à base d'agents offre une alternative capable de prendre en compte cet aspect, et qu'elle permet également de fournir le cadre et les outils appropriés pour construire des modèles qui soient réalistes et généralisables à différents contextes [8].

Ainsi, si [15] propose un modèle à base d'agents qui permet d'étudier l'impact d'interventions, telles que le confinement et la distanciation sociale, sur le nombre de patients hospitalisés, il ne prend pas en compte les aspects sociologiques : la transmission de la maladie ne s'appuie pas sur les foyers, les lieux de travail ou autres lieux de rassemblement, pourtant connus pour faciliter infection et propagation. De plus, la représentation simplifiée de l'espace ne permet pas de modéliser la contamination via l'environnement alors que plusieurs études ont montré que le virus y survit jusqu'à plusieurs heures [6], pouvant ainsi entraîner une contamination et une transmission environnementale, mais aussi des infections nosocomiales [13].

Un autre exemple de modèle à base d'agents est celui proposé par [3]. Ce modèle, dérivé d'un modèle de transmission de la grippe, permet de tester des interventions telles que la fermeture des écoles, l'interdiction de voyager, la distanciation sociale et l'isolement. Utilisé à l'échelle de l'Australie pour réaliser des simulations sur 24 millions d'individus ayant leurs caractéristiques et leur contexte social, il ne prend cependant pas en compte la capacité limite du système hospitalier ou les activités de loisir qui ont pourtant un impact fort sur la transmission de la maladie [7].

Dans le cas de [12], les auteurs se basent sur un modèle épidémiologique générique à base d'agents, doté d'une représentation explicite de l'espace, afin d'étudier les conditions de sortie

de confinement. L'étude au niveau de la France permet de tester différents scénarios et inclut une adhésion variable des agents aux contraintes telles que la distanciation sociale ou le port du masque, sans pour autant représenter la richesse et l'hétérogénéité des pratiques sociales, ni l'impact de la dynamique épidémiologique des pays alentours.

C'est dans ce contexte de foisonnement de modèles et d'urgence à produire des réponses adaptées que nous avons développé l'environnement logiciel COMOKIT (COVID-19 Modeling Kit)¹. Cet environnement repose sur un modèle à base d'agents conçu pour être générique, évolutif et donc applicable à une variété de contextes sociaux et géographiques : en plus de permettre une représentation très fine des politiques d'intervention, il permet par exemple de représenter les activités de loisir ou d'autres activités sociales, de prendre en compte la transmission environnementale, l'impact des zones environnantes non explicitement modélisées, ainsi que la capacité limite de prise en charge hospitalière [10]. COMOKIT combine quatre sous-modèles qui interagissent de façon dynamique : (1) un modèle de transmission de la maladie de personne à personne et via l'environnement, (2) un modèle d'évolution du statut épidémiologique individuel, (3) un modèle de mobilité individuelle et quotidienne à l'échelle de l'heure et du bâtiment et (4) un modèle permettant de représenter les impacts de la combinaison de plusieurs politiques d'intervention.

Cet article présente une vue d'ensemble de COMOKIT : ses outils de reconstruction des données géographiques et des populations synthétiques (Section 2.1), le modèle à base d'agents qui constitue le cœur de cet environnement (Section 2.2) et les outils d'analyse des résultats dont il est doté (Section 2.3). De plus, nous illustrons, au travers d'une application à un quartier de la ville de Nice (France), comment COMOKIT peut facilement être adapté à un nouveau cas d'étude (Section 3).

2 L'environnement COMOKIT

COMOKIT a été initialement conçu pour répondre au besoin du gouvernement vietnamien d'outils capables d'aider la prise de décision sur les choix de politiques d'intervention (confinement, obligation de port du masque, etc.) à l'échelle d'une petite ville ou d'un quartier

1. <https://comokit.org>

Contribution scientifique

(d'environ 10 000 habitants). Cette échelle correspond à la taille de la première zone mise en quarantaine au Vietnam au début de la pandémie (commune de Son Loi dans la province de Vinh Phuc, mise en quarantaine le 13 février 2020). Ce contexte initial explique qu'un élément important qui a guidé le développement de COMOKIT a été d'offrir une grande souplesse dans la gestion des données d'entrée : au Vietnam les données géographiques, démographiques ou de mobilités sont généralement inexistantes ou très lacunaires, contrairement à la France elles sont globalement accessibles. Ainsi, COMOKIT a été conçu pour pouvoir être rapidement appliqué à tout cas d'étude en utilisant le minimum de données en entrée, mais en permettant également d'intégrer des données supplémentaires ou plus détaillées lorsqu'elles sont ou deviennent disponibles. De même, un point fort de COMOKIT est sa modularité : comme illustré dans la Section 3, COMOKIT offre la possibilité à tout utilisateur de redéfinir pour son cas d'étude toutes les fonctions proposées, aussi bien concernant le pré-traitement des données que le contenu du modèle lui-même. COMOKIT est ainsi composé de plusieurs modules permettant à la fois de reconstruire toutes les données nécessaires à son application pour un cas d'étude particulier, de lancer des simulations sur un ordinateur ou une grille de calcul et d'analyser les résultats. Cet environnement comprend :

- un ensemble de fichiers modèles en GAML, le langage de programmation de la plateforme GAMA [14], permettant de recueillir les données spatiales nécessaires et générer la population synthétique avec leur agenda.
- un ensemble de fichiers modèles en GAML correspondant au modèle agent simulant le territoire et l'épidémie. Ce module constitue le coeur de COMOKIT.
- une librairie en Python permettant de générer une exploration exhaustive, l'exécuter sur une architecture de calcul haute performance et d'organiser les données brutes de sortie pour en permettre l'analyse.

Les sections suivantes décrivent les 3 composants de COMOKIT.

2.1 Module de construction et de prétraitement des données d'entrées

Appliquer COMOKIT sur un nouveau territoire requiert de disposer de données sur la localisation et la fonction des bâtiments, et de données

sur la population, telles que la taille et composition des ménages, ainsi que l'âge, le genre et les activités d'une semaine type pour chaque individu. Dans de nombreux cas, ces données sont soit partiellement disponibles soit proprement absentes. COMOKIT fournit 3 outils spécifiques et indépendants pour pallier l'absence de données sur la zone géographique étudiée, les populations qui l'habitent et leurs activités quotidiennes. Dans les trois paragraphes ci-dessous sont décrits le contenu et fonctionnement de ces modules.

Construction des données géographiques. Cet outil demande à l'utilisateur de fournir l'extension spatiale de son territoire d'étude sous la forme d'un fichier shapefile (fichier de données spatiales vectorielles). A partir de cette extension spatiale, COMOKIT télécharge toutes les données OSM (OpenStreetMap) correspondantes et les prétraite de façon à en extraire les bâtiments avec leur fonction et leur nombre de logements. Si les données OSM sont très complètes pour des pays comme la France, ce n'est pas le cas pour beaucoup de pays du Sud. COMOKIT offre dans ce cadre la possibilité de télécharger automatiquement les données Google map correspondant à l'extension spatiale étudiée, de les vectoriser et de les analyser de façon à compléter les données sur les bâtiments disponibles.

Génération de population. COMOKIT fournit deux méthodes de génération de population. La première se base sur des données démographiques (*e.g.* pyramide des âges selon le sexe) mobilisées par le générateur de population Gen* [5, 4]. L'autre méthode est un générateur de population *ad hoc* qui permet de générer une population à partir d'un minimum de données, tel que le nombre d'individus et la proportion d'hommes et de femme.

Génération d'agenda. Cet outil permet, pour chaque individu, en fonction de son âge, de son sexe et de son statut professionnel, de déterminer un agenda d'activités : pour chaque jour d'une semaine type, un individu se voit affecter un ensemble d'activités. Une activité est un tuple comprenant l'heure de début de l'activité, le type d'activité et la liste des personnes partageant l'activité (par exemple, aller au restaurant avec des amis).

2.2 Modèle principal

Échelles spatiales et temporelles. L'unité spatiale de base du modèle est le bâtiment : ces

Contribution scientifique

derniers sont représentés explicitement et constituent les lieux d'activité des agents. Un pas de simulation correspond par défaut à 1 heure. En conséquence, nous avons fait le choix de ne pas simuler les mouvements d'un lieu d'activité à un autre : les individus sont toujours situés dans un lieu d'activité (qui peut être un bâtiment proche ou même un parc extérieur). L'hypothèse sous-jacente est qu'aucune infection ne peut se produire pendant la durée du déplacement.

Entités. L'ensemble des entités constituant le modèle COMOKIT et leurs relations ont été synthétiquement représentées dans le diagramme de classes UML en Figure 1. Par soucis de clarté, les attributs ne sont pas représentés.

Les entités centrales du modèle sont les agents de type *Individual* : ils représentent les habitants de la zone d'étude avec leurs caractéristiques individuelles (âge, sexe, statut professionnel), leur statut épidémiologique (issu d'un modèle classique SEIR), et un ensemble de variables relatives aux interventions politiques (e.g. s'ils ont été testés, s'ils doivent se confiner). Ils effectuent leurs activités quotidiennes (par exemple aller au travail, à l'école, faire des courses) en fonction de leur agenda personnel. Ces agents ont également des attributs en lien avec leurs réseaux sociaux : la listes des individus du ménage et de leurs parents proches, leurs amis avec lesquels ils sont susceptibles d'engager des activités partagées, ainsi que leurs collègues de travail (ou selon l'âge, camarades de classe). Ils ont enfin un bâtiment de résidence et selon leurs activités un bâtiment de travail et/ou d'étude.

Les agents *Building* sont les entités spatiales où les agents *Individual* effectuent leurs activités. Deux types de bâtiments spéciaux ont été définis car ils jouent un rôle particulier : l'extérieur (*outside*) et les hôpitaux (*hospital*). L'agent *outside* représente tous les bâtiments situés en dehors de la zone simulée : il est utilisé pour représenter le fait que des personnes peuvent travailler ou effectuer des activités en dehors de la zone considérée. Pour cette zone, une dynamique de contamination particulière est appliquée. Les agents *hospital* sont le lieu où, dans certaines situations, les agents individuels infectés présentant des symptômes critiques peuvent être isolés, traités et le cas échéant guérir.

Le comportement des agents *Individual* est déterminé en grande partie par leur agenda : cet attribut associe à certaines heures de la journée une activité. Chacune des activités possibles, représentée par un agent *Activity*, spécifie l'unité

ou les unités spatiales dans lesquelles un agent peut effectuer ce type d'activité. Le choix de l'unité spatiale dépend des préférences de l'agent *Individual* encodées sous la forme de poids et qui sont définies en fonction de son âge et de son sexe : pour une activité de loisir, un enfant peut préférer aller dans un centre de jeux tandis qu'une personne plus âgée peut préférer aller au cinéma. Une fois le type de bâtiment choisi, COMOKIT propose 3 méthodes permettant de déterminer un bâtiment spécifique où effectuer l'activité : choix d'un lieu aléatoire uniforme, choix du lieu le plus proche de l'individu ou méthode gravitaire. Dans ce dernier cas, la probabilité de choisir un lieu est une fonction croissante de la superficie du lieu et inversement proportionnelle de sa distance à l'individu.

Nous avons prédéfini les principaux types d'activités classiques dans un environnement social urbain : visiter un voisin (*visiting_neighbor*), travailler (*working*), rester à la maison (*staying_at_home*), étudier (*studying*), rendre visite à un ami (*visiting_friend*). Bien entendu, des activités supplémentaires peuvent également être créées à partir de l'agent générique *Activity*.

Le dernier ensemble d'entités concerne la régulation des comportements sous la forme d'interventions non pharmaceutiques pour lutter contre la propagation du virus. La capacité d'un agents *Individual* à réaliser une activité peut être limitée par l'agent *Authority*. Cet agent gère les différentes politiques qui sont adoptées et représentées par des agents *Policy*. Avant chaque activité, un agent *Individual* demande l'autorisation à l'agent *Authority*. Celle-ci examine toutes les politiques adoptées, pour vérifier qu'elles autorisent cette activité pour cet individu. Ces politiques peuvent se combiner (*CompoundPolicy*) pour former des interdictions limitées (*ForwardingPolicy*) dans le temps et l'espace, à un sous groupe d'agents donné ou à un ensemble dynamique (e.g. *PositiveAtHome* : lorsqu'un agent *Individual* a été testé positif, il doit rester chez lui). A noter qu'un agent *Individual* peut choisir de respecter ou non la régulation définie : cette capacité est représentée par une probabilité de respecter la régulation qui est propre à chaque agent *Individual* et qui peut évoluer durant la simulation.

Dynamiques et processus. La dynamique du modèle peut être résumée en trois processus principaux : la dynamique de l'épidémie, les activités quotidiennes des agents *Individual* et

Contribution scientifique

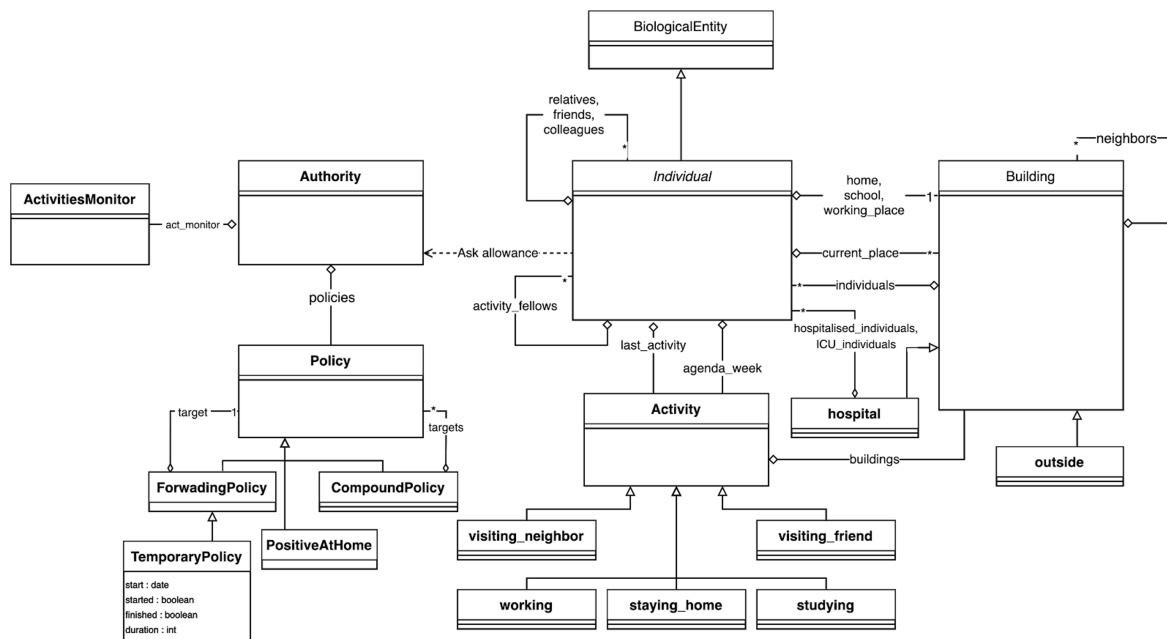


FIGURE 1 – Diagramme de classes UML du modèle COMOKIT

l'adoption et l'application des politiques d'intervention.

Un pas de simulation commence par l'évolution de la charge virale dans les bâtiments (elle diminue au fil du temps, avant de disparaître). Ensuite, les agents *Individual* agissent : ils évaluent d'abord s'ils ont été infectés et, dans le cas où ils sont contagieux, s'ils infectent d'autres agents *Individual* et déposent une charge virale dans le bâtiment où ils se trouvent. Ils mettent ensuite à jour leur statut épidémiologique (voir ci-après) et leur comportement individuel (*e.g.* choix de porter ou non le masque, mise en quarantaine). Enfin, ils exécutent leurs activités quotidiennes : ils trouvent l'activité correspondant à l'heure courante, demandent à l'agent *Authority* s'ils sont autorisés à l'exécuter et agissent en conséquence. Enfin, l'agent *Authority* vérifie sa politique actuelle et essaie de l'appliquer, comme par exemple en exécutant une campagne de test.

L'évolution du statut épidémiologique d'un individu est décrit par un ensemble d'états et de transitions inspiré d'un modèle SEIR : à l'initialisation, nous supposons que la population entière commence la simulation à l'état dit *Susceptible*. En effet, comme il s'agit d'une maladie émergente, personne n'a *a priori* été en contact avec le virus, aucun agent n'est donc immunisé². Le modèle intègre deux voies d'infection différentes

2. L'introduction du virus est un élément de modélisation à part,

pour les agents : soit par transmission d'individu à individu, soit par persistance du virus dans l'environnement. Lorsqu'un agent est en contact avec un autre agent infectieux ou se trouve dans un bâtiment à la charge virale non nulle, il peut être infecté par *Exposition* et passer à l'état *Latent*, en fonction du succès de la transmission. La probabilité qu'un contact avec une surface ou un agent infectieux aboutisse à une exposition effective dépend de la charge virale (*e.g.* la charge virale est *a priori* plus faible chez les agents pre ou a-symptomatiques), du type de contact (*e.g.* plus "intense" par définition entre deux parents qu'entre deux inconnus) et des précautions éventuelles prises par les agents (*e.g.* port du masque, distanciation sociale). Une fois la période de latence écoulée, un individu passe à l'un des trois états *Infectieux* possibles : asymptomatique, pré-symptomatique ou symptomatique. L'individu reste dans ces états pendant l'intervalle sériel pour les pré-symptomatiques et/ou la période infectieuse pour les symptomatiques et asymptomatiques. A l'issue de la période infectieuse, les agents passent systématiquement à l'état *Retiré* : ils ne sont plus infectieux et tombent dans l'un des deux sous-compartiments, *Rétabli* ou *Mort*. Dans le module épidémiologique de COMOKIT, ce dernier sous-compartiment est une conséquence de l'état clinique grave potentiellement développé par les individus symptoma-

soit introduite de manière exogène (*i.e.* bâtiment spécifique *outside*) soit en contaminant *a priori* un ou plusieurs agents.

tiques. En effet l'état *Infectieux* se décompose en sous-compartiment spécifique aux agent développant des symptômes : ils sont a priori placés dans un sous compartiment ne nécessitant pas d'hospitalisation, puis suivant une probabilité d'aggravation sont susceptibles de passer dans le sous compartiment nécessitant une hospitalisation (I_H), puis dans un troisième temps dans le sous compartiment nécessitant une unité de soins intensifs (I_{ICU}). Cette probabilité dépend de la gravité (donnée) de la maladie pour la catégorie d'âge de l'agent et des soins qui lui ont été prodigués ; en effet, si la capacité hospitalière (défini dans les bâtiments spéciaux *Hospital*) permet de traiter les agents du sous compartiment I_H , ils auront relativement au facteur déterminant de l'âge une moindre probabilité de nécessiter une prise en charge en sois intensifs (*i.e.* transition vers l'état I_{ICU}). A l'inverse si la prise en charge est impossible, la probabilité de transition entre ces deux états s'en trouve accrue. Comme souligné plus haut, dans le cas où par manque de capacité de traitement hospitalier, l'agent ne peut pas recevoir les soins intensifs requis par son état, il mourra systématiquement. A l'inverse, tout agent symptomatique qui ne requiert pas de soins intensifs sera systématiquement rétabli à l'issue de la phase infectieuse de la maladie.

Les temporalités ainsi que les conditions de transition entre état sont toutes paramétrables. Elles ont fait par ailleurs l'objet de recherche bibliographique pour en établir les valeurs par défaut en fonction des connaissances disponibles. Pour de plus ample détails sur le modèle, les dynamiques et le choix des valeurs de paramètre, voir [10].

2.3 Module d'analyse des résultats

Exploration du modèle. COMOKIT fournit un ensemble d'outils de visualisation et d'expérimentation permettant de sauvegarder les sorties d'intérêts, tel que l'incidence cumulée, le nombre de patients traités à l'hôpital et en soins intensifs, le nombre de cas réels et connus par voie de test, ou encore le nombre d'individu qu'une personne contaminée va infecter (base désagrégée du taux de reproduction R_0). Ces fichiers de sorties proposent par défaut la valeur de l'ensemble des indicateurs pour chaque pas de simulation, par tranche d'âge de 5 ans et pour chaque type d'activité, totalisant plusieurs dizaines de fichiers au format csv par simulation pouvant avoisiner plusieurs gigaoctets de données brutes par exploration.

Pour explorer le modèle, l'environnement COMOKIT inclut également un ensemble de scripts pour la définition d'exploration exhaustive et leur déploiement sur un supercalculateur. L'ensemble des paramètres du modèle peuvent ainsi être explorés conjointement, en spécifiant pour chacun d'eux l'ensemble des valeurs possibles. La condition d'arrêt des simulations peut également être spécifiée.

Dans [10], nous avons ainsi pu lancer plusieurs centaines de simulations pour déterminer *a priori* un seuil de stabilité des résultats afin de limiter le nombre de réplifications : suite à l'analyse de la dispersion des indicateurs d'intérêts en fonction du nombre de réplification de simulation, il a été déterminé qu'un seuil de 50 constituait le meilleur compromis entre temps de calcul et qualité des résultats.

Post-traitement. Le traitement des données massives (125Mo³ de données au format CSV par combinaison de paramètres) de sortie de l'exploration se fait à l'aide de scripts Python exécutés directement sur le supercalculateur pour limiter les transferts de données. Le post-traitement consiste en une agrégation de toutes les données brutes, puis la compilation et traitement de celles-ci pour produire un fichier CSV synthétique contenant divers indicateurs paramétrables (comme le calcul des quartiles, la valeur moyenne, etc.) et des images de visualisations des résultats (cf. graphiques Section 3.2).

3 Cas d'étude de Nice

3.1 Adaptation du modèle

Dans le cadre de cet article, nous présentons une application pour la ville de Nice en France où l'enjeu est d'utiliser COMOKIT pour mieux étudier l'impact spatiale de politiques de confinement sur la propagation du COVID-19. Cette exemple est une bonne illustration d'adaptation de COMOKIT à un cas d'étude. En effet, par rapport au cas d'étude initial au Vietnam, diverses adaptations de COMOKIT ont été nécessaires : disponibilité de données supplémentaires sur les bâtiments (données IGN), les populations et leurs mobilités (enquête ménage déplacement), meilleure prise en compte de la dynamique épidémiologique à l'extérieur de la zone et nouvelle politique de confinement basée sur

3. 115Ko CSV * 21 CSV * 50 réplifications = 125Mo

Contribution scientifique

la distance des lieux d'activités au logement des ménages.

Une première adaptation concerne l'intégration de données supplémentaires dans la génération de l'environnement. COMOKIT peut récupérer les données sur un cas d'étude à partir des données OSM (qui sont généralement en France de bonne qualité). Néanmoins, nous disposons ici des données IGN (BD Topo) sur la zone d'étude (qui sont plus complètes [11]), nous avons donc modifié la méthode de création des bâtiments sur la zone d'étude de façon à fusionner ces deux types de données, en particulier pour enrichir les données sur le type de bâtiments et le nombre de logements à l'intérieur.

En ce qui concerne la génération de la population, nous avons croisé, dans le cadre de l'application à l'aire urbaine de Nice, la disponibilité de données agrégées sur la composition des ménages, la distribution des attributs individuels et les densités de population (individus et ménages) à l'échelle de la cellule du carroyage INSEE (carreaux de 200m). En mobilisant un algorithme hiérarchique basé sur le principe de reconstruction synthétique [4] disponible dans la librairie Gen*, nous avons pu générer les caractéristiques individuelles, agencer les agents synthétiques en ménage tout en respectant au mieux les marges statistiques concernant leur structure, et localiser les lieux d'habitations sous contraintes du respect du nombre d'individus et de ménages par cellule (échelle spatiale de référence).

Toujours concernant la génération de la population, contrairement au cas d'étude vietnamien, il existe pour la France des données sur les mobilités quotidiennes des personnes : les enquêtes mobilités EMC² 4. Il existe ainsi pour la ville de Nice des données de 2017 sur les déplacements quotidiens d'un petit échantillon de ménages. Nous avons utilisé ces données pour régénérer des agendas pour nos agents *Individual*. Néanmoins, ces informations étant lacunaire (pas de données sur les activités du week-end), nous avons dû proposer une méthode hybride de reconstruction d'agenda utilisant quand cela était possible nos données et dans le cas contraire le générateur de COMOKIT.

Concernant les aspects dynamiques, nous avons modifié la dynamique d'infection en dehors de la zone géographique modélisée. En effet, dans COMOKIT, des agents *Individual* peuvent effectuer toute ou partie de leurs activités à l'extérieur de la zone d'étude (*i.e.* dans le bâtiment

spécifique *outside*) ou au contraire habiter à l'extérieur de la zone d'étude et venir faire une ou plusieurs activités dans l'espace explicitement modélisé. Par défaut, COMOKIT propose une dynamique épidémiologique très simple pour l'extérieur : chaque individu a , à chaque heure passée à l'extérieur, une probabilité fixe d'être infecté. L'idée pour notre application de Nice était de tirer partie des données disponibles sur la propagation de la maladie en France. Ainsi, le modèle ajuste la probabilité de contamination sur les valeurs connues de cas issues des bases de données nationales 5. La probabilité pour un agent d'être contaminée à l'extérieur correspond pour un jour donnée à l'incidence connue sur la zone la plus précise possible (*i.e.* Département des Alpes-Maritimes, Région PACA, France et monde) par rapport à la population totale considérée.

Une dernière modification a concerné la modélisation des politiques d'intervention non pharmaceutique. Nous souhaitons en effet tester l'effet de politiques spatiales, notamment pour n'autoriser que les activités proches du domicile des individus, dans un radiant fixe ou sur une zone urbaine pré-définie.

3.2 Expérimentations réalisées

De façon à évaluer l'impact de différentes politiques de confinement, nous avons simulé la propagation de la maladie dans un quartier de la ville de Nice du 24 janvier au 20 octobre 2020 en prenant en compte le contexte épidémiologique connu sur cette période. Parmi les conditions limites de notre expérience, nous avons choisi de ne pas contraindre le nombre de lits dans les hôpitaux, y compris les unités de soins intensifs, et de considérer que les agents *Individual* ont une attitude passive vis-à-vis de leur propre statut épidémiologique (le fait d'être infecté ne modifie pas leur comportement).

Nous avons testé trois scénarios, dont deux impliquant une intervention : le scénario de base peut être décrit par l'absence de politique d'intervention, les deux seconds se caractérisant d'une part par une politique réaliste inspirée par les actions gouvernementales menées en France et d'autre part par une politique hypothétique de fermeture de zones urbaines. Chacun des deux scénarios suit le développement temporel des mesures prises en France lors du 1er confinement entre le 17 mars et le 11 mai 2020.

5. donnée issue du dépôt officiel à l'adresse suivante : <https://github.com/opencovid19-fr/data>

4. anciennement enquête ménages-déplacements ou EMD

Contribution scientifique

Le confinement "réaliste" implique, à l'image des mesures prises par le gouvernement Français en Mars 2020, une suspension de la plupart des activités : les agents *Individual* ne peuvent pas quitter leur domicile pour effectuer une activité dans un autre bâtiment sauf pour les activités essentielles, à savoir les achats de première nécessité, soins, activités administratives, travail ne pouvant être réalisé à domicile. Concernant cette dernière limitation, nous avons fait le choix de définir un ensemble d'agents *Individual* autorisés à travailler : dans notre cas, nous exprimons cela comme une proportion de personnes qui peuvent agir librement (25%) - l'enquête INSEE Focus n°207 estime que durant le 1^{er} confinement, 45% des français ne travaillaient pas ou peu, alors que 47% étaient en télétravail. Les autres activités ne peuvent être effectuées que dans un rayon donné de 1 km autour du lieu de résidence et pour une durée maximale d'une heure.

La dernière politique testée est un confinement spatialisé. A l'image de la limitation d'un km autour du domicile pour les activités non essentielles dans le cas de la politique d'intervention française, l'idée est de produire des zones de clôture, que l'on ne peut pas quitter mais où toutes les activités sont autorisées, en s'appuyant sur la structure urbaine plutôt que sur une distance arbitraire par rapport au domicile. En effet, cette distance unique ne prend pas en compte le contexte sociale et urbain qui constitue le cadre de vie limite autorisé par les interventions. En proposant un découpage associé à la morphologie et au fonctions urbaines, il est *a priori* possible de délimiter des zones d'activités autorisées susceptibles de fournir l'ensemble des services urbains essentiels (*e.g.* parcs, commerce de proximité).

La zone d'étude choisie est le centre-ville de Nice, qui comprend 39 000 habitants, et qui voit venir chaque semaine près de 58 000 personnes uniques pour travailler ou faire des activités dans cette zone. De plus, plus de 60% des habitants de cette zone urbaine travaillent à l'extérieur.

Pour chaque simulation, 50 répliques ont été exécutées sur le super-ordinateur GAIA d'EDF R&D. Les résultats sur l'évolution du nombre de personnes susceptibles, pré-symptomatiques, asymptomatiques, symptomatiques, hospitalisées, en réanimations et mortes sont respectivement montrées Figure 2.a (pas de confinement), Figure 2.b (confinement réaliste) et Figure 2.c (confinement spatiale). Outre les valeurs de quartiles (ligne) encadrées par les va-

leurs min/max (couleur) sur chaque indicateurs pour chaque jour simulé, les figures montrent le temps d'application des politiques d'intervention (barre grisée) entre le 17 mars et le 11 mai 2020.

On observe sur les figures b et c des politiques de confinement une diminution relative, comparativement au scénario de base, du nombre de personnes nécessitant une hospitalisation, d'entrée en service de réanimation et de morts, ce qui va dans le sens de l'efficacité des politiques de confinement. La diminution s'explique en grande partie par la présence de trajectoire de simulation menant à une quasi absence d'épidémie dans le cas du confinement spatialisé (Q1 avec moins de 5 hospitalisations en soins intensifs et 1 mort) et une épidémie très faible dans le cas du confinement réaliste (Q1 avec moins de 20 hospitalisation en soins intensifs et 6 morts), alors que toutes les simulations dans le cas sans intervention mène à un nombre similaire de mort et d'occupation des soins intensifs.

Cette différence d'amplitude est d'autant plus marquée dans le cas du confinement "spatialisé" (la médiane du nombre de morts est 2 fois moins importante par rapport au confinement réaliste) qu'elle s'accompagne d'un délai plus important dans le déclenchement du pic épidémique. Sur ce point, on peut constater que le scénario sans intervention exhibe également un décalage de plusieurs mois avant d'observer une augmentation de cas significative : l'explication est à chercher du côté du choix fait de ne pas introduire à l'initialisation un cas infecté arbitrairement, mais de s'appuyer sur les données de cas d'infection avérés dans la région ou le département. Ainsi, cette information "contient" en quelque sorte les effets du confinement, puisque la courbe de cas, et donc la probabilité de voir un agent infecté en dehors de la zone y entrer, reste particulièrement basse dans les premiers mois simulés.

L'autre point important qui différencie le confinement spatial et détermine son relatif succès par rapport aux deux autres scénarios, est sa capacité à bloquer le mécanisme d'introduction exogène d'agents infectieux en interdisant durant l'intervention l'entrée d'agents externes. Dans le cas de la politique réaliste, un quart des agents externes, étant associé à des métiers essentiels, sont encore susceptibles d'infecter des habitants de la zone. La mise en place de politiques d'intervention cibler sur l'introduction exogène de nouveau cas (*e.g.* tracer et isoler), loin de contenir entièrement l'épidémie, permet néanmoins d'envisager d'infléchir les courbes de contamination : dans

Contribution scientifique

un cas sur quatre, le rebond épidémique n'intervient qu'à la fin des simulations, *i.e.* plus de 4 mois après la fin de la politique de restriction des activités.

Néanmoins, on observe qu'aucune des interventions modélisées ne permet d'éviter un rebond épidémique, souvent au coeur des débats lors du déconfinement. Cela confirme la nécessité d'introduire dans notre modèle des interventions pharmaceutiques, notamment l'introduction d'une politique de vaccination, afin d'étudier les possibilités de sortie de crise.

4 Conclusion

Nous avons présenté dans cet article l'environnement de modélisation COMOKIT, qui permet de tester l'impact de politiques d'intervention locales face à la pandémie de COVID-19. Une application au centre-ville de Nice, qui permet d'illustrer la versatilité de COMOKIT et le type d'adaptation pouvant être réalisée, a également été présentée. COMOKIT est aujourd'hui utilisé pour des projets de modélisation dans des contextes radicalement différents, ce qui démontre l'intérêt de son architecture et des choix de conception initiaux qui ont été réalisés ; néanmoins, des travaux sont toujours en cours afin d'améliorer sa généralité.

Par exemple, l'une des hypothèses sous-jacente à COMOKIT est que les infections se produisent en des lieux donnés (foyer, entreprises, écoles, etc.) mais pas pendant le déplacement entre ces lieux. Cette hypothèse, valide pour des territoires où la majorité des déplacements sont effectués dans des véhicules individuels (e.g. Vietnam), pose néanmoins question pour des terrains d'étude sans ce type de restrictions ou habitudes. Une extension future de COMOKIT sera de prendre en compte les transports collectifs, de même qu'autoriser la modulation du pas de temps horaire aujourd'hui appliqué par défaut, afin de représenter de façon plus fine des contextes possibles d'infection.

Enfin, l'une des contributions de COMOKIT a été de montrer à quel point l'approche de modélisation à base d'agents était adaptée à la représentation de phénomènes socio-environnementaux complexes. Les extensions prévues, ainsi que celles déjà réalisées par les autres utilisateurs de COMOKIT, en mettant en valeur la versatilité de cette approche, permettent en effet d'enrichir le domaine plus général de la simulation sociale, de plus en plus sollicitée pour répondre aux questionnements des décideurs et aux interrogations

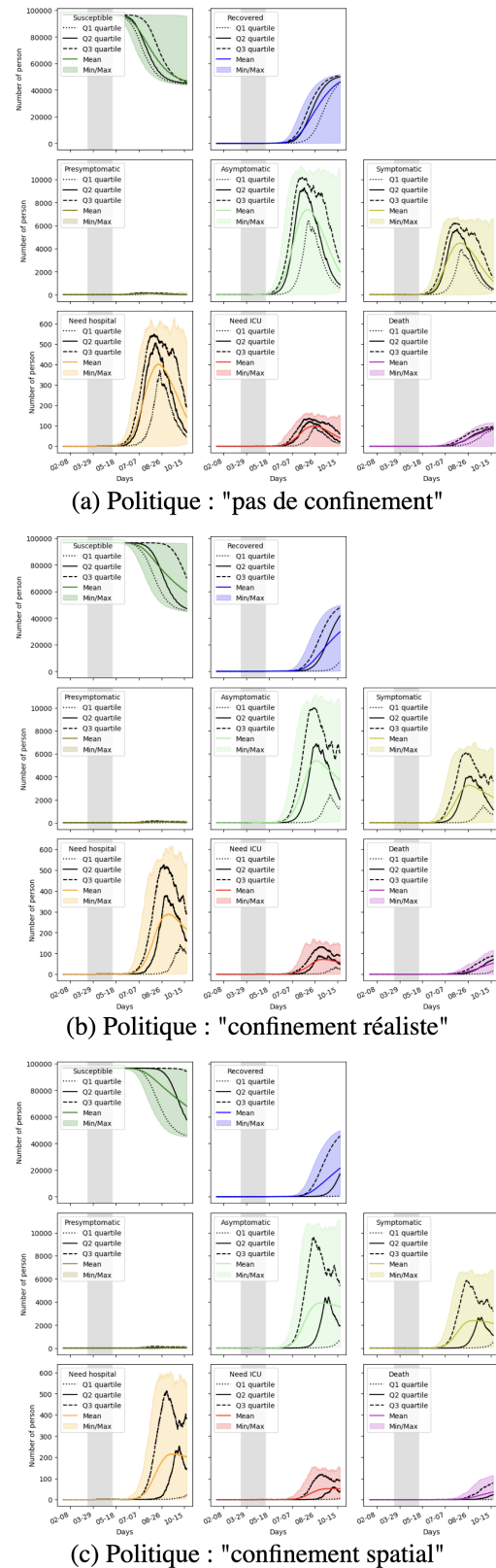


FIGURE 2 – Résultats des simulations pour les 3 politiques

des différentes parties prenantes dans les crises sociétales ou environnementales que l'humanité traverse (adaptation aux changements environnementaux, résilience face aux catastrophes...).

Remerciements

Le travail sur le noyau de COMOKIT est financé par l'ANRS (projet COV15, AAP COVID19-Sud Flash 2020). Les expérimentations à large échelle ont été rendues possibles par l'établissement d'une convention entre EDF et l'IRD pour l'utilisation du super-calculateur GAIA ⁶.

Références

- [1] How epidemiology has shaped the COVID pandemic. *Nature*, 589(7843) :491–492, Jan. 2021.
- [2] D. W. Berger, K. F. Herkenhoff, and S. Mongey. An seir infectious disease model with testing and conditional quarantine. Technical report, National Bureau of Economic Research, 2020.
- [3] S. L. Chang, N. Harding, C. Zachreson, O. M. Cliff, and M. Prokopenko. Modelling transmission and control of the covid-19 pandemic in australia. *Nature communications*, 11(1) :1–13, 2020.
- [4] K. Chapuis, P. Taillandier, B. Gaudou, F. Amblard, and S. Thiriot. Gen* : an integrated tool for realistic agent population synthesis. In *Social Simulation Conference*, 2019.
- [5] K. Chapuis, P. Taillandier, M. Renaud, and A. Drogoul. Gen* : a generic toolkit to generate spatially explicit synthetic populations. *Int. Journal of Geographical Information Science*, 32(6) :1194–1210, 2018.
- [6] A. Chin, J. Chu, M. Perera, K. Hui, H.-L. Yen, M. Chan, M. Peiris, and L. Poon. Stability of sars-cov-2 in different environmental conditions. *MedRxiv*, 2020.
- [7] C. L. Correa-Martínez, S. Kampmeier, P. Kümpers, V. Schwierzeck, M. Hennies, W. Hafezi, J. Kühn, H. Pavenstädt, S. Ludwig, and A. Mellmann. A pandemic in times of global tourism : Superspreading and exportation of covid-19 cases from a ski area in austria. *Journal of clinical microbiology*, 2020.
- [8] A. Drogoul, P. Taillandier, B. Gaudou, M. Choisy, K. Chapuis, Q. N. Huynh, N. D. Nguyen, D. Philippon, A. Brugière, and P. Larmande. Designing social simulation to (seriously) support decision-making : COMOKIT, an agent-based modelling toolkit to analyse and compare the impacts of public health interventions against COVID-19. *Review of Artificial Societies and Social Simulation*, 2020.
- [9] N. Ferguson. Capturing human behaviour. *Nature*, 446(7137) :733–733, 2007.
- [10] B. Gaudou, Q. N. Huynh, D. Philippon, A. Brugière, K. Chapuis, P. Taillandier, P. Larmande, and A. Drogoul. COMOKIT : A Modeling Kit to Understand, Analyze, and Compare the Impacts of Mitigation Policies Against the COVID-19 Epidemic at the Scale of a City. *Frontiers in Public Health*, 8 :563247–563247, 2020.
- [11] J.-F. Girres and G. Touya. Quality assessment of the french openstreetmap dataset. *Transactions in GIS*, 14(4) :435–459, 2010.
- [12] N. Hoertel, M. Blachier, C. Blanco, M. Olfson, M. Massetti, M. S. Rico, F. Limosin, and H. Leleu. A stochastic agent-based model of the SARS-CoV-2 epidemic in France. *Nature Medicine*, 26(9) :1417–1421, Sept. 2020.
- [13] S. W. X. Ong, Y. K. Tan, P. Y. Chia, T. H. Lee, O. T. Ng, M. S. Y. Wong, and K. Marimuthu. Air, surface environmental, and personal protective equipment contamination by severe acute respiratory syndrome coronavirus 2 (sars-cov-2) from a symptomatic patient. *JAMA*, 323(16) :1610–1612, 2020.
- [14] 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.
- [15] Q. Wang, N. Shi, J. Huang, T. Cui, L. Yang, J. Ai, H. Ji, K. Xu, T. Ahmad, C. Bao, et al. Effectiveness and cost-effectiveness of public health measures to control covid-19 : a modelling study. *medRxiv*, 2020.

6. <https://www.edf.fr/groupe-edf/qui-sommes-nous/activites/recherche-et-developpement/toutes-les-actualites-de-la-r-d/la-r-d-d-edf-et-des-equipes-de-l-ird-s-allient-dans-la-lutte-contre-la-covid-19>

L'organisation comme Design Pattern dans les systèmes multi-agents multi-niveaux

V. Sicard^a M. Andraud^b S. Picault^a
 vianney.sicard@inrae.fr mathieu.andraud@anses.fr sebastien.picault@inrae.fr

^aINRAE, Oniris, BIOEPAR, 44300, Nantes, France

^bANSES, Laboratoire Ploufragan-Plouzané-Niort, Unité de recherche en Santé et Bien-être, Ploufragan, France

Résumé

Cet article présente un design pattern destiné à introduire des mécanismes organisationnels dans les systèmes multi-agents multi-niveaux, afin de faciliter la modélisation de systèmes complexes fortement structurés. Le pattern proposé permet de spécifier comment coupler trois niveaux d'agents dans une architecture de simulation multi-niveaux en considérant des contraintes organisationnelles. Nous montrons une application de ce pattern à la modélisation à grain fin de la conduite en bande dans un élevage porcin, où il est employé pour représenter des organisations sociales et physiques. Nous montrons ainsi les avantages de la combinaison de concepts organisationnels et de patterns multi-niveaux pour représenter et simuler des systèmes dynamiques complexes.

[Cet article est adapté de (Sicard et al., 2021).]

Mots-clés : *simulation multi-agents multi-niveaux, design patterns, système organisationnel, systèmes complexes*

Abstract

This paper describes a design pattern aimed at introducing organizational mechanisms into multi-level agent-based simulation architectures, to facilitate the modelling of highly structured complex systems. This pattern makes it possible to specify how to couple any three levels of agents in a multi-level simulation architecture, taking into account organizational constraints. We applied this pattern to the fine-grained modelling of batch management in pig farms, and illustrate how the pattern can be instantiated and composed at several agent levels to reify physical and social organizations. We thus demonstrate the benefits of combining organizational concepts and multi-level patterns to represent and simulate complex dynamic systems.

Keywords: *multi-level agent-based simulation, design patterns, organizational system, complex systems*

1 Introduction

Les systèmes multi-agents multi-niveaux (MLABS, *Multi-Level Agent-Based Systems*) ont fait ces dernières années la preuve de leur intérêt pour modéliser des systèmes complexes (Morvan, 2012) en offrant un cadre de représentation riche et flexible de leur structure. Cette approche étend les systèmes multi-agents (SMA) en fournissant une représentation explicite du niveau macroscopique et de chaque niveau intermédiaire en tant qu'agents dotés de comportements propres. Plusieurs méta-modèles ou architectures ont été proposés pour l'agentification de groupes d'agents à plusieurs échelles (§ 2.1), sans toutefois développer de méthodes spécifiques pour prendre en compte explicitement les caractéristiques organisationnelles qu'on trouve dans certains systèmes complexes, notamment fortement anthropisés. Dans les systèmes naturels, l'organisation est étudiée comme un phénomène émergent qui résulte des interactions entre agents sous-jacents et n'est pas destiné à être introduit en tant que tel dans le modèle. Dans les systèmes conçus par l'homme, l'organisation est souvent un cadre explicite qui a un impact sur le comportement et les interactions entre individus, et doit donc être modélisée de manière explicite.

Les systèmes organisationnels (Ferber and Gutknecht, 1998; Hannoun et al., 2000; Hübner et al., 2009; Weyns et al., 2010) ne sont pas conçus pour s'intégrer à des structures multi-niveaux. Dans ces dernières, l'enjeu est de représenter des groupes d'agents correspondant à des éléments *structurels* imbriqués (e.g. les cellules dans un tissu, un organe, etc.), alors que les systèmes organisationnels, visent à représenter des groupes d'agents selon des caractéristiques ou des contraintes *fonctionnelles*. Les systèmes organisationnels séparent souvent les dimensions physique et sociale, alors que la modélisation d'écosystèmes complexes nécessite un couplage

Contribution scientifique

fort entre elles (Bousquet and Le Page, 2004). Les approches multi-niveaux et les systèmes organisationnels apparaissent ainsi complémentaires pour aborder la modélisation de systèmes complexes.

Notre objectif est d'introduire des mécanismes organisationnels dans une architecture multi-niveaux. Pour ce faire, nous proposons un *design pattern* qui permet de considérer un groupe d'agents à la fois comme une agrégation structurelle d'agents (approche multi-niveaux), et comme une partie structurée prédéfinie d'une organisation, soumise à des contraintes spécifiques. En outre, comme les groupes d'un MLABS sont des agents qui encapsulent un environnement, ce pattern conduit à spécifier la relation entre les dynamiques et contraintes organisationnelles et celles des environnements, sans distinction *a priori* entre environnements physiques et sociaux.

L'article est structuré comme suit : la section 2 analyse les travaux liés aux organisations dans les SMA et les MLABS. La section 3 décrit la nécessité d'une approche organisationnelle multi-niveaux et l'architecture correspondante. La section 4 présente une application du pattern à la modélisation d'un système d'élevage fortement structuré (gestion en bande d'un élevage porcin). Enfin, nous examinons les perspectives de ces travaux pour la modélisation épidémiologique et les SMA en général.

2 Travaux connexes

2.1 Les SMA multi-niveaux

Les MLABS se distinguent des systèmes holoniques (Fischer, 1999; Zhang and Norrie, 1999) ou des architectures récursives telles que SWARM (Minar et al., 1996) par leur capacité à décrire des structures non hiérarchiques, afin de représenter des couplages de niveaux non-arborescents rencontrés dans les systèmes complexes. Outre le nombre croissant d'applications des MLABS, des méta-modèles proposent une agentification de groupes d'agents aux échelles intermédiaires entre des individus atomiques et le système entier (Kubera et al., 2011; Morvan et al., 2011; Drogoul et al., 2013; Camus et al., 2015; Hjorth et al., 2020). Les approches récentes prônent par ailleurs une conception basée sur des *design patterns* (Mathieu et al., 2015; Mathieu et al., 2018) à l'instar d'autres domaines SMA (Juziuk et al., 2014; Klügl and Karlsson,

2009) afin d'améliorer la généricité et la réutilisabilité des solutions opérationnelles développées en réponse à des problèmes conceptuels récurrents. Dans la suite, nous nous appuyons sur le méta-modèle PADAWAN (Picault and Mathieu, 2011), dont le formalisme simple repose sur peu d'hypothèses spécifiques, afin de faciliter la transposition du design pattern que nous proposons à d'autres méta-modèles MLABS. L'aspect multi-niveaux y est représenté par la capacité des agents à encapsuler un environnement. Un agent peut être situé dans un ou plusieurs environnements (sans distinction a priori entre environnements physiques et sociaux), et en même temps encapsuler un autre environnement où peuvent être situés d'autres agents, et ainsi de suite (Fig. 1). Les différents niveaux entre l'ensemble du système et les individus sont représentés par les agents qui, par l'environnement qu'ils encapsulent, hébergent d'autres agents.

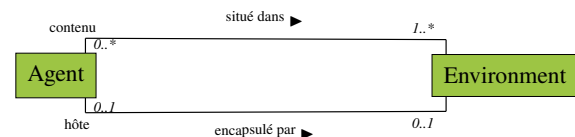


FIGURE 1 – Architecture multi-niveaux. Diagramme de classes montrant les relations entre agents et environnements pour gérer plusieurs niveaux pouvant représenter des structures non-hiérarchiques.

De plus, PADAWAN suppose qu'une matrice d'interaction est associée à chaque environnement pour définir les interactions possibles entre une famille d'agents source et cible dans cet environnement (Kubera et al., 2011). L'élément clef de l'hypothèse sous-jacente à retenir est que les comportements des agents peuvent être spécifiques à chaque niveau.

La principale limite des architectures MLABS, en ce qui concerne les caractéristiques organisationnelles, est qu'elles se focalisent sur les spécifications structurelles des groupes (en tant qu'agents hébergeant d'autres agents via l'environnement) plutôt que sur les spécifications fonctionnelles, sans par exemple la notion de *rôle*. Elles fournissent un cadre pour décrire les couplages entre les agents, les environnements et les interactions, mais assez peu pour les contraintes qui contrôlent les agents et les comportements autorisés dans chaque groupe. Les architectures MLABS sont bien adaptées à la modélisation de systèmes fortement structurés dans lesquels des agents macroscopiques sont construits au-dessus d'agents microscopiques, sous forme d'agrégations, par exemple pour si-

Contribution scientifique

muler des organismes vivants ou le trafic routier. La démarche de conception inverse, qui commence avec des agents macroscopiques et tente de propager les contraintes jusqu'au niveau microscopique, reste difficile.

2.2 L'organisation dans les SMA

La définition de l'organisation donnée par Ferber repose sur trois principes : « (1) le niveau organisationnel décrit le « quoi » et non le « comment » ; (2) aucune description des agents ni de leurs capacités cognitives n'est spécifiée au niveau de l'organisation ; (3) une organisation fournit un moyen de partitionner le système, chaque partition (ou groupe) constitue un contexte d'interaction pour les agents. » (Ferber et al., 2004, traduit) L'organisation peut donc être comprise comme un processus, i.e. un ensemble d'individus avec des contraintes (structure, règles, modèles), ou comme une entité avec ses propres exigences et objectifs (Dignum et al., 2008). L'organisation fournit un cadre pour structurer et gérer les interactions entre les agents, et ajuster leur niveau d'autonomie (Hübner et al., 2009).

Le méta-modèle « Agent-Groupe-Rôle », AGR (Ferber and Gutknecht, 1998), définit les organisations comme un niveau d'abstraction supplémentaire dans un SMA classique, composé de groupes d'agents ayant des objectifs communs ou des tâches communes. Il peut être considéré comme une sorte de cadre dynamique dans lequel les agents sont des composantes. L'environnement (supposé social) représente le contexte de communication entre les agents. Au sein d'un groupe, chaque agent joue un rôle, qui décrit les contraintes (obligations, exigences, compétences) que l'agent doit satisfaire, les avantages (capacités, autorisations, profits) qu'il reçoit, et les responsabilités qui lui incombent. Un agent peut jouer plusieurs rôles et donc se trouver dans plusieurs groupes de la même organisation en même temps. Mais un groupe ne peut être rattaché qu'à une seule organisation. AGRE (Ferber et al., 2005) étend AGR en introduisant un environnement physique. La notion « d'espace » généralise les groupes physiques et sociaux, mais ils restent intrinsèquement distincts (il ne peut y avoir qu'un seul environnement physique mais plusieurs environnements sociaux). Cette approche offre un niveau élevé d'abstraction et établit la base de la structure minimale d'une organisation. Toutefois, la stricte affectation des groupes à une organisation unique et la distinction entre les environnements physiques et sociaux constituent une li-

mite à la représentation des systèmes à structure complexe, car un même groupe ne peut participer à plusieurs organisations. En outre, l'espace dans AGRE n'est qu'un contexte pour un modèle d'activité : il permet de partitionner le système mais pas de représenter la structure ni la dynamique d'un environnement complexe.

Dans MOISE (Hannoun et al., 2000), l'organisation est considérée comme un système de règles qui contraignent le comportement des agents. Ces contraintes correspondent au rôle, i.e. aux spécifications des comportements autorisés d'un agent dans l'organisation. Un groupe est défini par un ensemble de rôles et un sous-ensemble d'objectifs de ces rôles qui peuvent être atteints dans le groupe. La notion de rôle est ici assez similaire à celle d'AGRE, à la différence que les contraintes ne sont pas directement liées à la cohérence du système. Un agent possède un ensemble de contraintes, appelé missions, qu'il doit remplir pour l'exécution d'activités spécifiques. Les groupes sont des compositions d'agents avec leurs rôles et leurs missions, et ne sont pas explicitement liés à la notion d'environnement. Cependant, un groupe, en tant que contexte d'interaction entre agents, peut être envisagé comme un environnement social. Le concept d'organisation se divise formellement en deux : la structure organisationnelle qui est un graphe défini par un ensemble de rôle (sommets) et d'interactions (arêtes) ; et l'entité organisationnelle, qui en est l'implémentation. MOISE permet de définir un système organisationnel et une division de ce système, mais la notion d'environnement structuré et dynamique n'est pas prise en compte.

D'autres approches abordent les organisations d'un point de vue de l'ingénierie logicielle, comme ORA4MAS (Hübner et al., 2009) et MACODO (Weyns et al., 2010). MACODO, lequel est basé sur ORA4MAS est un intergiciel pour l'organisation dynamique des agents en fonction du contexte. Il propose une abstraction de l'organisation en séparant les aspects de coordination et de structure à partir du comportement local des agents. L'environnement représente le contexte logiciel de communication, de perception et d'action entre les agents. Cette approche apporte plusieurs solutions à la problématique de la relation dynamique entre les agents, le niveau organisationnel et l'environnement, mais elle ne fournit pas de concepts explicites pour décrire la dynamique des organisations.

Les approches organisationnelles, élaborées dans un cadre SMA classique, ne prennent pas en compte les aspects multi-niveaux, comme la

Contribution scientifique

dynamique de l'environnement, la dynamique structurelle des groupes et le couplage entre ces différents éléments. La modélisation de systèmes fortement structurés exige une représentation explicite des interactions entre les éléments atomiques et leur environnement, mais également entre les agents et chaque niveau. Dans ces différentes approches, il n'est pas possible de représenter de manière simple la dynamique des environnements et le couplage entre les groupes et les organisations, particulièrement dans un contexte multi-niveaux.

C'est pourquoi nous proposons dans ce qui suit un design pattern destiné à introduire des caractéristiques organisationnelles dans les MLABS. L'objectif de cette approche est de pouvoir représenter des couplages propres aux organisations entre des agents appartenant à des niveaux hétérogènes, de décrire les dynamiques des environnements, et d'exprimer les contraintes des organisations (adhésion, exclusion...).

3 Le pattern organisationnel multi-niveaux

L'objectif d'un design pattern est de fournir une solution générique, réutilisable et modulaire à un problème spécifique (Gamma et al., 1994). Ainsi, des patterns structurels ont été définis pour décrire les couplages entre deux niveaux (Mathieu et al., 2018). Indépendamment de ces choix structurels, il s'agit ici de permettre l'introduction d'une représentation explicite de relations organisationnelles dans les MLABS, ce qui suppose notamment de faire des groupes et de l'organisation elle-même des agents.

Représenter explicitement une organisation dans les MLABS suppose trois critères : 1) exprimer et formaliser la relation structurelle entre les agents ; 2) contraindre les comportements et les interactions entre les agents (via la notion de rôle, au moins implicite) ; 3) participer au contrôle de l'environnement structurellement et fonctionnellement.

Le pattern doit également être compatible avec les modèles classiques MLABS, en particulier par rapport au groupement d'agents au sens multi-niveaux. Ainsi, l'appartenance à un groupe n'implique pas nécessairement d'être membre d'une organisation particulière, par contre l'adhésion à une organisation implique l'appartenance à un groupe.

Comme les méta-modèles MLABS ne préjugent pas de ce que représente chaque niveau, les ca-

ractéristiques organisationnelles visées peuvent concerner des agents de niveaux a priori quelconques dans le système, en fonction du domaine d'application. La transposition de concepts organisationnels dans les MLABS via un design pattern est donc pertinente, puisqu'elle ne demande pas de transformer tout le système mais seulement d'identifier les agents rattachés à chaque organisation et de spécifier les relations appropriées entre les niveaux, i.e. comment les agents et les environnements sont liés, avec quelles contraintes et quelle dynamique.

3.1 Structure du pattern organisationnel multi-niveaux

Le pattern organisationnel décrit la structure et la dynamique relationnelle entre trois niveaux : organisation, groupes et atomes (Fig. 2).

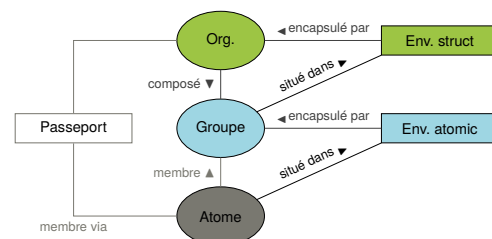


FIGURE 2 – Structure du pattern organisationnel multi-niveaux. L'agent Organisation encapsule un environnement structuré qui est composé d'agents groupes, lesquels encapsulent des environnements atomiques. Les agents impliqués en tant qu'Atomes appartiennent à une organisation par leur localisation dans un environnement atomique. Le Passeport sert de médiateur entre un atome et son organisation.

L'organisation est une agentification d'un environnement structuré, i.e. un environnement qui peut être divisé en sous-environnements (« environnements atomiques »). L'atome représente tout agent membre d'une organisation : l'appartenance à une organisation résulte de la localisation de l'atome dans un environnement atomique, selon les contraintes associées. L'environnement atomique contient de l'information et est encapsulé par un agent Groupe qui gère la dynamique environnementale. Les Groupes sont eux-mêmes situés dans l'environnement structuré encapsulé par l'agent Organisation, qui gère ainsi la dynamique des environnements et assure sa propre cohérence via des contraintes. Comme dans (Ferber et al., 2005), l'organisation est un cadre dans lequel les agents se comportent. Les informations relatives à l'appartenance d'un atome à une organisation (e.g. la localisation de l'atome, l'état de l'atome vis-à-vis

Contribution scientifique

des contraintes, etc.) sont stockées dans le passeport. Nous détaillons ces éléments ci-dessous.

3.2 Les atomes et leurs états

Un atome correspond à tout agent hébergé par une organisation et qui est soumis aux contraintes de celle-ci. Le terme « atome » signifie que nous ne considérons pas sa structure sous-jacente (il peut, ou non, encapsuler un environnement dans lequel d'autres agents peuvent être situés, etc.).

Les atomes sont situés dans un environnement atomique où ils peuvent interagir avec d'autres agents atomiques : tous appartiennent à un groupe, c'est-à-dire l'agent qui encapsule l'environnement considéré. L'atome peut agir dans cet environnement, i.e. prendre ou déposer de l'information (§3.5).

Comme tout agent, l'atome possède des états qui changent en fonction de son comportement (« états réels »). Ces états réels peuvent violer des contraintes de l'organisation. L'organisation peut alors décider soit que cette violation est interdite (et exclure l'atome), soit que l'atome peut être malgré tout considéré comme respectant temporairement les contraintes (et garder l'atome). Dans ce dernier cas, l'organisation passe outre la divergence entre l'état réel de l'atome et l'état attendu en attribuant un « état nominal » à l'atome. Pour préserver l'autonomie des agents, l'état nominal ne peut être imposé directement à l'atome, il est donc consigné dans une structure de données spécifique appelée « Passeport » (§3.3) : l'état nominal d'un agent pour une organisation relève donc de l'interprétation que fait l'organisation des états réels, et non d'un état intrinsèque à l'atome.

3.3 Organisation, contraintes, passeport

L'agent organisation est la concrétisation du niveau abstrait Organisation, dotée de ses propres comportements et états. Une organisation encapsule un environnement structuré qui est divisé en sous-environnements, eux-mêmes encapsulés par des agents Groupes, où les atomes sont localisés (Fig. 2). L'organisation doit vérifier que chacun de ses membres (les atomes) respectent les contraintes qui définissent son intégrité. Si ce n'est pas le cas, elle décide soit d'exclure les contrevenants, soit de reconsidérer comment elle les perçoit en leur attribuant un état nominal qu'elle place dans leur passeport.

Les contraintes permettent à l'organisation d'admettre des atomes et de les localiser, ou de les rejeter. Elles constituent un ensemble de règles concernant les états réels ou nominaux des atomes, auxquelles ceux-ci doivent se plier pour entrer ou rester dans une organisation, et les actions associées au respect ou à la violation de ces règles.

Le passeport constitue un pattern médiateur entre une organisation et un atome. Il stocke des informations sur la localisation et les états nominaux, pour représenter le point de vue de l'organisation sur l'atome, à un temps donné. Le passeport est la propriété d'un atome mais n'est géré que par le système organisationnel (agent Organisation, Fig. 2). Le passeport est composé de deux éléments : l'historique des localisations de l'atome, et le visa (état nominal de l'atome en fonction d'une organisation). Si un atome appartient à plusieurs organisations, il doit disposer d'un passeport répondant à chacune, sans préjuger de l'implémentation.

3.4 Rôle

Les rôles sont considérés comme des comportements abstraits que les agents peuvent manifester au sein d'un groupe (Ferber and Gutknecht, 1998). Comme dans le modèle PADA-WAN (Kubera et al., 2011), nous supposons que les comportements que les agents peuvent adopter dans un environnement donné sont spécifiés par une matrice d'interaction. Cette matrice est une fonction qui détermine les interactions possibles entre une famille source et une famille cible. Les familles sources ou cibles peuvent être, selon l'application, soit un agent spécifique, soit un nom arbitraire tel qu'une classe d'agents, soit encore une valeur particulière d'un état réel ou nominal. Les rôles sont alors définis simplement une combinaison de comportements spécifiés dans les matrices d'interaction.

3.5 Environnement, environnement structuré et groupe

En toute généralité, les environnements sont définis formellement selon (Mathieu et al., 2015) comme un espace doté de deux fonctions : placer des agents et fournir de l'information. Une telle notion de l'environnement permet de représenter indifféremment un espace physique ou social. De plus, les environnements ont leur propre dynamique liée à leur topologie (voisinage, transmission d'informations, etc.).

Contribution scientifique

Un environnement structuré est composé de sous-parties (environnements atomiques) organisées selon des relations topologiques spécifiques et animées d'une dynamique propre. La représentation topologique repose sur un graphe dont les sommets sont les environnements atomiques (qui portent de l'information) reliés par des arêtes pondérées (représentant les flux d'information) qui décrivent la dynamique de l'environnement (Fig. 3).

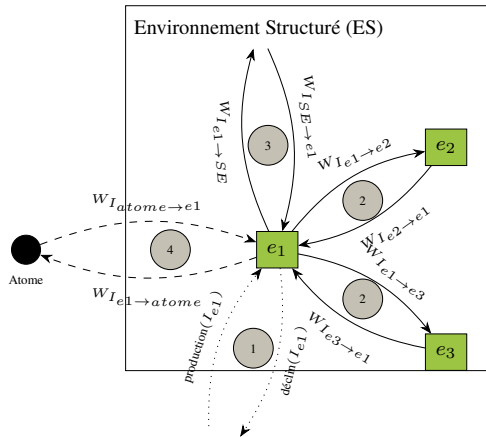


FIGURE 3 – Représentation de la dynamique d'une information I dans un environnement atomique e_1 , résultant des flux entrants/sortants (pondérés par des valeurs $W_{I_{e \rightarrow e'}}$) provenant 1) de la dynamique intrinsèque (sources/puits), 2) des échanges avec d'autres environnements atomiques voisins (e_2, e_3) ou 3) avec l'environnement structuré ES , et 4) des actions effectuées par les atomes situés dans e_1 .

Considérer un environnement comme atomique relève du point de vue du pattern, au sens où les agents concrets du système qui doivent instancier les notions d'Organisation, de Groupe et d'Atome dépendent de la partie du sous-système concernée par la problématique organisationnelle. Ainsi, l'environnement vu comme atomique à un niveau (groupe ou atome) peut lui-même être structuré de façon à implémenter le pattern de façon récursive. De même, un agent considéré comme atomique du point de vue d'une organisation, peut être vu à son tour comme une autre organisation selon une autre instance du pattern (Fig. 4 & 5).

4 Application à la modélisation de systèmes d'élevage complexes

Les modèles épidémiologiques mécanistes permettent de comprendre et de prédire la propagation d'agents pathogènes à différentes échelles (de l'individu au territoire) selon différents scénarios

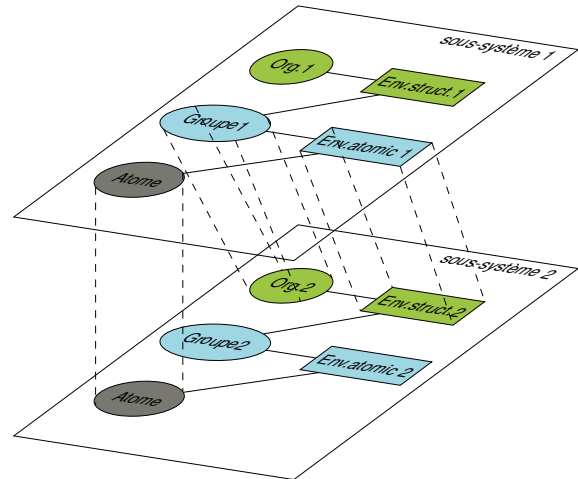


FIGURE 4 – Composition du pattern articulée sur le Groupe. Le niveau Groupe du sous-système 1 est lui-même structuré en organisation. L'Atome est le même dans les deux sous-systèmes.

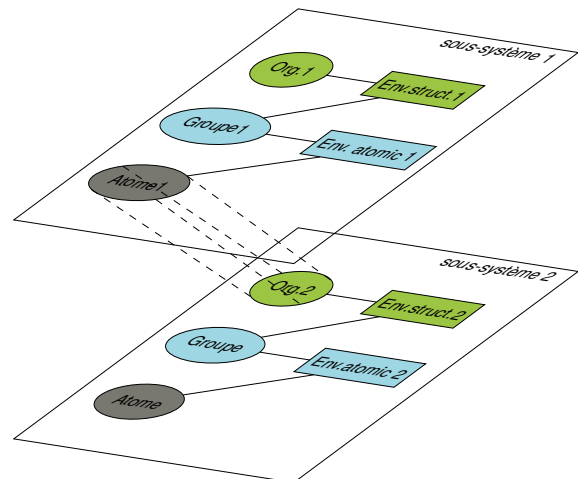


FIGURE 5 – Composition du pattern articulée sur l'Atome. Le niveau Atome du sous-système 1 se décline comme une Organisation dans le sous-système 2.

(Keeling and Rohani, 2008; Ezanno et al., 2020). La prise en compte de la complexité des pathosystèmes (structuration des populations et des environnements, dynamiques, contraintes...) est difficile, mais nécessaire pour rendre les modélisations réalistes, d'identifier les mécanismes impliqués et les leviers d'action possibles.

Le framework open source EMULSION¹, dédié à la modélisation épidémiologique mécaniste stochastique, a montré que les MLABS organisés à partir de design patterns (Mathieu et al., 2018) constituaient un cadre pertinent pour cette problématique (Picault et al., 2019). C'est donc as-

1. <https://sourcesup.renater.fr/www/emulsion-public/>

Contribution scientifique

TABLE 1 – Conduite type d'un élevage porcin en 7 bandes avec intervalle de 21 jours (Chambre d'Agriculture de Bretagne, 2010). Le temps passé dans un secteur pour chaque bande est calculé de manière à optimiser l'occupation des salles.

	Secteur Insémination	Secteur Gestation	Secteur Maternité	Secteur Post-sevrage	Secteur Engraissement
Stade physiologique	Insémination	Gestation	Allaitement	Post-sevrage	Engraissement
Nombre de bandes à loger	2	4	2	3	6
Entré dans les mêmes salles tous les ... (jours)	35 ou 42	77 ou 84	35 ou 42	56 ou 63	119 ou 126
Durée d'occupation (jours)	35	77	28	4 × 61 ou 3 × 54	6 × 114 ou 1 × 121
Temps disponible pour le nettoyage, la désinfection et le vide sanitaire (jours)	5 × 4 ou 2 × 1	7 et 1	14 et 7	2	5

sez naturellement que nous avons choisi d'adapter ce framework pour implémenter le pattern dans l'architecture multi-niveaux et en réaliser une preuve de concept en santé animale.

4.1 Application à la conduite en bande

Dans ce qui suit, nous montrons comment le pattern organisationnel multi-niveaux peut-être appliqué pour modéliser la conduite en bande en élevage porcin français, ainsi que sa plus-value. Les porcs sont élevés en bandes afin de garantir une évolution homogène des états physiologiques des animaux : gestation et allaitement pour les truies et engraissement pour les porcelets. Cela implique que les bandes doivent être cohérentes, i.e. que tous les animaux doivent être dans le même état physiologique au même moment. Les animaux, selon leur type (truie ou porcelet), leur état physiologique (âge et stade de reproduction), et leur appartenance à une bande, sont placés dans des espaces spécifiques (secteur, salle) qui sont des environnements. Ce type de conduite illustre notre problématique :

- l'environnement physique est fortement structuré (secteur, salle, portée, case) ;
- l'environnement physique possède sa propre dynamique (accumulation, propagation, diffusion et décroissance d'agents pathogènes) ;
- les bandes (environnement social) doivent conserver leur cohérence (critères d'homogénéité) ;
- les agents peuvent avoir différents statuts (réel : la valeur effective, et nominal : selon la valeur attendue dans la bande) pour un même état (appartenance à une bande, stade physiologique, localisation, etc.) en fonction du contexte ;
- les environnements sont étroitement liés entre eux (couplage entre bande et localisation physique) via différents niveaux

de regroupement d'agents.

Nous nous basons sur une structure d'exploitation « type » (Salines et al., 2020), composée de cinq secteurs correspondant aux différents stades physiologiques : 1) insémination, 2) gestation, 3) maternité (truies et porcelets ensemble pour l'allaitement), 4) post-sevrage (porcelets séparés de leur mère), 5) engraissement (porcs engraisés avant envoi à l'abattoir) (Chambre d'Agriculture de Bretagne, 2010). Le nombre de bandes détermine le type de conduite (logement et calendrier). Dans notre exemple, nous considérons une gestion en 7 bandes décalées de 21 jours. L'évolution temporelle d'une bande dépend du temps passé dans un secteur, correspondant aux stades physiologiques (Tableau 1). Avant d'aborder la propagation d'agents pathogène et son contrôle, notre objectif est de simuler la conduite et d'observer le comportement global du système et l'évolution des animaux dans les bandes, tant pour les aspects sociaux (bandes) que physiques (hébergement).

Un animal correspond à un atome, et nous considérons deux organisations principales : une pour les bandes (sociale) et une pour l'hébergement (physique). L'organisation « bande » se décompose en plusieurs portées (une truie et « ses » porcelets, naturels ou adoptés, par portée). Chaque portée correspond à un environnement atomique (social) encapsulé par un agent groupe. La portée est liée à l'organisation par un environnement structuré, lequel reflète les relations entre les environnements atomiques. L'organisation « hébergement » est décomposée en plusieurs secteurs (environnements atomiques). Ces environnements atomiques sont des organisations représentant la structuration d'un secteur en salles (Fig 6).

L'évolution des stades physiologiques est fonction du temps et la durée spécifique de chaque étape est directement contrôlée par l'atome et

Contribution scientifique

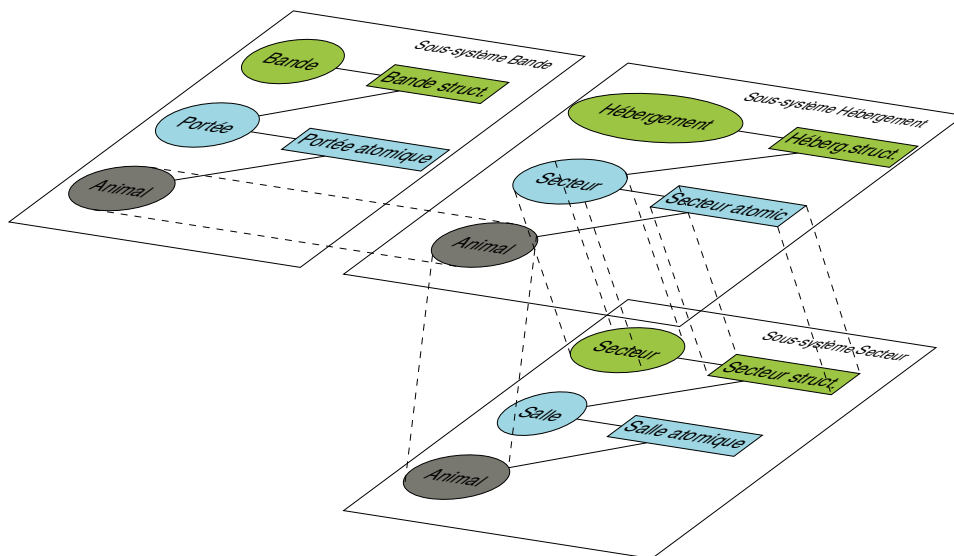


FIGURE 6 – Composition de trois instances du pattern organisationnel multi-niveaux pour modéliser la conduite en bande d'un élevage porcin. Deux organisations principales : Hébergement (organisation physique), et Bande (organisation sociale). L'organisation Secteur est un « zoom » du niveau Secteur d'Hébergement pour décrire les contraintes spécifiques à la gestion d'un Secteur.

dépend de son type (truie ou porcelet).

La localisation des atomes dépend de leurs états (stade physiologique, type, bande) et des contraintes qui définissent le couplage entre les organisations « bande » et « hébergement » :

- la localisation dans un secteur dépend de l'état physiologique
- tous les animaux d'une même bande sont localisés dans une même salle
- tous les animaux d'une même portée sont localisés dans le même lieu

Par exemple, pour être localisés dans le secteur maternité, les animaux peuvent être soit une truie soit un porcelet et doivent être dans l'état allaitement. Certains animaux de différentes bandes peuvent potentiellement satisfaire ces contraintes et être localisés dans le même secteur au même moment, c'est pourquoi les bandes sont réparties dans différentes salles (une bande par salle). La gestion des bandes est optimisée en fonction de l'occupation des salles (Chambre d'Agriculture de Bretagne, 2010).

Pour la preuve de concept du pattern organisationnel, nous avons représenté l'occupation des secteurs et des salles sur une durée de 452 jours. Du fait du décalage de 21 jours, nous avons commencé par la première bande en allaitement, et calculer à partir de là les démarrages des bandes suivantes (deuxième bande en gestation depuis 21 jours, troisième bande en gestation depuis 42 jours, etc.). L'élevage est composé de 15 truies par bande (i.e. 105 truies en tout).

Les résultats de la simulation (Fig. 7) sont conformes à la littérature (Chambre d'Agriculture de Bretagne, 2010) et aux pratiques de terrain. Le pattern organisationnel multi-niveaux est efficace pour représenter, modéliser et simuler à travers un cadre homogène les spécificités de la conduite en bande, il permet à la fois d'observer le comportement spatial (localisation) et de suivre l'aspect social selon différents niveaux, de la bande aux individus. Ainsi, la capacité de modélisation multi-niveaux prend en compte les aspects précis des interactions entre les agents, qu'il s'agisse d'atomes, de groupes, d'organisations ou des environnements encapsulés.

5 Conclusion et perspectives

Nous proposons un pattern organisationnel pour étendre la capacité des MLABS à représenter et mettre en œuvre des relations ou des contraintes dynamiques entre les agents, les niveaux d'organisation et les environnements. Cette proposition se présente sous la forme d'un design pattern qui peut être réutilisé et adapté à d'autres architectures MLABS, et qui peut être appliqué à plusieurs sous-systèmes dans le cas de structures très complexes (sous-organisations ou organisations concurrentes). En outre, elle repose entièrement sur le fait que toutes les entités d'un MLABS sont représentées par des agents, ce qui favorise l'homogénéité structurelle du système.

Ce modèle a été aisément implémenté dans

Contribution scientifique

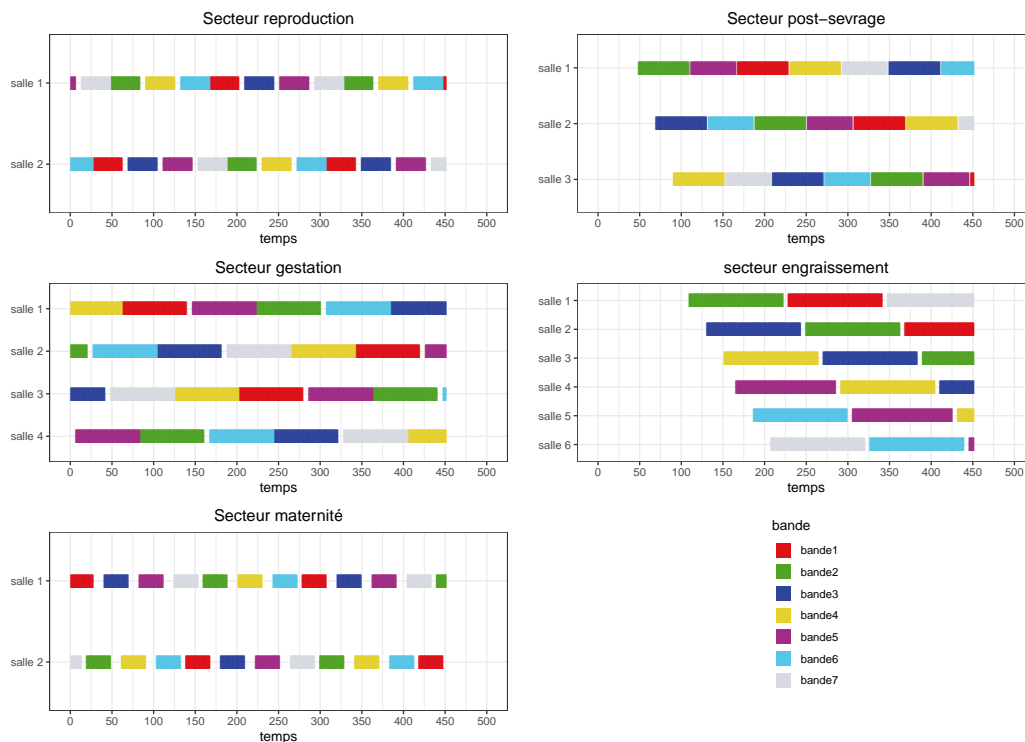


FIGURE 7 – Résultats des localisations issues des simulations. Les animaux sont localisés dans les différents secteurs et salles en fonction de leur état nominal, i.e. vu par leurs organisations (Chambre d’Agriculture de Bretagne, 2010).

une plate-forme de simulation MLABS existante (EMULSION), ce qui démontre la flexibilité et la cohérence de notre proposition avec les approches MLABS déjà structurées par des patterns. La preuve de concept mise en œuvre dans EMULSION contribue également à la modélisation épidémiologique multi-échelles des écosystèmes complexes. La représentation explicite des couplages entre agents, environnements et organisations est une étape importante pour représenter des populations et des environnements fortement structurés et étendre les paradigmes de modélisation classiques. Cela permettra de mieux comprendre la dynamique des environnements anthropisés et aider à évaluer des mesures de contrôles réalistes.

Dans un cadre applicatif, nous avons illustré comment le pattern organisationnel multi-niveaux permet de représenter avec précision l’organisation complexe d’un système agricole fortement structuré : l’élevage porcin en bande. Notre prochain objectif sera d’étudier la propagation d’un agent pathogène respiratoire dans le système en tenant compte, à un niveau de détail fin social et spatial, de la dynamique de l’exposition des animaux aux pathogènes portés par leur environnement et de celle de leurs contacts se-

lon leurs stades physiologiques. Cela permettra de mieux comprendre la circulation des agents pathogènes dans de tels systèmes complexes, notamment pour construire et évaluer des scénarios de maîtrise réalistes et efficaces.

Enfin, l’introduction de caractéristiques organisationnelles dans les MLABS contribue à réduire l’écart entre les approches structurelle et fonctionnelle des systèmes complexes. L’approche par design pattern permet d’élargir le concept d’organisation en tant que relation locale entre plusieurs niveaux réifiés par des agents, ce qui amène à une architecture flexible et homogène.

Remerciements

Ce travail est co-financé par la division Santé Animale d’INRAE (Institut national de recherche pour l’agriculture, l’alimentation et l’environnement) et de la région Pays de la Loire.

Références

- Bousquet, F. and Le Page, C. (2004). Multi-agent simulations and ecosystem management : a review. *Ecological Modelling*, 176(3-4) :313–332.
- Camus, B., Bourjot, C., and Chevrier, V. (2015). Considering a Multi-Level Model as a Society of Inter-

Contribution scientifique

- acting Models : Application to a Collective Motion Example. *Journal of Artificial Societies and Social Simulation*, 18(3) :7.
- Chambre d'Agriculture de Bretagne (2010). Les conduites en bandes en production porcine - cohérence de la chaîne de bâtiments, Organisation du travail, Truies en groupe.
- Dignum, V., Meyer, J.-J. C., Weigand, H. G., Dignum, F., and Meyer, J.-J. C. (2008). An Organization-oriented Model for Agent Societies.
- Drogoul, A., Amouroux, E., Caillou, P., Gaudou, B., Grignard, A., Marilleau, N., Taillandier, P., Vavasseur, M., Vo, D.-A., and Zucker, J.-D. (2013). GAMA : multi-level and complex environment for agent-based models and simulations. In *Proceeding of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2013)*, pages 1361–1362.
- Ezanno, P., Andraud, M., Beaunée, G., Hoch, T., Krebs, S., Rault, A., Touzeau, S., Vergu, E., and Widgren, S. (2020). How mechanistic modelling supports decision making for the control of enzootic infectious diseases. *Epidemics*, 32 :100398.
- Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agent systems. In *Proceedings International Conference on Multi Agent Systems (ICMAS'98)*, pages 128–135. IEEE Comput. Soc.
- Ferber, J., Gutknecht, O., and Michel, F. (2004). From Agents to Organizations : An Organizational View of Multi-agent Systems. In *Agent-Oriented Software Engineering IV*, volume 2935, pages 214–230. Springer.
- Ferber, J., Michel, F., and Baez, J. (2005). AGRE : Integrating Environments with Organizations. In *Environments for Multi-Agent Systems*, Lecture Notes in Computer Science, pages 48–56. Springer.
- Fischer, K. (1999). Robotics and Autonomous Systems. *Robotics and Autonomous Systems*, page 11.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison Wesley.
- Hannoun, M., Boissier, O., Sichman, J. S., and Sayettat, C. (2000). MOISE : An Organizational Model for Multi-agent Systems. In *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 156–165. Springer.
- Hjorth, A., Head, B., Brady, C., and Wilensky, U. (2020). LevelSpace : A NetLogo Extension for Multi-Level Agent-Based Modeling. *Journal of Artificial Societies and Social Simulation*, 23(1) :4.
- Hübner, J. F., Vercouter, L., and Boissier, O. (2009). Instrumenting Multi-agent Organisations with Artifacts to Support Reputation Processes. In *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, Lecture Notes in Computer Science, pages 96–110. Springer.
- Juziuk, J., Weyns, D., and Holvoet, T. (2014). *Design Patterns for Multi-agent Systems : A Systematic Literature Review*, pages 79–99. Springer.
- Keeling, M. J. and Rohani, P. (2008). *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press.
- Klügl, F. and Karlsson, L. (2009). Towards Pattern-Oriented Design of Agent-Based Simulation Models. In *Multiagent System Technologies*, volume 5774 of *Lecture Notes in Computer Science*, pages 41–53. Springer.
- Kubera, Y., Mathieu, P., and Picault, S. (2011). IODA : An interaction-oriented approach for Multi-Agent Based Simulations. *Journal of Autonomous Agents and Multi-Agent Systems*, 23(3) :303–343.
- Mathieu, P., Morvan, G., and Picault, S. (2018). Multi-level agent-based simulations : Four design patterns. *Simulation Modelling Practice and Theory*, 83 :51–64.
- Mathieu, P., Picault, S., and Secq, Y. (2015). Design patterns for environments in multi-agent simulations. In *Proceedings of the 18th Conference on Principles and Practice of Multi-Agent Systems (PRIMA 2015)*, volume 9387, pages 678–686. Springer.
- Minar, N., Burkhart, R., Langton, C. G., and Askenazi, M. (1996). The Swarm Simulation System : A Toolkit for Building Multi- Agent Simulations. Technical report.
- Morvan, G. (2012). Multi-level agent-based modeling - A literature survey. *arXiv :1205.0561 [cs]*. arXiv : 1205.0561.
- Morvan, G., Veremme, A., and Dupont, D. (2011). IRM4MLS : The Influence Reaction Model for Multi-Level Simulation. In *Multi-Agent-Based Simulation XI*, volume 6532 of *Lecture Notes in Computer Science*, pages 16–27. Springer.
- Picault, S., Huang, Y.-L., Sicard, V., Arnoux, S., Beaunée, G., and Ezanno, P. (2019). EMULSION : Transparent and flexible multiscale stochastic models in human, animal and plant epidemiology. *PLOS Computational Biology*, 15(9) :e1007342.
- Picault, S. and Mathieu, P. (2011). An Interaction-Oriented Model for Multi-Scale Simulation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'2011)*, pages 332–337. AAAI.
- Salines, M., Andraud, M., Rose, N., and Widgren, S. (2020). A between-herd data-driven stochastic model to explore the spatio-temporal spread of hepatitis E virus in the French pig production network. *PLOS ONE*, 15(7) :e0230257.
- Sicard, V., Andraud, M., and Picault, S. (2021). Organization as a Multi-level Design Pattern for Agent-based Simulation of Complex Systems :. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*, pages 232–241. SCITEPRESS.
- Weyns, D., Haesevoets, R., Helleboogh, A., Holvoet, T., and Wouter, J. (2010). The MACODO Middleware for Context-Driven Dynamic Agent Organizations. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 5(4) :16.
- Zhang, X. and Norrie, D. H. (1999). Holonic Control at the Production and Controller Levels. In *In Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, pages 215–224.

Usage des parking relais dans les comportements de déplacements intermodaux : Génération de demande de population d'agents à partir de données réelles

A. O. Diallo^a
azise.oumar.diallo@imt-lille-douai.fr

G. Lozenguez^a
guillaume.lozenguez@imt-lille-douai.fr

A. Doniec^a
arnaud.doniec@imt-lille-douai.fr

R. Mandiau^b
rene.mandiau@uphf.fr

^aCERI Systèmes Numériques, Institut Mines Télécom Lille Douai

^bLAMIH UMR CNRS 8201, Université Polytechnique Hauts-de-France, Valenciennes, France

Résumé

La disponibilité de scénarios de simulation des systèmes de transport à partir de données ouvertes facilite l'étude de nouvelles politiques de mobilité. Ces données sont souvent anonymisées pour être en accord avec le règlement général sur la protection des données (RGPD). Dans cet article, nous présentons une génération de simulation de mobilité à partir de données anonymisées basée sur un modèle de choix modal. Ce modèle intègre un comportement de déplacement intermodal combinant la voiture personnelle (VP) et les transports publics (TP) nécessitant l'usage des parkings relais. Nous montrons comment reconstruire les communes et les coordonnées des origines/destinations des déplacements nécessaires à l'estimation du modèle de choix discret. Ce modèle a été appliqué à la Métropole Européenne de Lille à partir de la dernière enquête ménage-déplacement. Les résultats obtenus montrent que la simulation est capable de reproduire les comportements de déplacements effectués en combinant la VP et les TP.

Mots-clés : Intermodalité, parking relais, modèle agent, modèle de choix discret, données ouvertes, MATSim, MEL

Abstract

The availability of transportation scenarios from open data introduces new research motivations. These data are often anonymized to protect the privacy of the survey participants. In this paper, we present a generation of mobility simulation from anonymized data based on a modal choice model. This model integrates an intermodal alternative combining personal car and public transport (pt) for commutes through park and ride (PR) facilities. We show how to reconstruct the municipalities and the coordinates of

the origins/destinations of the trips necessary for the estimation of the discrete choice model. This model was applied to the European Metropolis of Lille from the last household-travel survey. The preliminary results obtained show that the simulation is capable of reproducing the behaviors of trips performed by combining car and PT.

Keywords: Agent-based model; Open-data; Discrete mode choice; Inter-modality; MATSim; MEL

1 Introduction

Les systèmes de transport représentent un moyen important pour accompagner le développement des activités socio-économiques. Ils sont également source de plusieurs conséquences négatives sur la santé et le bien-être des usagers à cause de la pollution, la congestion, les nuisances sonores ou encore les accidents routiers. Pour réduire ces impacts négatifs, les pouvoirs publics s'appuient sur plusieurs leviers qui visent à réduire l'usage de la voiture personnelle en faveur des transports publics (TP) et autres modes plus doux (p. ex., la marche, le vélo). Ces politiques de mobilité qui se basent souvent sur des ajustements de la tarification des TP et l'aménagement d'infrastructures comme les parkings relais, ne sont efficaces que si des mesures d'accompagnement existent.

Dans le cas de la mise en place de parkings relais par exemple, l'emplacement, la capacité, la disponibilité d'infrastructures de recharge des véhicules électriques, le coût de stationnement et de recharge, sont autant de facteurs déterminants pour obtenir une politique d'intermodalité efficace [16].

La disponibilité de scénarios de transport à partir de données ouvertes crée de nouvelles perspectives d'études et de recherche. Ces scénarios peuvent être utilisés par la suite pour valider et évaluer différents algorithmes (p. ex., routage, émissions de CO₂ et évacuations) et de nouvelles stratégies de mobilité (p. ex., partage de vélos, intermodalité, péage intra-urbain). De plus en plus de villes, par le biais de politiques de données ouvertes, mettent à disposition plusieurs bases de données sur les transports telles que les enquêtes sur les déplacements (p. ex., EMD), les horaires des transports publics, les données de comptage du trafic. De plus, avec le développement des logiciels de simulation de transport à base d'agent [1] tels que *MATSim* [10], il est possible d'expérimenter différents scénarios de mobilité tout en prenant en compte le comportement des usagers dans leurs déplacements [14, 15, 3].

L'étude comparative récente de simulateurs réalisée dans [7] pointe le simulateur *MATSim* pour simuler des politiques d'intermodalité en étant le plus adapté possible pour reproduire les comportements de mobilité intermodale à grande échelle. Plusieurs scénarios reproductibles ont été proposés en utilisant *MATSim* pour différentes villes comme Santiago, la capitale du Chili [12], Berlin en Allemagne [20] et Paris en France [11, 8]. C'est sur la base de ces récents travaux [8] que Diallo et al. [6] ont généré le premier scénario de mobilité prenant en compte la combinaison de la voiture personnelle et les transports publics à partir des données ouvertes de la Métropole européenne de Lille (MEL).

Cet article est une extension des travaux présentés dans [6]. Il décrit de manière détaillée l'approche méthodologique (section 3.1) et présente les derniers résultats obtenus sur la fréquentation des PR. La motivation de ce travail est d'étudier sous l'angle des systèmes multi-agents les comportements de mobilité intermodale, c'est-à-dire la possibilité de combiner plusieurs modes de transport au cours d'un même déplacement. L'intermodalité est présentée comme l'une des solutions aux problèmes de mobilité. Il est au cœur des politiques de mobilité dans les grandes villes afin de réduire l'utilisation de la voiture personnelle (VP) [5]. Nous sommes particulièrement intéressés par la combinaison de la VP avec les TP à travers l'installation de parkings relais. A notre connaissance, très peu de travaux de simulation se sont focalisés sur la définition et l'évaluation de politiques d'intermodalité à partir des habitudes de déplacement des individus.

Plusieurs facteurs (p. ex., la catégorie socio-

professionnelle (CSP), la durée du déplacement, le coût du transport, le motif du déplacement et la météo) ont été identifiés comme ayant un impact sur le choix modal. Ces facteurs doivent, par conséquent être pris en compte pour la mise en place de systèmes de transport intermodaux [17]. Cependant, afin d'estimer les modèles de choix de modes discrets, il est nécessaire d'utiliser des bases de données contenant ces informations. Bien que différentes sources de données ouvertes sur la mobilité existent (p. ex., EMD), elles sont pour la plupart anonymisées et n'intègrent pas directement les aspects d'intermodalité. Afin de préserver la vie privée des participants à l'enquête, les informations spatiales sur les lieux de résidence et d'activité (p. ex., le travail, l'éducation et les loisirs) sont souvent floutées et les coordonnées des origines/destinations (O/D) sont supprimées des versions accessibles au public.

Dans cet article, nous présentons une démarche de simulations d'intermodalité à partir de données anonymisées. Nous décrivons d'abord comment reconstituer les informations manquantes à savoir les zones d'activités et les O/D des déplacements. Ensuite, nous montrons comment estimer un modèle de choix modal intégrant la combinaison de la voiture personnelle et des transports publics et comment l'intégrer dans un outil de simulation. Enfin, nous présentons un cas d'application du modèle sur le périmètre de la Métropole Européenne de Lille (MEL). Ce modèle peut être utilisé comme cadre d'étude pour évaluer les politiques de mobilité actuelles mais aussi pour en tester de nouvelles comme l'augmentation des parkings relais ou encore la mise en place de péages intra-urbains.

Le suite de l'article est présentée comme suit. Dans la section 2, nous introduisons les données ouvertes traditionnellement disponibles et les ajustements nécessaires à effectuer pour récupérer les informations manquantes. La section 3 décrit la méthodologie pour générer une demande de transport intermodale. Ensuite, dans la section 4, nous présentons les expérimentations basées sur les données de la MEL en détaillant la procédure de la simulation, les premiers résultats et une discussion. Enfin, la section 5 conclut et donne différentes perspectives.

2 Les données dans un contexte d'intermodalité

Dans cette section, nous présentons les bases de données disponibles sur le comportement des individus dans une situation de choix du mode

de transport et comment reconstituer les zones d'activités (communes et coordonnées des Origines/Destinations (O/D)).

2.1 Les données disponibles

Pour estimer un modèle de choix modal et construire un scénario (demande et offre de transport) de simulation, il est nécessaire de disposer d'une ou plusieurs bases de données pour mettre en évidence les caractéristiques socio-démographiques des individus et leurs comportements de mobilité. Il existe très peu de données spécifiques sur les comportements de mobilité intermodale des populations. La seule enquête propre aux comportements intermodaux à notre connaissance a été réalisée pour la ville de Berlin en Allemagne [17]. En France, les principales bases de données sur la mobilité accessibles au public sont le recensement et les Enquêtes Ménage-Déplacement (EMD).

Le recensement de la population contient toutes les caractéristiques socio-démographiques des individus (p. ex., l'âge et la CSP) et le principal mode de transport pour leurs déplacements domicile-travail et domicile-étude. Bien que l'échantillon de cette enquête soit représentatif de la population (environ 35%), il contient néanmoins très peu d'informations sur les habitudes de déplacement (p. ex., la fréquence et les motifs secondaires comme le loisir ou les achats ne sont pas informés). La représentation statistique spatiale disponible est à l'échelle du *Canton-ou-Ville* et de l'*IRIS* (au moins 200 habitants) ¹.

L'EMD fournit des informations détaillées sur les habitudes de déplacement de la population en renseignant, par exemple, sur toutes les chaînes d'activités quotidiennes (p. ex., travail, éducation, loisir, achats), le mode de transport effectivement utilisé, et la possession de matériel de transport (p. ex., le permis de conduire et le titre de transport public). Il contient également certaines caractéristiques socio-démographiques telles que l'âge et les CSP. Les principaux inconvénients de cette base de données sont son faible taux d'échantillonnage (entre 0,8 à 2%) et l'absence de coordonnées O/D qui ne sont connues qu'à l'échelle des secteurs de tirage ².

1. *Canton-ou-ville* est un groupement de une ou plusieurs municipalités entières. Les communes d'au moins 10 000 habitants et la plupart des communes de 5 000 à 10 000 habitants sont divisées en *IRIS*.

2. L'échantillon de ménages est tiré au hasard sans remplacement et est géographiquement stratifié. Les secteurs de tirage

En général, ce sont les données de l'enquête de mobilité qui sont les plus souvent utilisées pour estimer des modèles de choix modaux et générer des demandes de transport pour la simulation. Il est également possible de croiser/fusionner plusieurs sources de données comme le recensement avec l'enquête de mobilité pour obtenir une source de données plus complète et représentative dans le cas de la génération de la demande de transport (voir [8]).

Ainsi, nous avons utilisé les données de l'EMD pour l'estimation du modèle de choix discret de mode de transport en commençant par reconstruire les communes et les coordonnées des O/D à partir des secteurs de tirage (sans trahir la protection des données personnelles). Pour la génération de la demande de transport, nous avons utilisé le recensement et l'EMD. Ces différents éléments sont développés dans la section 3.

2.2 Reconstruction des communes

En général, les zones géographiques de résidence et d'activités sont représentées par le code *IRIS* (6 chiffres dont les cinq premiers correspondent au code commune). Pour flouter ces informations, les zones ont été codées en 4 chiffres dont les trois premiers correspondent aux secteurs de tirage disponibles avec les données d'enquête. Le dernier chiffre renseigne sur la nature de la zone : 0 pour zone fine simple ³, 1 – 8 pour désigner un générateur de trafic (p. ex. 1 pour commerce, 3 pour éducation). Par conséquent, pour trouver les communes, nous avons superposé les formes géographiques des secteurs de tirage avec les informations de délimitation des communes de la zone d'étude (généralement disponible en données ouvertes). Ces communes ainsi reconstituées sont utilisées pour la génération de la population synthétique pour créer la demande de transport.

La Figure 1 donne un aperçu de cette superposition du cas d'étude de la MEL : les contours des communes sont en bleu et ceux des secteurs de tirage en vert. On voit que le centre correspondant à la ville de Lille (commune principale) est composé de plusieurs secteurs de tirage tandis que certaines communes (périphériques) appartiennent à une même zone statistique.

son des zones du périmètre d'étude d'où sont tirés les participants à l'enquête (au moins 70 ménages et 160 personnes par zone, seuil de représentativité statistique : https://www.cnis.fr/wp-content/uploads/2017/12/DPR_2013_2e_reunion_COM_territoires_enquetes_deplacements_certu.pdf).

3. Les zones fines décrivent précisément les O/D des déplacements dans une zone géographique plus petite que le secteur de tirage.

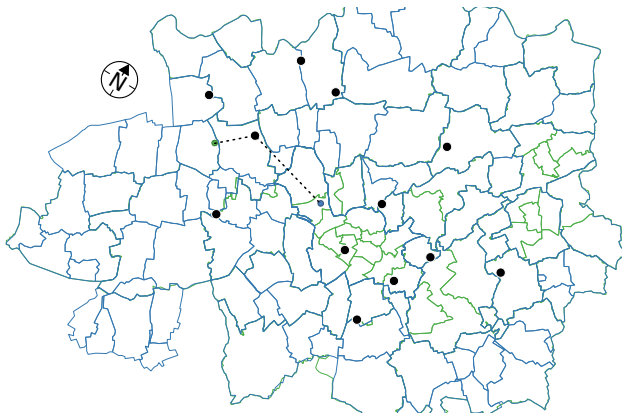


FIGURE 1 – Contours des communes (en bleu) et des secteurs de tirage (en vert) de la MEL.

Grâce à cette méthode, il est possible d'identifier facilement les communes de résidence et d'activités à partir des secteurs de tirage. En effet, comme on peut le voir sur la Figure 1, la taille du secteur de tirage est inférieure ou égale à celle de la commune. Quand la commune a une forte densité démographique (comme le cas de Lille), elle est composée de plusieurs secteurs de tirage.

2.3 Reconstruction des O/D des déplacements

La principale limite de l'enquête de mobilité (EMD) est l'absence totale des coordonnées origine/destination (O/D) des déplacements. Seules la distance euclidienne (distance à vol d'oiseau) et les O/D à l'échelle des secteurs de tirage pour chaque déplacement sont disponibles. Ces coordonnées sont nécessaires à l'estimation des paramètres du modèle de choix discret de mode de transport. En effet, il faut reproduire le déplacement initial (comme défini dans l'enquête) avec toutes les autres alternatives de transport. Pour cela, nous avons besoin des coordonnées géographiques des O/D du déplacement pour prendre en compte toute l'offre de transport disponible (p. ex., la proximité d'un arrêt de transport public). Ainsi, l'utilisateur pourra faire son choix en fonction de l'utilité que lui apporte chaque mode. La méthode que nous utilisons est définie en deux étapes : 1) nous générons aléatoirement $N \in \mathbb{N}$ points dans les secteurs de tirage des O/D, 2) nous recherchons un point dans la zone d'origine correspondant à un point de la zone de destination, de sorte que la distance (euclidienne) entre ces deux points soit la plus proche possible à celle fournie dans l'enquête. Il faut noter ici que nous conservons la coordonnée du

point correspondant au domicile pour garantir la consistance des déplacements entre le domicile et les autres activités (p. ex., travail, éducation). La qualité des points obtenus est discutée en section 4.4.

3 Mise en place de la simulation

Cette section présente notre approche méthodologique ainsi que ses différentes composantes.

3.1 Approche méthodologique

Nous présentons ici l'approche méthodologique que nous avons suivie pour réaliser nos scénarios de simulations. La Figure 2 donne un aperçu de notre approche.

La demande de mobilité concerne tous les besoins de déplacements de la population d'étude. Les données nécessaires pour l'estimation de la demande de transport proviennent généralement des enquêtes comme décrit dans la sous-section 2.1. L'offre de transport est composée du réseau multimodal et des services de mobilité (p. ex., TP, parkings relais, co-voiturage, vélo libre partage). Le réseau de transport peut être modélisé à partir des données géographiques (p. ex., *OpenStreetMaps*⁴). Les données sur les services de mobilité sont disponibles auprès des structures organisatrices.

La réalisation d'une simulation de mobilité à base d'agents passe par l'estimation de plusieurs modèles. Le premier est relatif à la génération d'une population synthétique d'agents qui contient les différents profils d'utilisateurs de la population d'étude. Notre cas d'étude étant lié au choix modal, il est nécessaire de définir un modèle spécifique de choix discret de mode de transport pour bien retranscrire le comportement de mobilité des usagers. La demande de transport est générée à partir des matrices O/D et la population synthétique.

Une fois les différents modèles mis en place, il faut effectuer des simulations pour mesurer les écarts par rapport à la situation de référence. Cette étape peut nécessiter le développement ou l'extension d'un logiciel de simulation pour réaliser ces expérimentations. Une phase de calibration sera nécessaire pour avoir un modèle fidèle aux données réelles.

4. OpenStreetMaps est un projet collaboratif de cartographie en ligne qui vise à constituer une base de données géographiques libre du monde, en utilisant le système GPS et d'autres données libres.

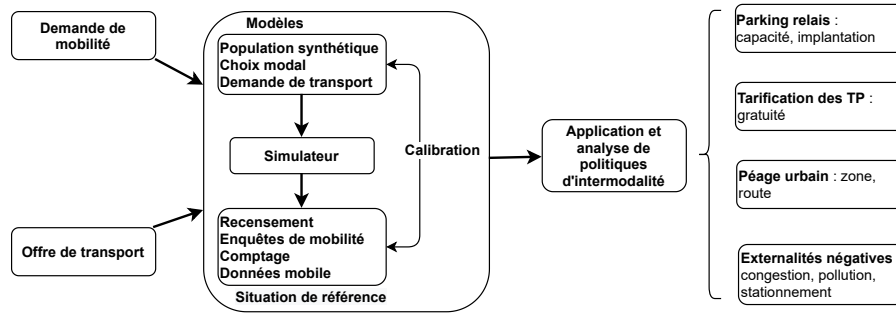
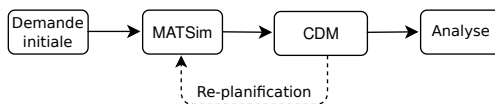


FIGURE 2 – Approche méthodologique pour la simulation des politiques d'intermodalité

3.2 Description de *eqasim*

Dans notre approche, nous nous intéressons aux comportements de mobilité des individus en particulier à leur choix modal. Nous utilisons l'outil de simulation *eqasim* [9] qui intègre un module de choix discret de mode de transport dans le micro-simulateur *MATSim* [10]. La Figure 3 décrit le cycle général de simulation de *eqasim*.

FIGURE 3 – Cycle de simulation de *eqasim*.

La demande initiale est composée du réseau routier et des chaînes d'activités quotidiennes (p. ex., domicile - travail - achats - domicile) ainsi que des déplacements à effectuer par chaque individu (agent). Ces éléments, contenus dans le plan de l'agent, seront simulés par l'outil de micro-simulation pour un nombre donné d'itérations. À la fin de chaque itération, de nouveaux temps et distances de déplacement sont fournis au module de Choix Discret de Mode de transport (*CDM*). Ces nouvelles valeurs sont issues des interactions entre les agents et peuvent varier d'une itération à une autre (p. ex., présence ou non de congestions). Au niveau du module de choix discret, une portion (entre 5 à 10%) d'agents sont autorisés à modifier leur plan initial en changeant le mode de transport en fonction de ces informations dans une étape de re-planification. Par conséquent, ces agents chercheront à optimiser leur temps de parcours à chaque itération jusqu'à atteindre un équilibre utilisateur où aucune amélioration n'est possible. Cette proportion d'agents est déterminée de façon à obtenir des temps de trajet stables sur le réseau. En effet, une forte valeur implique que

beaucoup d'agents ont effectué des choix (p. ex., utilisation ou non de la voiture). Ainsi, ces choix engendreront des changements instables sur le réseau (parfois des congestions, parfois des espaces libres). Un faible pourcentage engendrera des temps de simulation plus longs pour obtenir une situation stable. Il convient de noter ici que l'étape de notation (*scoring*) dans la version standard de *MATSim* n'est plus utilisée. En effet, dans ce cas, le choix du plan et par conséquent du mode de transport est *a posteriori* que *a priori*. De plus, la fonction étant basée uniquement sur la réalisation de l'activité, la durée et le coût du déplacement, ne traduit pas assez les comportements hétérogènes des individus dans leurs choix.

3.3 Génération de la demande initiale et du réseau multimodal

La demande de transport initiale est généralement créée à partir d'un générateur de population synthétique d'agents. Cette population doit refléter les données de mobilité, par exemple issues des enquêtes de mobilité et/ou des données de recensement (voir sous-section 2.1). Pour ce faire, nous avons utilisé le synthétiseur⁵ initialement développé pour l'Île-de-France [8] pour générer la demande de transport. Ce synthétiseur utilise une *mise en correspondance (matching) statistique* entre les données issues du recensement et de l'EMD pour obtenir une population générique plus complète. Le processus de mise en correspondance est basé sur certaines caractéristiques socio-démographiques (p. ex., l'âge, le genre, le revenu et la catégorie socio-professionnelle) et la représentation géographique (commune dans ce cas) partagée par ces deux bases.

5. Les outils de synthèse (ou synthétiseur) font référence ici à un ensemble de méthodes et de scripts qui permet de générer une population synthétique d'agents ou une demande de transport à partir de données réelles.

Le réseau de transport multimodal est produit à partir des données *OSM* et *GTFS*⁶ de la zone d'étude en utilisant l'outil *pt2matsim* [18].

3.4 Estimation du modèle de choix discret de mode

Pour une introduction complète aux modèles de choix discrets avec simulations, le lecteur pourra se référer à l'ouvrage de *Train* [19]. Nous présentons ici les bases pour comprendre notre modèle de choix discret.

Les modèles de choix discrets sont généralement basés sur la notion d'utilité aléatoire qu'un individu i a pour une alternative (mode de transport dans ce cas) m [2]. L'utilité est une fonction de **variables explicatives** composées des caractéristiques de l'individu (p. ex., CSP, âge) et des attributs de l'alternative (p. ex., temps de parcours, coût de transport) à laquelle on ajoute un **terme d'erreur**. Le terme d'erreur permet entre autres de tenir compte de la nature aléatoire du comportement humain, du manque d'informations plus précises dans la prise de décision (individuelle) et les attributs non observables des alternatives. En général, les termes d'erreurs sont rapprochés soit par une loi normale pour les modèles *Probit* (corrélation possible entre les alternatives), soit par une loi de Gumbell pour les modèles *Logit* (pas de corrélations supposées entre les alternatives). Dans la littérature, les modèles les plus utilisés sont les modèles *Logit Multinomial* (MNL) qui sont des modèles *Logit* à plus de deux alternatives. Ce choix se justifie principalement par la facilité de calcul des probabilités de choix offerte par ces modèles.

Par la suite, nous définissons notre modèle de choix discret de type *MNL* en ajoutant une alternative intermodale consistant à combiner la voiture personnelle avec les transports publics (*VP+TP*). Les fonctions d'utilités U pour toutes les alternatives sont adaptées de [9] et ont été développées dans [6]. Nous présentons ici uniquement l'utilité de l'alternative intermodale *VP+TP* à titre d'exemple pour des raisons de concision. Cette fonction d'utilité est basée sur celles de la voiture et des TP. Elle n'est pas utilisée dans l'estimation du modèle, mais uniquement dans l'étape de simulation du comportement de mobilité.

6. La spécification *GTFS* (*General Transit Feed Specification*) est une norme pour les horaires de transports publics et les informations géographiques : <https://gtfs.org/>

$$\begin{aligned}
 U_{i,VP+TP} = & \beta_{ASC,VP+TP} + U_{i,VP} + U_{i,PT} \\
 & + \theta_{tempsEchange} \times x_{tempsDansLeVehicule,VP} \\
 & - \theta_{penaliteRechercheParking} \times \theta_{tempsDansLeVehicule,VP} \\
 & - \beta_{ASC,VP} - \beta_{ASC,TP}
 \end{aligned}
 \tag{1}$$

Où, i : individu, β : paramètres (coefficients) du modèle à estimer, x : variables explicatives (p. ex., le temps de parcours, le coût) du choix du mode, *ASC* (*Alternative Specific Constantes*) : constantes spécifiques à l'alternative décrivant la variation de choix non expliquée par les attributs uniquement, θ : paramètres de calibration qui seront à ajuster (estimer) manuellement. Nous prenons en compte le temps d'échange (*tempsEchange*) entre les deux modes. Nous retirons la pénalité relative à la recherche de parking (*penaliteRechercheParking*) appliquée pour l'usage de la voiture uniquement.

Les coûts d'utilisation de la voiture sont évalués par km parcouru et incluent les coûts d'utilisation, les coûts de carburant et les frais de péage autoroutier (le cas échéant) [4]. Le coût du déplacement en TP dépend de la possession d'un titre de transport valide.

Les paramètres estimés du modèle à partir d'un tiers des données de l'enquête ménage-déplacement de la MEL sont présentés dans [6].

3.5 Routage des déplacements effectués par *VP+TP*

La version actuelle de *MATSim* (*eqasim*) ne prend pas directement en compte le routage des déplacements effectués en *VP+TP* via les parkings relais (PR). Comme mentionné ci-dessus, la combinaison de la voiture personnelle et des TP dépend fortement des installations de PR. Ainsi, l'algorithme de routage des déplacements en *VP+TP* que nous avons développé, se présente comme suit : 1) trouver le PR le plus proche du domicile de l'agent, 2) effectuer le routage en voiture entre le domicile et le PR en utilisant le module de routage de *MATSim*, 3) effectuer le routage entre le PR et la destination en faisant appel au module de routage des TP (*SwissRailRaptor*). Pour le retour, on procède de la même manière mais cette fois en commençant par le trajet en PT. Cette approche garantit de passer toujours par le même PR pour faire le déplacement en voiture et en PT. Il faut souligner ici, que nous nous intéressons pour l'instant qu'au premier déplacement dont l'origine est le domicile

des agents. En outre, l'utilisation du PR le plus proche du domicile est un choix de modélisation qui se justifie par le manque de données relatives à l'usage des PR pour identifier des comportements spécifiques.

Un exemple de déplacement à partir des PR, illustré par la ligne noire en pointillée, est présenté dans la Figure 1. Les installations des 12 PR sont représentées par les points noirs. Ce déplacement commence par le point vert (domicile) au point bleu (un magasin est dans ce cas) en passant par un PR. Dans cet exemple, on peut voir que l'agent dispose de trois PR à côté de son domicile. Il choisit celui le plus proche de son domicile.

4 Un exemple d'application à la MEL

Cette section présente les résultats pour la génération de scénarios de mobilité intermodale en incluant la possibilité de trajets mixte (*VP+TP*), sur le territoire de la Métropole Européenne de Lille (MEL). Elle donne les informations générales sur la zone d'étude et les premiers résultats de validation et de calibration de la simulation.

4.1 Présentation de la MEL

La MEL est une coopération intercommunale de type métropolitaine française située au centre du Département du Nord dans les Hauts-de-France. En 2020, elle était composée de 95 communes avec 1,1 million d'habitants sur une superficie de 672 km².

La MEL a engagé plusieurs investissements pour développer les transferts modaux vers les modes actifs et encourager les déplacements intermodaux combinant la voiture et les transports publics (*VP+TP*). Aujourd'hui, le réseau de transports publics comprend deux lignes de métro (43,6 km), deux lignes de tramway (22 km) et environ 90 lignes d'autobus. La MEL compte 12 parkings relais correspondant à 5 005 places de stationnement pour la correspondance avec le bus, le métro et le train et pour le covoiturage. Ces différentes offres de transport et les projets d'investissement qui en résultent font de la MEL une étude de cas intéressante pour l'analyse des politiques de transport alternatives à l'échelle métropolitaine.

L'enquête ménage-déplacement (EMD) de la MEL, réalisée en 2016, a été utilisée dans la présente étude. Les participants à l'enquête sont

âgés de 5 ans et plus et étaient composés de 9 479 personnes appartenant à 4 539 ménages soit environ 0,87% de la population. Une version des données de cette enquête est disponible en libre accès sur le site open data de la MEL ⁷.

4.2 Configuration de la simulation

Nous donnons quelques détails de la configuration de la simulation tels que la taille de la population simulée, le nombre d'itérations, la part d'agents autorisés à changer de mode de transport et certains paramètres du réseau routier.

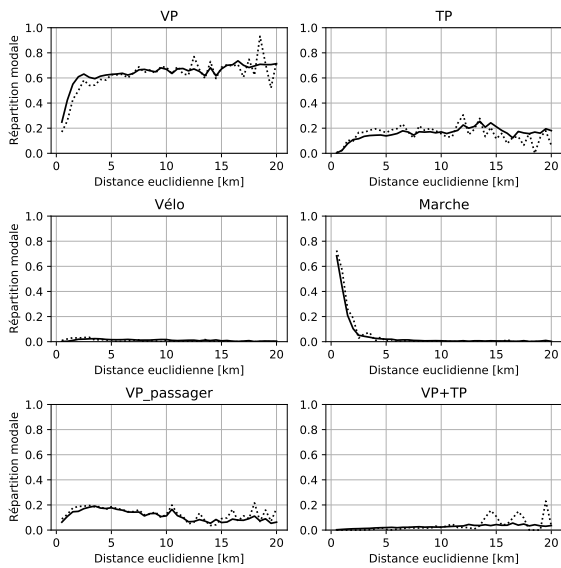
La population synthétique *MATSim* générée dans cette étude représente un échantillon de 10% de la population réelle ce qui correspond à 87 057 agents avec un total de 370 446 déplacements contre 24 629 déplacements dans l'EMD. Pour prendre en compte cet échantillonnage sur les flux de trafic sur le réseau routier réel, le débit et les capacités de stockage de tous les tronçons du réseau automobile généré ont été multipliés par 0,08 (ce qui correspond à une configuration assez classique de *MATSim*). Le nombre d'itérations est fixé à 100 et la portion d'agents autorisée à la re-planification à chaque itération a été fixée à 5%.

4.3 Résultats de validation et calibration de la simulation

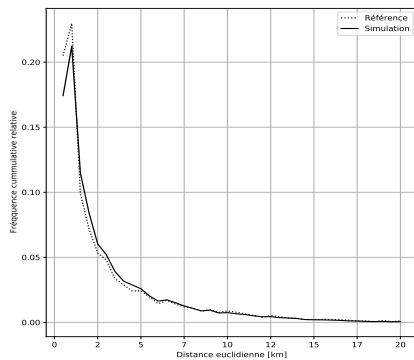
La Figure 4a présente une comparaison entre les parts modales issues de l'enquête ménage-déplacement de la MEL (référence en pointillé) et celles provenant de la simulation (en trait plein). Les parts modales pour chaque mode sont présentées en ordonnée tandis que les distances euclidiennes des déplacements sont en abscisse. Cette représentation permet une meilleure appréciation des distributions modales. Par exemple, nous pouvons voir que la part de la marche est très importante pour les petites distances et tend vers zéro pour les grandes distances.

Les résultats de simulation correspondent très bien à la situation de référence (EMD 2016) pour tous les modes de transport avec seulement un tiers des données de l'EMD utilisées pour estimer le modèle de choix discret de mode de transport. Il faut noter ici que la répartition modale ne dépend pas seulement du module de choix

7. <https://opendata.lillemetropole.fr/explore/dataset/enquete-deplacement-2016/information/?location=10,50.65641,3.03338&basemap=jawg.streets>



(a) Parts modales en fonction de la distance euclidienne issues de la simulation (trait plein) et de l’EMD 2016 de la MEL (trait en pointillé).



(b) Distribution des distances parcourues en simulation (ligne pleine) et dans l’EMD (ligne pointillée).

FIGURE 4 – Résultats des simulations

discret mais aussi des temps de parcours en VP et en TP qui changent d’une itération à une autre et dépendent récursivement du choix du mode. Ces résultats ne sont pas à confondre avec l’évaluation du modèle de choix discret qui n’est pas discutée dans ce présent article. Dans ce cas effectivement, nous aurons utilisé les deux tiers de la base de l’EMD pour la validation et les tests.

La combinaison *VP+TP* est bien représentée après une calibration manuelle de β_{ASC,car_pt} à 1,25 pour accentuer ce type de déplacement. Cette phase de calibration était nécessaire car le mode combiné n’est pas utilisé assez fréquemment dans l’enquête (0,5% pour obtenir des résultats robustes lors de l’estimation du modèle de choix discret. Cependant, il y a quelques écarts

pour les longs déplacements (plus de 12 km) effectués en VP, TP et *VP+TP*. Ceci est principalement dû au manque de données de référence sur ces types de déplacements, lesquels sont pour la plupart occasionnels (dans le périmètre restreint à la MEL). La Figure 4b donne un aperçu de la distribution des distances de trajet à partir de la simulation et de l’enquête déplacement. Nous pouvons ainsi mettre en évidence que les déplacements générés pour la population synthétique correspondent correctement aux données de l’EMD. De plus, la partie la plus importante des déplacements comporte moins de 12 km de distance.

La Figure 5 donne un aperçu de la fréquence d’utilisation (nombre d’entrées) journalière des PR dans la ville de Lille. La période d’utilisation en abscisse et le nombre d’utilisation des PR est en ordonnée. La capacité de chaque PR est fournie en parenthèses pour donner une idée du taux de remplissage. Les résultats présentés ici sont basés sur des données de sortie redressées à 100% pour correspondre à la population réelle de la MEL.

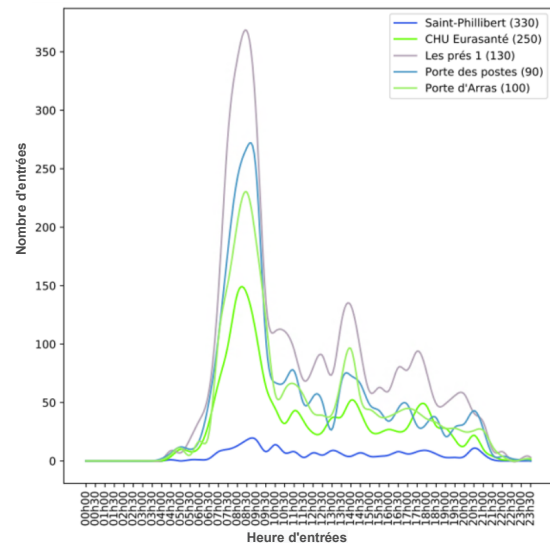


FIGURE 5 – Fréquence d’utilisation (nombre d’entrées) journalière des PR situés à Lille.

On observe une forte affluence dans les PR pendant les heures de pointe (p. ex., entre 6h et 8h), ce qui semble intuitivement cohérent. Les parkings relais situés dans la zone lilloise sont les plus fréquentés. Cette affluence s’explique principalement par le flux des déplacements à destination de Lille, lequel est le plus important dans la MEL. Par manque de données réelles sur la fréquentation des PR, nous ne sommes pas en

mesure, en l'état, de faire une comparaison entre les résultats de notre simulation et une situation de référence.

4.4 Limitations

Les principaux points de discussion sur notre méthode de création de scénario de mobilité à partir de données ouvertes anonymisées sont : la reconstruction des O/D et le modèle de choix discret de mode qui en résulte. La précision de l'algorithme de reconstruction des coordonnées O/D repose à la fois sur la taille de la zone et sur le nombre de points à générer aléatoirement. Plus la taille de la zone est petite, meilleure est la qualité des points reconstruits. De plus, plus le nombre de points aléatoires est élevé, plus nous avons de chances de trouver des points O/D dont la distance est plus proche de la distance de référence. Cependant, l'analyse considérée produit un temps de calcul très élevé. Par exemple, pour reconstruire les coordonnées O/D des 24 630 déplacements de l'enquête ménage-déplacement avec 500 points aléatoires, l'algorithme prend 15 heures de calcul sur une machine *i7* avec 16 Go de RAM.

Une solution à ce problème serait d'utiliser des bases de données géographiques comme la BD TOPO⁸, la base permanente des équipements (BPE)⁹ et les données *OSM* pour générer un ensemble fixe d'emplacements O/D correspondant aux zones d'activités éventuelles (p. ex., domicile, lieu de travail, école, commerce). Cette approche permettra d'éviter de générer un trop grand nombre de points aléatoires irréalistes (p. ex. couples de points trop proches pour des distances euclidiennes plus importantes). Par conséquent le temps de mise en correspondance sera réduit.

Concernant le modèle de choix discret, la fonction d'utilité de l'alternative *VP+TP* est essentiellement basée sur celles de la voiture de particulier (VP) et des transports publics (TP). Cependant, les déplacements combinant VP et TP sont fortement dépendants des installations de parkings relais. Par conséquent, il peut être intéressant de prendre en compte les attributs relatifs au parkings relais tels que le prix (abonnement), la localisation, la capacité, l'accessibilité et le temps de transfert. Malheureusement, à notre connaissance, il n'existe pas de base de données contenant ces informations. De plus, la construc-

tion d'un modèle *MNL* imbriqué comme proposé dans [13] permettrait de mieux prendre en compte les (éventuelles) dépendances entre VP et TP dans la combinaison des deux modes. Si l'estimation d'un tel modèle est simple, son intégration dans *MATSim* peut être ardue.

5 Conclusion

Dans cet article, nous avons présenté une création d'un scénario reproductible fondé sur les approches agent de la Métropole Européenne de Lille (MEL) à partir de données ouvertes. Ces données étant anonymisées, nous avons proposé une méthode pour reconstituer les informations floutées, à savoir les secteurs de résidence et d'activités et les coordonnées O/D des déplacements. À partir de là, nous avons développé un nouveau modèle de choix discret de mode qui prend en compte la combinaison voiture personnelle (VP) et les transports publics (TP) et qui a été intégré dans la plateforme de micro-simulation *eqasim*. Les résultats de la simulation relatifs aux partages modaux correspondent bien aux données de référence de l'enquête ménage-déplacement de la MEL. La simulation semble également rendre compte d'une utilisation réaliste des parkings relais dans les heures d'affluence dans les grandes villes de la MEL, comme Lille.

Dans le cadre de travaux futurs, nous prévoyons d'inférer l'impact qu'auraient des péages urbains et une optimisation des emplacements de parking relais sur les partages modaux. L'objectif est d'identifier les leviers permettant de rendre la combinaison *VP+TP* plus attractive par rapport au VP seul et ainsi voir des comportements de mobilité qui en découleraient.

6 Remerciements

Ce travail a été financé par la région des Hauts-de-France et le ministère en charge de l'industrie.

Références

- [1] Ana LC Bazzan and Franziska Klügl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(3) :375–403, 2014.
- [2] Moshe E Ben-Akiva, Steven R Lerman, and Steven R Lerman. *Discrete choice analysis : theory and application to travel demand*, volume 9. MIT press, 1985.

8. https://geoservices.ign.fr/blog/2020/10/26/Grand_angle_diffusion_BDTopo.html

9. <https://www.insee.fr/fr/statistiques/3568638>

- [3] Alexandre Bonhomme, Philippe Mathieu, and Sébastien Picault. Simuler le trafic routier à partir de données réelles. *Revue d'Intelligence Artificielle*, 30(3) :329–352, 2016.
- [4] CEREMA. Modélisation multimodale des déplacements de voyageurs concevoir un modèle de choix modal. Technical report, Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement, 2015.
- [5] S.G. Dacko and C. Spalteholz. Upgrading the city : Enabling intermodal travel behaviour. *Technological Forecasting and Social Change*, 89 :222–235, 2014.
- [6] Azise Oumar Diallo, Arnaud Doniec, Guillaume Lozenguez, and René Mandiau. Agent-based simulation from anonymized data : An application to lille metropolis. In *Proceedings of the 12th International Conference on Ambient Systems, Networks and Technologies (ANT)*, 2021.
- [7] Azise Oumar Diallo, Guillaume Lozenguez, Arnaud Doniec, and René Mandiau. Comparative evaluation of road traffic simulators based on modeler's specifications : An application to intermodal mobility behaviors. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART)*, volume 1, pages 265–272, 2021.
- [8] Sebastian Hörl and Milos Balac. Open data travel demand synthesis for agent-based transport simulation : A case study of Paris and île-de-france. 2020.
- [9] Sebastian Hörl, Milos Balać, and Kay W Axhausen. Pairing discrete mode choice models and agent-based transport simulation with matsim. In *TRB Annual Meeting*. Transportation Research Board, 2019.
- [10] Andreas Horni, Kai Nagel, and Kay Axhausen, editors. *Multi-Agent Transport Simulation MATSim*. Ubiquity Press, London, Aug 2016.
- [11] Joseph Kamel, Reza Vosooghi, Jakob Puchinger, Feirouz Ksontini, and Göknur Sirin. Exploring the impact of user preferences on shared autonomous vehicle modal split : A multi-agent simulation approach. *Transportation Research Procedia*, 37 :115–122, 2019.
- [12] Benjamin Kickhofer, Daniel Hosse, Kai Turnera, and Alejandro Tirachinic. Creating an open matsim scenario from open data : The case of santiago de chile. <http://www.vsp.tuberline.de/publication> : TU Berlin, *Transport System Planning and Transport Telematics*, 2016.
- [13] Daniel Krajzewicz, Matthias Heinrichs, and Sigrun Beige. Embedding intermodal mobility behavior in an agent-based demand model. *Procedia computer science*, 130 :865–871, 2018.
- [14] Jinjian Li, Arnaud Doniec, Jacques Boonaert, and Guillaume Lozenguez. Which traffic simulator is suitable for customized behaviors in multi-modal scenarios. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3791–3796. IEEE, 2018.
- [15] René Mandiau, Alexis Champion, Jean-Michel Auberlet, Stéphane Espié, and Christophe Kolski. Behaviour based on decision matrices for a coordination between agents in a urban traffic simulation. *Appl. Intell.*, 28(2) :121–138, 2008.
- [16] Kyriacos C Mouskos, Maria Boile, Neville Parker, et al. Technical solutions to overcrowded park and ride facilities. Technical report, New Jersey. Dept. of Transportation, 2007.
- [17] Rebekka Oostendorp and Laura Gebhardt. Combining means of transport as a users' strategy to optimize traveling in an urban context : empirical results on intermodal travel behavior from a survey in berlin. *Journal of Transport Geography*, 71 :72–83, 2018.
- [18] Flavio Poletti, Patrick M Bösch, Francesco Ciari, and Kay W Axhausen. Public transit route mapping for large-scale multimodal networks. *ISPRS International Journal of Geo-Information*, 6(9) :268, 2017.
- [19] Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.
- [20] Dominik Ziemke, Ihab Kaddoura, and Kai Nagel. The matsim open berlin scenario : A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data. *Procedia computer science*, 151 :870–877, 2019.

Approche multi-agent combinant raisonnement et apprentissage pour un comportement éthique ¹

R. Chaput^a J. Duval O. Boissier^b M. Guillermin^c
remy.chaput@univ-lyon1.fr jeremy-duval1@hotmail.fr Olivier.Boissier@emse.fr mguillermin@univ-catholyon.fr
S. Hassas^a
salima.hassas@univ-lyon1.fr

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

^bInstitut Henri Fayol,
MINES Saint-Etienne, France

^cSciences and Humanities Confluence Research Center, Lyon Catholic University

Résumé

Le besoin d'incorporer des considérations éthiques au sein d'algorithmes d'Intelligence Artificielle est de plus en plus présent. Combinant raisonnement et apprentissage, ce papier propose une méthode hybride, où des agents juges évaluent l'éthique du comportement d'agents apprenants. Le but est d'améliorer l'éthique de leur comportement dans des environnements dynamiques multi-agents. Plusieurs avantages découlent de cette séparation : possibilité de co-construction entre agents et humains ; agents juges plus accessibles pour des humains non-experts ; adoption de plusieurs points de vue pour juger un même agent, produisant une récompense plus riche. Les expérimentations sur la distribution de l'énergie dans un simulateur de Smart Grid montrent la capacité des agents apprenants à se conformer aux règles des agents juges, y compris lorsque les règles évoluent.

Mots-clés : *Éthique, Machine Ethics, Apprentissage Multi-Agent, Apprentissage par Renforcement, Hybride Neural-Symbolique, Jugement Éthique*

Abstract

The need to imbue Artificial Intelligence algorithms with ethical considerations is more and more present. Combining reasoning and learning, this paper proposes a hybrid method, where judging agents evaluate the ethics of learning agents' behavior. The aim is to improve the ethics of their behavior in dynamic multi-agent environments. Several advantages ensue from this separation : possibility of co-construction between agents and humans ; judging agents more accessible for non-experts humans ; adoption of several points of view to judge the same

agent, producing a richer feedback. Experiments on energy distribution inside a Smart Grid simulator show the learning agents' ability to comply with judging agents' rules, including when they evolve.

Keywords: *Ethics, Machine Ethics, Multi-Agent Learning, Reinforcement Learning, Hybrid Neural-Symbolic Learning, Ethical Judgment*

1 Introduction

Alors que le nombre d'applications utilisant des modèles d'Intelligence Artificielle (IA) augmente, il y a un débat de société et de recherche au sujet du moyen d'introduire des capacités éthiques dans ces modèles.

Le domaine des *Machine Ethics* s'intéresse à la conception d'"agents à impact éthique" [18] ayant un impact éthique sur des vies humaines ; en particulier, l'"éthique dans la conception" [15] vise à ce que ces agents prennent des décisions selon des considérations éthiques, ce que nous appellerons un "comportement éthique".

Cette demande de capacités éthiques pour des agents autonomes artificiels a été largement documentée [21], toutefois les moyens d'implémenter ces compétences ne sont pas clairs : certains travaux proposent des approches descendantes par raisonnement symbolique, tandis que d'autres préfèrent utiliser des approches ascendantes par apprentissage [1]. Les deux approches offrent différents avantages mais ont également des inconvénients ; ainsi, dans cet article, nous présentons une nouvelle approche, hybride [15], avec un apprentissage de comportements guidé par des récompenses issues de raisonnements symboliques.

¹Cet article est basé sur la traduction d'un article soumis à AIES2021.

Contribution scientifique

Ce papier est structuré comme suit : nous présentons d’abord la littérature sur laquelle nous appuyons notre approche hybride ; cette proposition est ensuite détaillée ; des expérimentations sur un cas d’application des *Smart Grids* et leurs résultats démontrent la faisabilité ; finalement, la dernière section compare l’approche à la littérature, examine les limitations actuelles et présente des perspectives.

2 Fondements

Afin d’identifier les principes de conception qui sous-tendent notre approche, nous explorons d’abord la littérature des *Machine Ethics*. Nous considérons ensuite le champ de l’IA Hybride (Neural-Symbolique), qui combine les méthodes symboliques et d’apprentissage.

2.1 Éthique et IA

La plupart des travaux existants en *Machine Ethics* se focalisent sur un unique agent isolé dans son environnement [26]. Nous arguons qu’il est important de considérer plusieurs agents en interaction dans un environnement commun, comme il s’agit d’une situation plus réaliste, qui soulève le problème de la confrontation de plusieurs éthiques.

Comme en IA, les travaux en *Machine Ethics* sont divisés en trois catégories [1] : approches descendantes, ascendantes et hybrides.

Les approches descendantes s’intéressent à la formalisation de principes éthiques, tel que l’Impératif Catégorique de Kant. En utilisant un raisonnement logique sur des représentations symboliques, de telles approches peuvent s’appuyer sur des connaissances expertes et offrir une meilleure lisibilité et explication des décisions prises. Par exemple, l’*Ethical Governor* [3] vérifie l’adéquation des actions avec des règles pré-établies comme les Règles d’Engagement ou le Droit de la guerre. Dans *Ethicaa* [13], des agents raisonnent sur plusieurs principes éthiques pour décider de leur comportement et juger les actions des autres agents. Cependant, ces approches descendantes, du fait de leur corpus de connaissances explicite mais figé, ne peuvent pas s’adapter, sans reconception, à des situations non prévues ou à une évolution de l’éthique.

Les approches ascendantes cherchent à apprendre un comportement à partir d’un jeu de données, e.g., des exemples étiquetés ou des expériences obtenues par interactions. Par

exemple, *GenEth* [2] utilise des décisions d’éthicistes dans de multiples contextes pour apprendre un principe éthique ; une autre approche utilise l’Apprentissage par Renforcement (RL) en ajoutant à la récompense de la tâche une composante éthique sous forme de différence entre les comportements de l’agent et d’un humain moyen, supposé exhiber des considérations éthiques [25]. Ces approches, bien qu’utilisant de l’apprentissage, n’ont pas considéré la question de l’adaptation sur le long-terme en réponse à des situations pouvant évoluer. De plus, les approches ascendantes sont plus difficiles à interpréter que les approches descendantes.

Finalement, les approches Hybrides couplent les approches descendantes et ascendantes, de telle sorte que les agents puissent apprendre des comportements éthiques par expérience tout en étant guidés par un cadre éthique existant afin de forcer des contraintes et les empêcher de diverger. Pour plus de détails, le lecteur peut se référer à [1, 15].

Nous discutons des différents moyens d’IA Hybride Neural-Symbolique et comment les intégrer dans un agent éthique dans la prochaine section.

2.2 Approches Hybrides

Les approches hybrides en IA visent à coupler le raisonnement symbolique avec l’apprentissage numérique pour bénéficier des avantages des deux approches en réduisant leurs inconvénients. Plusieurs manières pour les intégrer existent, voir par exemple [9]. Les auteurs avancent que les plans dans un agent BDI sont plus faciles à expliquer à un humain ; il est aussi admis qu’il est plus facile d’introduire des connaissances, par exemple d’un expert du domaine non-développeur, avec des règles symboliques. Des exemples d’approches Hybrides incluent SOAR-RL [19] ou BDI-RL [10], qui intègrent des algorithmes de RL avec du raisonnement. Plusieurs travaux ajoutent une couche de raisonnement symbolique, souvent BDI, par-dessus un agent artificiel [3, 11], et sont souvent qualifiés d’Hybride.

Le projet *Ethicaa* propose un système Multi-Agent dans lequel les agents juges déterminent un jugement sur les actions d’autres agents, en utilisant des croyances sur une situation donnée [13]. À notre connaissance, l’intégration d’un jugement symbolique pour donner une récompense numérique aux agents apprenants n’a pas

Contribution scientifique

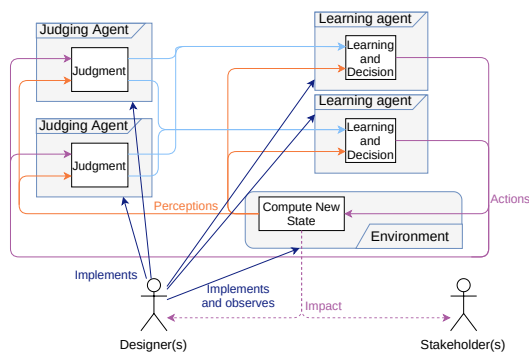


FIGURE 1 – Architecture de notre approche, comprenant des concepteurs humains implémentant des règles pour juger les agents apprenants. Les actions des apprenants modifient un environnement partagé, ce qui impacte les humains.

été étudié dans le domaine des Machine Ethics, mais nous pouvons appliquer les travaux d’Ethicaa au jugement d’agents utilisant de l’apprentissage numérique.

3 Modèle

Dans cette section, nous décrivons notre proposition, basée sur les principes de conception retenus de la littérature.

3.1 Architecture Abstraite

Considérons un système Multi-Agent comprenant des humains et des agents artificiels, représenté dans la Figure 1. Les concepteurs créent un environnement partagé et des agents autonomes afin qu’ils accomplissent des tâches ; les actions effectuées pour ce faire vont impacter l’environnement partagé et les humains. Le but des concepteurs est d’intégrer des *considérations éthiques* dans ces agents, afin de contraindre leur “impact éthique”, en accord avec un ensemble de *valeurs morales* sélectionnées par les concepteurs.

Nous nous focalisons sur une approche Hybride et plus particulièrement sur une séparation du jugement et de l’apprentissage, afin de permettre une co-construction. Pour cela, nous introduisons des agents juges et apprenants séparés, qui pourront évoluer indépendamment, par exemple pour mettre à jour les règles des juges par les concepteurs humains, tandis que les apprenants adaptent leur comportement en accord avec ces nouvelles règles.

De plus, nous proposons de rendre le jugement

plus riche en confrontant plusieurs valeurs morales, que nous représentons par des agents juges séparés afin de clarifier le modèle ; cela ouvre également la voie à des interactions plus complexes entre juges tels que des processus d’argumentation, et offre un moyen simple de changer les règles en ajoutant ou supprimant des agents juges. Ces agents juges sont basés sur les agents Ethicaa [13], manipulant un ensemble de règles morales pour raisonner et juger les actions des autres agents dans l’environnement. Les juges utilisent ces règles pour déterminer un jugement (e.g., “moral”, “immoral”), qui sont transformés en récompenses pour les apprenants. Ceux-ci l’intègrent dans leur processus d’apprentissage pour apprendre à effectuer de meilleures actions. La pluralité d’apprenants permettra d’évaluer leur impact dans un environnement partagé, plutôt qu’un agent isolé.

Finalement, le comportement attendu peut évoluer au cours du temps, du fait des dynamiques de notre société, les agents devraient donc être capables de s’adapter à des règles changeantes. Les concepteurs humains observent les actions des agents apprenants dans l’environnement et rectifient ces comportements indirectement en ajustant les règles à la base du jugement des agents juges. Il est ainsi possible d’envisager une approche d’IA centrée sur l’humain, avec un contrôle humain dans la boucle, comme le préconise le rapport Européen HLEG AI ¹.

Les valeurs et règles morales à la base des considérations éthiques sont clairement visibles dans ce modèle car choisies explicitement par les concepteurs. Nous faisons l’hypothèse que cela améliore l’intelligibilité du jugement et donc du comportement attendu, ce qui est reconnu comme important, notamment par le rapport du HLEG. Bien que ce point ne sera pas évalué dans les expérimentations, il sert en partie de motivation à notre approche ; les humains, y compris les utilisateurs, devraient pouvoir vérifier la compatibilité avec leurs propres principes éthiques.

3.2 Modèle Formel

Considérons l’ensemble J des agents *juges* : chaque agent $j \in J$ est associé à une seule valeur morale et un ensemble de règles morales permettant de décider si une action *soutporte* ou *trahit* cette valeur morale (e.g., la Justice, l’Inclusivité, la Sécurité). Dans le second ensemble L d’agents *apprenants*, chaque agent

1. <https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai>

Contribution scientifique

$l \in L$ apprend un comportement et effectue des actions dans l'environnement, en se basant sur l'évaluation F_l , agrégation des jugements $Jugement_j(l)$ des $j \in J$ sur le comportement de l : $\forall l \in L : F_l = \{\forall j \in J : Jugement_j(l)\}$.

Ce modèle d'apprentissage correspond conceptuellement à un jeu Markovien (ou *Stochastic Game*), une extension du Processus de Décision Markovien (MDP) à plusieurs agents. Formellement, il s'agit d'un n-uplet $(S, P, A_0, \dots, A_n, T, R_0, \dots, R_n)$. S est l'ensemble de tous les états possibles, sous forme de vecteurs de nombres réels (états multi-dimensionnels et continus) pour permettre des cas d'applications riches. P , de taille n , est l'ensemble de tous les agents; les agents juges n'agissant pas dans l'environnement, il correspond à L , l'ensemble des agents apprenants. A_l est l'ensemble des actions possibles pour l'agent l , également sous forme de vecteurs de nombres réels (actions paramétrées). L'ensemble des actions jointes $A = A_0 \times \dots \times A_n$ regroupe la combinaison des actions possibles des différents agents. T est la fonction de probabilité de transition, définie par $T : S \times A \times S \rightarrow [0, 1]$, i.e., $T(s, a, s')$ est la probabilité de passer de l'état s à s' en effectuant l'action a . R_l est la fonction de récompense pour l'agent l , définie par $R_l : S \times A_l \times S \rightarrow \mathbb{R}$, i.e., $R_l(s, a_l, s')$ est la récompense de l'agent l pour avoir effectué l'action a_l dans l'état s , résultant en l'état s' .

Les MDPs et jeux Markovien peuvent être résolus avec de l'Apprentissage par Renforcement (RL) [22], une méthode pour apprendre la probabilité $\pi(s, a)$ de sélectionner chaque action a dans chaque état s . Pour chacun des agents apprenants, le but de l'algorithme RL est d'apprendre la stratégie optimale, qui maximise l'espérance des récompenses reçues.

Traditionnellement, la fonction de récompense est une fonction mathématique qui indique si l'action exécutée était bonne, i.e. un objectif à optimiser. Nous voulons utiliser le jugement symbolique calculé par les agents juges; pour cela, la fonction de récompense agrège et transforme ces jugements en une valeur numérique. Nous décrivons d'abord les agents apprenants et le processus par lequel ils apprennent une stratégie optimale π , en mettant de côté les détails de R que nous décrivons ensuite.

3.3 Agents Apprenants

Les agents apprenants doivent apprendre comment sélectionner une action dans un état donné,

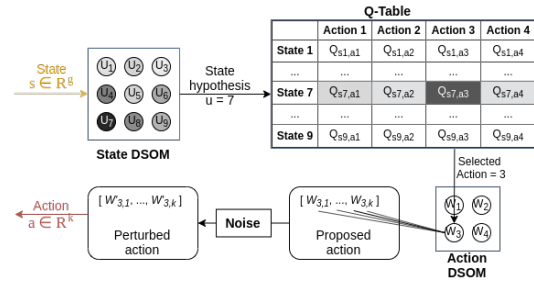


FIGURE 2 – Exemple de décision : l'agent reçoit un état, vecteur de réels, comparé à la State-DSOM. Le 7^{ème} neurone, qui a le vecteur prototype le plus proche, est choisi comme hypothèse d'état. À partir de la Q-Table et de ce 7^{ème} état, la 3^{ème} action est choisie. L'action obtenue est le résultat de la perturbation par un bruit aléatoire du vecteur associé au 3^{ème} neurone de l'Action-DSOM.

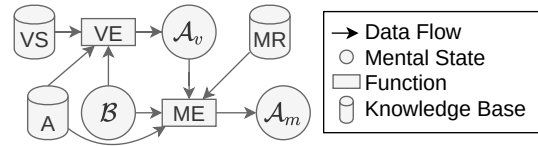


FIGURE 3 – Architecture des agents juges, adaptée d'Ethicaa [13].

afin de maximiser la récompense reçue sur l'ensemble des pas de temps. Nous utilisons l'algorithme Q-DSOM pour sa capacité à manipuler des états et actions multi-dimensionnels et continus [12]. Cet algorithme utilise deux *Dynamic Self-Organizing Map* (DSOM) [20], inspiré des Cartes Auto-Organisatrices de Kohonen [17], afin d'apprendre les espaces d'états (State-DSOM) et d'actions (Action-DSOM).

Les neurones des deux DSOMs sont liés à une Q-Table [24], de telle sorte que chaque neurone corresponde à un état ou une action discrets, i.e., une ligne ou une colonne dans la Q-Table. La Q-Table permet d'apprendre l'intérêt, ou Q-Value, d'une paire état-action, afin que l'agent puisse choisir la meilleure action pour chaque état.

Les agents peuvent donc représenter n'importe quel état ou action multi-dimensionnel et continu comme un identifiant discret via les DSOMs et utilisent la Q-Table pour déterminer l'intérêt associé. Se reporter à l'article originel [12] pour une description détaillée de l'algorithme. Un exemple est présenté dans la Figure 2.

Contribution scientifique

3.4 Agents Juges

L'architecture BDI des agents juges (cf. Figure 3) s'appuie sur [13] en simplifiant le mécanisme d'évaluation morale (des travaux futurs seront dédiés à enrichir cette composante).

À chaque pas de temps, les agents juges génèrent des croyances (B) à partir de leurs perceptions de l'environnement (les mêmes que les agents apprenants) et des actions effectuées par ces apprenants. Les agents juges traitent de manière séparée chacun des composants des vecteurs réels, correspondant aux paramètres des actions. En d'autres termes, les agents juges reçoivent un ensemble d'actions $\{\forall i \in [[1, k]] : a_{l,i}\}$, tel que a_l est un vecteur de k composants et $a_{l,i}$ est le i -ème composant du vecteur, i.e., un nombre réel.

L'agent juge utilise un ensemble fixé de valeurs et des règles morales associées (VS et MR) pour déterminer si chaque composant de l'action trahit ou supporte la valeur. L'Évaluation Morale (ME) utilise les croyances générées (B) et les actions (A) pour produire une valuation morale (Am), parmi l'ensemble $V = \{moral, immoral, neutral\}$. À chaque pas de temps, le jugement par un agent juge j de l'action a_l d'un agent apprenant l est l'évaluation morale de chacun des k paramètres de cette action a_l : $Jugement_j(l) = \{i \in [[1, k]] : ME_j(B, a_{l,i})\}$. Chaque agent juge calcule un jugement différent pour chaque apprenant, de sorte que la fonction d'évaluation finale $F : L \rightarrow (V^k)^{|J|}$ retourne une liste de listes de valuations, soit $\forall l \in L : F_l = \{\forall j \in J : Jugement_j(l)\}$.

3.5 Jugement pour l'apprentissage

Dans cette section, nous faisons le lien entre les symboles (jugements et croyances) manipulés par les agents juges et les nombres réels (perceptions, actions, et récompenses) utilisés dans l'algorithme Q-DSOM.

À la fin de chaque étape de la simulation, après que les agents apprenants aient choisi leur action, les agents juges perçoivent plusieurs données et les transforment en croyances : liste des agents, les actions de chacun, les propriétés globales représentant l'état de l'environnement et les propriétés individuelles par agent.

Finalement, la fonction de récompense $R_l : S \times A_l \times S \rightarrow \mathbb{R}$ retourne un nombre réel. Plusieurs méthodes sont possibles pour transformer les valuations symboliques en une récompense

numérique ; dans cette première étape, nous proposons la solution suivante. Nous transformons le jugement de chaque agent juge, i.e., une liste de valuations, en un nombre unique, en comptant le nombre de valuations *moral* et en divisant par le nombre de valuations *moral* et *immoral*, de telle sorte que les actions morales tendent vers 1, tandis que les actions immorales tendent vers 0. Comme cas spécial, si la liste ne consiste que de valuations *neutral*, nous considérons que l'action était ni bonne ni mauvaise, et nous la mettons à 0.5. La récompense finale est calculée comme la moyenne des récompenses de chaque agent juge.

On peut remarquer que cette méthode permet de résoudre de manière simple les conflits entre les agents juges ; par exemple, le premier agent juge peut juger que le premier composant de l'action est moral selon sa propre valeur morale, tandis qu'un second juge peut déterminer que ce même premier composant est immoral, en accord avec sa valeur morale (différente).

4 Expérimentations

Afin de tester la validité de notre approche, nous reprenons le cas d'application dans [12] : il s'agit d'une micro-grille électrique hypothétique, dans laquelle la production d'énergie est décentralisée au lieu de reposer uniquement sur le réseau national. La grille possède une source d'électricité principale (e.g., une station hydro-électrique, ou une ferme à éoliennes) ; les utilisateurs, ou *prosumers* (producteurs-consommateurs), peuvent eux-mêmes produire une petite quantité d'énergie (e.g., via des panneaux photovoltaïques). Considérant la difficulté de stocker de grandes quantités d'énergie sur une longue période, et que la production et la demande peuvent fluctuer sur de courtes périodes, les *prosumers* peuvent échanger de l'énergie afin de ne pas la gaspiller. De tels échanges supposent une forme de coopération pour éviter les situations d'inégalité ; de manière similaire, quand la source principale est trop sollicitée, les *prosumers* doivent réduire leur consommation temporairement, et ainsi réduire leur confort, afin d'éviter des coupures.

Les simulations considèrent un ensemble de bâtiments (type habitations, bureaux et écoles ; voir la Figure 4) : la gestion de l'énergie de chaque bâtiment est prise en charge par un agent apprenant. Il doit apprendre à consommer et échanger de l'énergie pour satisfaire le besoin en confort de ses occupants, tout en considérant les intérêts des autres *prosumers* de la grille.

Contribution scientifique

Nous considérons ce simulateur simplifié et l'opposition d'intérêts entre les différents participants comme étant suffisamment plausibles et un cadre intéressant pour des comportements éthiques.

4.1 Règles et Valeurs Morales

Nous avons choisi des valeurs morales à partir de la littérature des réseaux électriques intelligents [4] et traduit ces valeurs afin de refléter le point de vue de citoyens participant à un tel système, et prenant des décisions pour allouer de l'énergie. En effet, les agents apprenants agissent en tant que mandataires pour ces *prosumers* et doivent donc soutenir les mêmes valeurs morales. Nous proposons quatre valeurs morales et les règles associées comme références communes pour tous les agents de la simulation : MR1 — Assurance de confort : une action permettant à un *prosumer* d'améliorer son confort est morale ; MR2 — Affordabilité : une action qui coûte trop cher à un *prosumer*, par rapport au budget configuré par l'utilisateur, est immorale ; MR3 — Inclusion sociale : une action qui améliore l'équité des confort entre les *prosumers* est morale ; MR4 — Viabilité Environnementale : une action qui limite les échanges avec le réseau national est morale.

Ces règles sont partiellement en conflit : par exemple, les agents consomment de l'énergie pour satisfaire le confort des utilisateurs, en accord avec MR1, mais il n'y a pas assez d'énergie pour satisfaire tous les agents, ce qui trahit la valeur associée à MR3 ; acheter de l'énergie transgresserait MR4. En d'autres termes, chaque action implique une transgression d'au moins une règle morale, ce qui classe cette simulation comme un dilemme éthique selon la définition de Bonnemains [6].

Puisque les actions a_i sont des vecteurs de réels, le but des agents apprenants est de déterminer les bons paramètres, i.e., les composants du vecteur, afin de minimiser leur regret. En d'autres termes, la question pourrait être, par exemple, "Quelle quantité d'énergie devrais-je acheter afin de minimiser la transgression de MR4 tout en me permettant d'améliorer mon confort, en accord avec MR1 ?".

4.2 Simulateur

Le simulateur que nous utilisons est illustré dans la Figure 4 ; nous résumons ses composants ci-après.

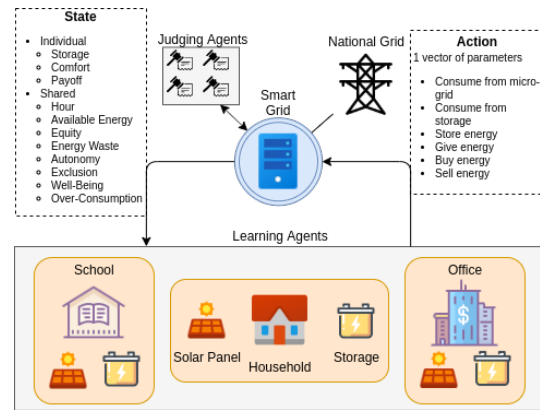


FIGURE 4 – Schéma du simulateur de *Smart Grid*. Une micro-grille, liée à la grille nationale, contient plusieurs agents représentant des bâtiments (Habitations, Bureaux, ou Écoles). Ils reçoivent un vecteur décrivant l'état de la micro-grille et effectuent une action avec plusieurs paramètres permettant, e.g., de consommer de l'énergie.

Agents Apprenants. Trois profils d'agents ont été créés pour répondre aux types de bâtiment et introduire de la variété entre les agents : Habitation, Bureau et École. Chaque profil est constitué : d'un profil de consommation, i.e. la quantité d'énergie dont il a besoin à chaque heure ; d'une courbe de confort, i.e. la fonction qui calcule son confort pour une consommation et un besoin donnés ; d'une capacité d'action, e.g. la quantité maximale d'énergie qu'il peut consommer ; et d'une capacité de stockage personnelle.

Nous utilisons un jeu de données publique de consommation d'énergie² comme source des profils de consommation. Trois bâtiments ont été sélectionnés : *Residential*, *Small Office* et *Primary School* ; chacun dans la même ville (Anchorage) afin de minimiser le risque de biais. Le jeu de données contient la charge horaire, i.e. la quantité d'énergie consommée par un bâtiment pour chaque heure sur une année ; tandis que le travail précédent utilisait un profil moyenné sur tous les jours de l'année (profil journalier), nous avons retenu le profil annuel complet. Les courbes de confort et le besoin en énergie par heure sont visibles dans la Figure 5.

Actions. À chaque pas de temps, chaque agent apprenant effectue une action, vecteur de para-

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

Contribution scientifique

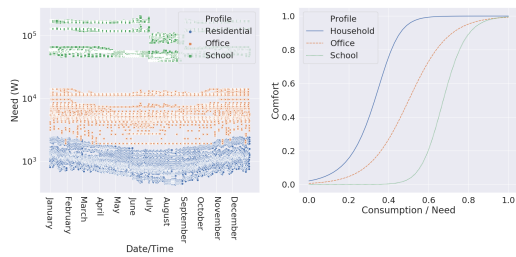


FIGURE 5 – Le besoin et le confort pour chaque profil d’agent.

mètres suivants : la quantité d’énergie consommée depuis la micro-grille, la quantité consommée depuis la batterie personnelle, l’énergie stockée dans sa batterie depuis la grille et inversement la quantité donnée de la batterie vers la micro-grille. Si la grille ne dispose pas d’assez d’énergie, l’excès est automatiquement acheté depuis la grille nationale pour éviter une coupure, mais dans ce cas l’excès est considéré comme sur-consommé. L’agent peut également interagir avec la grille nationale en achetant ou vendant de l’énergie dans sa batterie personnelle.

Perceptions. Afin de prendre une décision, l’agent obtient l’état de l’environnement représenté par un vecteur de nombres réels. Ces perceptions incluent des données communes partagées par tous les agents : l’heure, l’énergie disponible, l’équité des confort (calculée comme une dispersion statistique par l’index de Hoover), la quantité d’énergie non-utilisée et donc perdue, l’autonomie (l’absence de transactions avec le réseau national), le bien-être (médiane des confort), l’exclusion (la proportion d’agents dont le confort est inférieur à 50% du bien-être), la sur-consommation. Les agents perçoivent en plus des détails sur eux-mêmes, auxquels les autres agents n’ont pas accès : la quantité d’énergie disponible dans la batterie personnelle, le confort au pas de temps précédent, et le bénéfice obtenu en vendant et achetant de l’énergie.

Récompenses. Comme décrit dans notre modèle, les récompenses sont calculées à partir des jugements des agents juges. Ceux-ci sont implémentés en langage Jason [8] sur la plateforme JaCaMo [5], qui est implémentée en Java. Une API REST permet d’assurer la communication entre les agents apprenants, implémentés en Python, et les agents juges sur JaCaMo.

Nous avons implémenté quatre agents juges, un pour chacune des valeurs morales proposées, contenant des règles dans

un langage pseudo-Prolog, par exemple `supporte(donne_energie(X) :- X > 0`, qui signifie que l’action de donner une quantité X d’énergie supporte la valeur associée (dans ce cas, Viabilité Environnementale) si la quantité est positive. De manière similaire, des règles “trahit” déterminent si l’action trahit la valeur.

5 Résultats

Nous avons mené plusieurs simulation, en considérant différents paramètres. En variant le nombre d’agents apprenants, les simulations “Petit” (20 Habitations, 5 Bureaux, 1 École) et “Moyen” (80 Habitations, 19 Bureaux, 1 École) permettent d’évaluer le passage à l’échelle de notre approche. Les simulations “Journalier” utilisent le profil de consommation moyenné sur une seule journée, tandis que les simulations “Annuel” utilisent le jeu de données complet, introduisant ainsi des variations saisonnières et donc plus de difficultés pour l’apprentissage. Enfin, nous proposons sept scénarios pour la configuration des agents juges, dont quatre qui incluent un seul agent juge (nommés “mono-valeur”), un dans lequel les juges sont activés un par un à des pas de temps différents (“Incrémental”), un dans lequel les juges sont initialement tous actifs et désactivés un par un à des pas de temps différents (“Décémental”) et un scénario “Défaut” dans lequel les quatre agents sont activés en permanence. Cette variété de scénarios permet de comparer la présence et l’absence de chaque règle morale, l’impact sur les autres règles, et la capacité des agents apprenants à s’adapter quand les règles évoluent au fil du temps, soit en les ajoutant soit en les enlevant. Chaque ensemble de simulations a été lancé 20 fois sur 10 000 pas de temps.

La comparaison entre les expérimentations “Petit” et “Moyen” n’a pas montré de réelle différence entre les moyennes (T-Test, p-value=0.83), ce qui indique que notre approche passe à l’échelle, bien que le temps d’exécution soit naturellement bien plus long.

Nous nous concentrons sur les scénarios “Défaut” et “Incrémental” car ils sont les plus pertinents ; les “mono-valeurs” sont utiles en tant que scénarios de contrôle pour comparer les effets d’une valeur morale sur le comportement des agents quand la valeur est isolée ou agrégée avec d’autres. Le scénario “Décémental” montre la capacité de supprimer des règles mais n’est pas aussi intéressant que la capacité d’en ajouter (le scénario “Incrémental”). Nous nous

Contribution scientifique

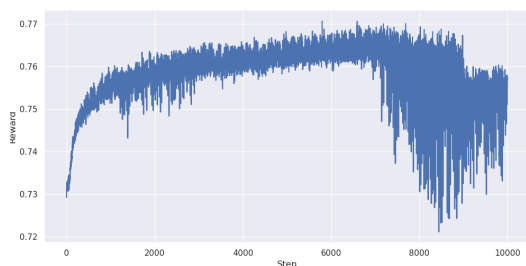


FIGURE 6 – Récompense moyenne pour tous les agents apprenants, sur chaque pas de temps, dans les simulations “Petit - Annuel - Défaut”.

concentrons également sur le profil “Annuel” comme il contient des variations de saisons et donc plus de difficultés.

La figure 6 montre que la récompense moyenne augmente au fur et à mesure de la simulation ; la méthode proposée est donc efficace et les agents apprenants sont capables de se conformer aux règles données. Toutefois, les récompenses chutent vers la fin de la simulation, en grande partie à cause de la récompense d’“Inclusion”, qui semble plus difficile à apprendre. Il n’est pas clair s’il s’agit d’un problème lié à l’algorithme d’apprentissage ou à l’implémentation proposée des règles morales. Nous remarquons que la valeur d’“Inclusion” est celle avec le plus grand nombre de règles implémentées ; peut-être que cela est lié à son apparente difficulté d’apprentissage.

De plus, la Figure 7 montre que les agents ont été capables d’apprendre la valeur de “Viabilité Environnementale”, et en particulier quand ils ne disposaient que de la récompense agrégée, bien que la variation n’était pas aussi importante que quand ils disposaient spécifiquement de cette valeur comme récompense. Il est intéressant de noter que la comparaison entre “Défaut” et “Incrémental” montre que l’addition un par un des Juges semble mitiger l’impact négatif d’“Inclusion”. Les agents sont encore capables d’apprendre la “Viabilité Environnementale” et performant légèrement mieux sur l’“Inclusion”.

Selon ces figures, les agents *Écoles* ont eu les plus grandes variations dans les récompenses, tandis que les *Habitations* et *Bureaux* avaient une augmentation plus stable. Ce n’est pas surprenant, car les agents *École* ont la plus grande capacité d’action et ont donc un impact plus important sur l’environnement.

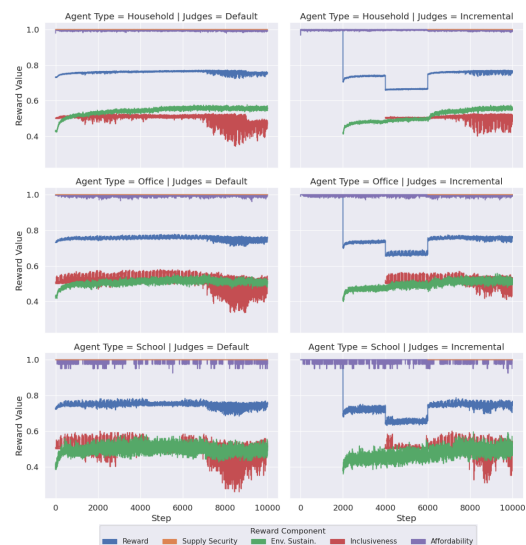


FIGURE 7 – Comparaison entre les récompenses individuelles de chaque agent apprenant, au fil des pas de temps, moyennées sur les simulations “Petit - Annuel - Défaut” et “Petit - Annuel - Incrémental”, et sur les agents de même profil. La courbe “Récompense” est la moyenne des quatre composantes.

6 Discussion

Pour rappel, notre contribution est une nouvelle méthode pour apprendre des “comportements éthiques”, i.e. des comportements qui exhibent une ou plusieurs valeurs morales et qui seraient considérés comme éthiques d’un point de vue humain, en utilisant des jugements symboliques comme sources de récompenses pour les agents apprenants dans un système multi-agent. Nous avons évalué cette approche sur un problème de répartition d’énergie, dans un contexte de *Smart Grid* simulé. Les expérimentations menées et en cours servent de preuve de concept pour montrer l’intérêt de notre approche.

Par rapport à la littérature existante, cette approche offre plusieurs avantages. Tout d’abord, il est important de noter que l’acceptation courante de la société sur l’éthique peut évoluer au fil du temps ; ainsi, les approches qui visent l’“éthique par conception” doivent considérer la capacité du système à s’adapter à des règles changeantes. Cet aspect n’a pas été extensivement étudié, y compris parmi les travaux se focalisant sur l’apprentissage ; dans cet article, nous avons montré grâce aux scénarios “Incrémental” et “Décrémental” la capacité de nos agents à s’adapter à l’ajout ou la suppression de règles. Cela est par-

Contribution scientifique

ticulièrement visible en se comparant au travail de *reward-shaping* discuté précédemment [25] : si le comportement exemple n'est plus en accord avec le comportement attendu, il leur serait probablement nécessaire de re-cr  er un nouveau jeu de donn  es et d'entra  ner l'agent depuis z  ro. Dans notre cas, nous pouvons simplement ajouter ou supprimer les r  gles. Toutefois, un avantage de leur approche est qu'ils supposent une r  compense   thique qui n'est pas sp  cifique    la t  che, tandis que nos r  gles morales sont sp  cifiques aux domaines. Il serait peut-  tre possible d'impl  menter des r  gles morales plus g  n  riques, toutefois cela requiert l'existence de telles r  gles ; une possible source d'inspiration peut   tre les nombreux principes directeurs propos  s sur *Ethical AI* ou *Responsible AI* [21].

Les pr  c  dents cas d'applications pour des "agents   thiques"   taient limit  s    des actions discr  tes (e.g., dilemmes tel que le Dilemme du Tramway [23], robot accompagnant, [2], soldats robots [3], gestion autonome d'actifs en bourse [14]). Ces travaux sont importants mais il existe de nombreuses situations requ  rant de plus fines actions ; il est ainsi important de proposer et d'exp  rimer sur des environnements avec actions continues tel que le simulateur utilis   ici.

Certains travaux proposent d'utiliser des m  thodes de v  rification formelle pour garantir la conformit   aux r  gles morales dans n'importe quelle situation identifi  e [11, 7]. Dans notre cas, l'introduction d'agents apprenants nuit    cette possibilit   ; il existe toutefois des travaux qui tentent d'appliquer de la v  rification formelle    des algorithmes de RL [16].

Ce travail cible les consid  rations "par conception", mais il y a   galement d'autres implications    l'  ventuelle int  gration d'un tel syst  me dans la soci  t  . En effet, nous pouvons noter au moins un impact positif et un n  gatif : d'un c  t  , l'utilisation de r  gles symboliques est suppos  e plus facile    comprendre qu'une fonction math  matique. Toutefois, l'intelligibilit   n'  tait pas l'objectif principal de ce travail et n'  tait pas   valu  e par nos exp  rimentations. Nous pensons que l'intelligibilit   du processus de r  compense est cruciale, en particulier pour de la supervision humaine, que ce soit par les concepteurs du syst  me ou des r  gulateurs externes. Il s'agit ainsi d'un point important    consid  rer et am  liorer dans de futurs travaux.

D'un autre c  t  , les jugements n  cessitent de nombreuses donn  es sur les agents apprenants, e.g. leurs actions, leurs perceptions, ce qui en-

trave leur vie priv  e. Il pourrait   tre possible de limiter les donn  es   chang  es en offrant des jugements limit  s, ou d'anonymiser les donn  es pour que les juges ne puissent pas identifier les agents. Dans cet article, nous avons simplement consid  r   que les donn  es   taient librement accessibles.

Notre approche a toutefois quelques limites. Prem  rement, les r  gles morales utilis  es servent de preuve de concept pour montrer l'int  r  t de notre approche hybride, mais il serait int  ressant d'  tendre les agents juges et leurs r  gles afin de juger des situations plus complexes. Deuxi  mement, la m  thode utilis  e pour transformer les jugements symboliques en r  compense num  rique par association des symboles    des nombres pour prendre la moyenne permet de facilement r  soudre les conflits entre r  gles, mais d'autres m  thodes sont possibles, en particulier un m  canisme d'argumentation entre juges de sorte      tablir une priorit   entre les r  gles selon le contexte. Par exemple, imaginons le cas o   un h  pital est en manque crucial d'  nergie, la r  gle interdisant l'achat d'  nergie selon la valeur de Viabilit   Environnementale pourrait   tre mise de c  t   dans ce cas pr  cis.

Remerciements

Ce travail a   t   financ   par la R  gion Auvergne Rh  nes-Alpes (AURA), au sein du projet Ethics.AI (Pack Ambition Recherche). Les auteurs remercient leurs partenaires dans ce projet.

R  f  rences

- [1] Colin ALLEN, Iva SMIT et Wendell WALLACH. "Artificial Morality : Top-down, Bottom-up, and Hybrid Approaches". en. In : *Ethics and Information Technology* 7.3 (sept. 2005), p. 149-155. ISSN : 1572-8439.
- [2] Michael ANDERSON, Susan Leigh ANDERSON et Vincent BERENZ. "A Value-Driven Eldercare Robot : Virtual and Physical Instantiations of a Case-Supported Principle-Based Behavior Paradigm". In : *Proc. IEEE* 107.3 (2019), p. 526-540.
- [3] Ronald C ARKIN, Patrick D ULAM et Britany DUNCAN. *An ethical governor for constraining lethal action in an autonomous system*. Rapp. tech. Georgia Institute of Technology, 2009.

Contribution scientifique

- [4] Anne BOIJMANS. “The Acceptability of Decentralized Energy Systems”. Mém. de mast. Delft University of Technology, juil. 2019.
- [5] Olivier BOISSIER et al. *Multi-Agent Oriented Programming : Programming Multi-Agent Systems Using JaCaMo*. The MIT Press, 2020.
- [6] Vincent BONNEMAINS. “Formal ethical reasoning and dilemma identification in a human-artificial agent system.” Thèse de doct. Institut supérieur de l’aéronautique et de l’espace, Toulouse, France, 2019.
- [7] Grégory BONNET, Bruno MERMET et Gaële SIMON. “Vérification formelle et éthique dans les SMA.” In : *JFSMA*. 2016, p. 139-148.
- [8] Rafael H BORDINI, Jomi Fred HÜBNER et Michael WOOLDRIDGE. *Programming multi-agent systems in AgentSpeak using Jason*. T. 8. John Wiley & Sons, 2007.
- [9] Rafael H BORDINI et al. “Agent programming in the cognitive era”. In : *Autonomous Agents and Multi-Agent Systems* 34 (2020).
- [10] Michael BOSELLO et Alessandro RICCI. “From programming agents to educating agents—a jason-based framework for integrating learning in the development of cognitive agents”. In : *International Workshop on Engineering Multi-Agent Systems*. Springer. 2019, p. 175-194.
- [11] Paul BREMNER et al. “On proactive, transparent, and verifiable ethical reasoning for robots”. In : *Proceedings of the IEEE* 107.3 (2019), p. 541-561.
- [12] Rémy CHAPUT et al. “Apprentissage adaptatif de comportements éthiques”. In : *28e Journées Francophones sur les Systèmes Multi-Agents (JFSMA’2020)*. Cépaduès. 2020.
- [13] Nicolas COINTE, Grégory BONNET et Olivier BOISSIER. “Jugement éthique dans les systèmes multi-agents.” In : *JFSMA*. 2016, p. 149-158.
- [14] Nicolas COINTE, Grégory BONNET et Olivier BOISSIER. “Multi-agent based ethical asset management”. In : *1st Workshop on Ethics in the Design of Intelligent Agents*. 2016, p. 52-57.
- [15] Virginia DIGNUM. *Responsible Artificial Intelligence : How to Develop and Use AI in a Responsible Way*. Springer Nature, 2019.
- [16] Nathan FULTON et André PLATZER. “Safe reinforcement learning via formal methods : Toward safe control through proof and learning”. In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 32. 1. 2018.
- [17] Teuvo KOHONEN. “Essentials of the self-organizing map”. In : *Neural Networks* 37 (2013), p. 52-65.
- [18] James H MOOR. “The nature, importance, and difficulty of machine ethics”. In : *IEEE intelligent systems* 21.4 (2006), p. 18-21.
- [19] Shelley NASON et John E LAIRD. “Soar-RL : Integrating reinforcement learning with Soar”. In : *Cognitive Systems Research* 6.1 (2005), p. 51-59.
- [20] Nicolas P. ROUGIER et Yann BONIFACE. “Dynamic self-organising map”. In : *Neurocomputing* 74.11 (2011), p. 1840-1847.
- [21] Daniel SCHIFF et al. “What’s Next for AI Ethics, Policy, and Governance? A Global Overview”. In : *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, p. 153-158.
- [22] Richard S. SUTTON et Andrew G. BARTO. “Reinforcement Learning : An Introduction”. In : *IEEE Trans. Neural Networks* 9.5 (1998), p. 1054-1054.
- [23] Judith Jarvis THOMSON. “Killing, letting die, and the trolley problem”. In : *The Monist* 59.2 (1976), p. 204-217.
- [24] Christopher J. C. H. WATKINS et Peter DAYAN. “Q-Learning”. en. In : *Machine Learning* 8.3 (mai 1992), p. 279-292. ISSN : 1573-0565.
- [25] Yueh-Hua WU et Shou-De LIN. “A low-cost ethics shaping approach for designing reinforcement learning agents”. In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 32. 1. 2018.
- [26] Han YU et al. “Building Ethics into Artificial Intelligence”. In : *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI’18. Stockholm, Sweden : AAAI Press, juil. 2018, p. 5527-5533. ISBN : 978-0-9992411-2-7.

Contribution scientifique

composer de manière adéquate la modélisation d'un système composé d'une multitude d'agents autonomes. Nous nous tournons naturellement vers une méthode de résolution de problèmes incrémentale régie par une optimisation continue de l'interaction entre les éléments locaux : les systèmes multi-agent. Pour ce faire, nous avatarisons les composants en agents virtuels [13]. L'objectif des agents est décidé par l'utilisateur qui peut demander au système de concevoir un produit. Ce faisant, il donne aux agents leurs désirs (*e.g.* un agent peut souhaiter faire partie d'une coalition avec un autre agent à bas prix). Ce problème de formation de groupe est appelé dans la littérature "problème de formation de coalition" (CF) [16]. Comme les agents sont des entités autonomes, nous les considérons comme des humains qui ont des désirs, des croyances et des objectifs. C'est pourquoi nous nous inspirons des sciences humaines et sociales (SHS) et de la dynamique de groupe pour aborder notre problème de CF. Cependant, il s'agit d'un vaste problème comportant de nombreuses variantes. Comme nous le verrons dans les sections suivantes, nous nous concentrons plus spécifiquement sur le problème de la génération des structures de coalitions (CSG).

Nous présentons ABSG (Attraction Based Structures Generation), un modèle d'agent d'inspiration sociale conçu pour traiter un problème de formation de coalitions décentralisé dans un système ouvert et variable. La partie 2 fait une brève présentation de la littérature sur le sujet. La partie 3 définit le problème. La partie 4 présente les propriétés et la prise de décision de notre modèle d'agent. Nous présentons l'opérationnalisation de notre modèle dans la partie 5 et discutons de nos résultats expérimentaux dans la partie 6.

2 Revue de littérature

Le problème de CSG produit des groupes d'agents qui ont trois principales caractéristiques : (1) ils sont axés sur un objectif et ont une durée de vie courte, (2) ils sont formés avec un but et se séparent lorsque ce but n'existe plus et (3) les structures organisationnelles sont généralement plates [16]. Plusieurs approches sont utilisées pour aborder le problème de CSG, telles que, de manière non exhaustive, la programmation dynamique, le choix social informatique, l'intelligence distribuée, le multi-agent, etc.

Les algorithmes de programmation dynamique sont des algorithmes créant des coalitions optimales. Le problème de formation de coalition

est traité par les CSP (Complete set partitioning) dans lesquels l'objectif est de trouver, à partir d'un ensemble d'entités fini, un ensemble de coalitions dont la somme des score est maximisé. Le premier algorithme de programmation dynamique pour un problème de partitionnement d'un ensemble a été proposé par [20]. Il a plus tard servi d'inspiration à Michalak *et al.* dans [12] qui ont réalisé un des algorithmes les plus rapide pour résoudre le problème de CSP, ODP-IP. Bien que trouvant des solutions optimales, les algorithmes produits dans cette discipline sont très sensibles à l'explosion combinatoire et ne peuvent traiter des problèmes de formation de coalitions possédants plus d'une trentaine d'agents.

Le choix social s'intéresse à la conception et l'analyse de méthodes de prise de décisions collectives. Le problème de formation de coalition est aussi référé comme un problème d'assortiment (*matching*) dans lequel on peut affecter des agents à d'autres agents mais aussi des agents à des ressources. Cette seconde alternative est très similaire au problème d'allocation de tâches [2]. Le problème de l'assortiment est introduit par David Gale et Lloyd Shapley [6]. Dans ce travail les auteurs proposent un algorithme de mariages stables dans lequel l'objectif est de regrouper des hommes et des femmes deux à deux de manière à respecter au mieux les préférences de tous. Le problème est plus tard étendu par le problème hôpitaux/résidents [11] [4]. Il est similaire à celui des mariages stables. Cependant, ce n'est plus un problème liant deux éléments de deux ensembles, mais un élément d'un ensemble à plusieurs éléments d'un autre ensemble. Ce problème peut être rapproché à celui de la formation de coalition où les hôpitaux seraient des coalitions d'un nombre d'individus prédéfini. Contrairement au problème de formation de coalition comme nous l'abordons, ce ne sont pas les membres qui veulent s'associer à d'autres membres mais les groupes qui possèdent des préférences sur les futurs membres.

L'intelligence distribuée se concentre sur la coordination dans l'espace de multiples entités, appelées agents ou particules, sans contrôle centralisé. Les agents permettent via leur comportement le calcul distribué d'une solution à un problème de coordination. L'approche la plus courante consiste à analyser la formation et la stabilité d'un unique essaim constitué de quelques dizaines d'individus homogènes. Les particules sont dites homogènes lorsque celles-ci sont identiques les unes aux autres et que rien ne permet

Contribution scientifique

de les distinguer. Au contraire, elles sont dites hétérogènes lorsqu'elles ont leurs propres caractéristiques modifiant leur comportement. Kumar *et al.* présentent une méthode parvenant à doter les agents d'un essaim d'un comportement ségréatif [10]. Leur méthode se base un champs de potentiel appliqué sur chaque agent leur permettant de les agréger en un essaim stable. Inácio *et al.* utilisent une stratégie inspirée de l'optimisation par essaims particuliers (PSO) pour la ségrégation d'agents hétérogènes [8]. De la même manière que dans [10], les agents possèdent un type leur servant à identifier les individus avec lesquels former un essaim. Bien que plus adapté à un passage à l'échelle que les méthodes de programmation dynamique, notre problème de formation de coalitions ne consiste pas à regrouper des agents selon un type en plusieurs groupes disjoints mais de faire émerger des coalitions, possiblement chevauchantes, en fonction des caractéristiques et des désirs variables des agents du système.

Le multi-agent s'intéresse aussi au problème de formation de coalition. [17] présente la méthode SACF permettant de construire des coalitions dans un système ouvert. SACF est une méthode dynamique appliquée à un scénario de réseau électriques intelligent où des agents peuvent former des coalitions afin de fournir de l'énergie à un réseau. Dans ce scénario, les agents doivent former des coalitions en interagissant uniquement avec le voisinage proche ce qui permet de limiter une explosion combinatoire liée aux problèmes de CSG. Les auteurs évaluent leur méthode jusqu'à 70 agents et montrent qu'elle est robuste à l'entrée et à la sortie des agents. En outre, leur méthode permet de former des coalitions d'une qualité très proche de l'optimum dans des expériences à petite échelle. Bien que leur approche donne de bons résultats, la dynamique de leur système est limitée. En effet, l'énergie que les agents peuvent apporter au réseau ne peut pas varier pendant l'expérience. La réorganisation des coalitions n'est donc pas traitée par cette méthode. Dans [1], les auteurs présentent deux méthodes pour construire des coalitions basées sur les caractéristiques et les préférences de plusieurs agents qui réduisent le nombre d'échecs dans la recherche de coalitions. Leur méthode permet à l'agent de prendre des décisions en fonction de préférences qui sont agrégées avec une intégrale de Choquet. Ce travail présente des similitudes avec le notre puisque nous utilisons également une notion de préférences, appelée attraction dans la suite. Cependant, leur objectif et les contraintes utilisées sont différentes.

Alors que nous essayons de faire émerger les meilleures coalitions possible dans un système multi-agent ouvert et variable, ils essaient d'empêcher leur méthode de ne pas trouver de coalition dans un système multi-agent non variable. Même si leur méthode empêche l'échec dans le processus de CF, ils n'évaluent pas la qualité des coalitions formées.

3 Définition du problème

Les agents ont leurs propres caractéristiques et désirs qui peuvent être modifiés à tout moment. Au lancement du système, chaque individu ignore les caractéristiques et les désirs des agents. Leur objectif est de former des coalitions avec les agents qui répondent le mieux à leurs désirs. Pour ce faire, les agents peuvent interagir et échanger leurs caractéristiques et leurs désirs. Les agents utilisent ces informations pour évaluer leurs accointances en calculant une valeur d'attraction. L'attraction représente la mesure dans laquelle un agent est attiré par une de ses accointances en fonction de plusieurs facteurs (le traitement de l'attraction est expliqué dans les parties suivantes). Plus la valeur est faible, plus l'attrait est important. Les agents ont la possibilité de former, de rejoindre ou de quitter des coalitions en fonction de la valeur d'attraction qu'ils ont pour les autres agents et pour leurs coalitions.

Dans le problème de CSG, nous considérons un ensemble d'agents N tel que $Agents = \{a_1, \dots, a_n\}$ dans un système ouvert où les agents peuvent entrer ou sortir à tout moment. Soit CS l'ensemble des coalitions réalisées par les agents de telle sorte que $CS = \{Co_1, Co_2, \dots, Co_n\}$. Une coalition Co_i est définie comme un tuple : $Co_i = \langle i, A, s \rangle$ où i est l'identifiant du groupe, $A \subset Agents$ et $|A| \geq 2$ et s est un agent leader avec $s \in A$. Un leader est un chef de coalition qui donne l'autorisation aux autres membres de la modifier lorsque ceux-ci veulent proposer à un membre extérieur de la rejoindre ou s'il souhaite la quitter. La représentation des coalitions n'étant pas centralisée, son rôle permet aux membres de maintenir une représentation correct de leur coalition. C et D sont des ensembles de caractéristiques et de désirs tels que $C = \{c_1, c_2, \dots, c_m\}$, $D = \{d_1, d_2, \dots, d_m\}$ et $|C| = |D|$. Une caractéristique c_i est la satisfaction d'un désir d_i .

Chaque agent est défini comme un tuple : $a = \langle C^a, D^a, Acq^a, V^a, Coal^a, V^{ac} \rangle$ où :

Contribution scientifique

- C^a est un ensemble de caractéristiques : $C^a = \{c_1^a, c_2^a, \dots, c_n^a\}$ tel que $C^a \subset C$;
- D^a est un ensemble de désirs : $D^a = \{d_1^a, d_2^a, \dots, d_n^a\}$ appartenant à l'agent tel que $D^a \subset D$;
- Acq^a est une liste d'agents. Elle représente ses accointances ;
- VA^a est la liste des valeurs d'attraction liées à ses accointances telle que la valeur d'attraction pour l'accointance Acq_i^a est VA_i^a et $VA_i^a \geq 0$;
- $Coal^a$ est la liste des coalitions auxquelles l'agent appartient ;
- VC^a est la liste des valeurs d'attraction liées à ses coalitions de sorte que la valeur d'attraction pour la coalition $Coal_i^a$ est VC_i^a et $VC_i^a \geq 0$.

Les agents choisissent de former ou de rejoindre une coalition avec d'autres agents qui répondent à leurs désirs de telle sorte que $C_{a_j} \subset D_{a_i}$. Les agents sont égoïstes, ils essaient d'atteindre leur propre but. Ils ne sont pas conscients de l'objectif global du système multi-agent. Nous partons du principe que la connaissance des agents est exacte, que les communications ne peuvent pas être perturbées et que les agents sont dans un réseau entièrement connecté.

4 Modèle d'agent

La figure 1 montre un aperçu du modèle d'agent ABSG. ABSG est composé de trois principaux modules : (1) les mémoires, (2) le module social et (3) le module décisionnel. Les sections 4.1, 4.2 et 4.3 détaillent ces modules. Les agents ABSG fonctionnent à plusieurs, ils ont besoin d'un protocole d'interaction pour se comprendre. Les actions représentent les messages qu'un agent peut envoyer. Les perceptions sont les messages reçus des autres agents. La partie 4.4 explique comment les agents interagissent.

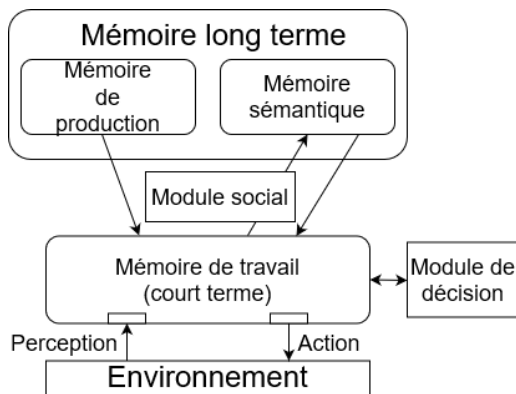


FIGURE 1 – Modèle ABSG

4.1 Mémoires

Comme décrit dans la partie précédente, nous sommes dans un contexte très variable où les agents peuvent être modifiés à tout moment. Notre modèle doit prendre en compte les modifications concernant l'agent lui-même et les autres agents de son environnement. Ainsi, les agents doivent être suffisamment adaptables pour former des coalitions avec de nouveaux entrants ou quitter une coalition en fonction des caractéristiques des autres membres. Pour répondre à ces besoins, notre modèle doit être capable de construire une représentation des agents et de leur environnement. Il doit stocker des informations sur ses propres caractéristiques et désirs et sur ses accointances (*e.g.* comment les contacter, quelles sont leurs caractéristiques, *etc.*). Ces informations peuvent être stockées dans trois types de mémoires : (1) une mémoire de travail : c'est une mémoire à court terme stockant les variables temporaires utilisées par le module de prise de décision, (2) une mémoire sémantique : c'est une mémoire à long terme stockant des faits sur l'environnement de l'agent, (3) une mémoire de production : c'est une mémoire à long terme stockant des règles de prise de décision.

4.2 Module social

En plus de la représentation des connaissances, ABSG doit prendre des décisions en fonction de son environnement. Nous avons donc intégré une heuristique de sélection d'action et un module d'évaluation de la qualité dans notre modèle d'agent. Comme les humains sont bien adaptés pour former des groupes et y travailler efficacement, nous nous inspirons du champ de la *dynamique de groupe* [5] en SHS. *La dynamique de groupe* est un domaine qui décrit ce que sont les groupes restreints, comment ils sont formés et comment ils sont maintenus. En utilisant cette approche, nous introduisons une nouvelle métrique dans notre modèle d'agents permettant d'estimer pas à pas la qualité des autres agents. Ce module produit une valeur d'attraction représentant la mesure dans laquelle un agent répond aux désirs d'un autre agent. La valeur d'attraction est traitée en s'inspirant de l'intelligence distribuée [18][3]. Elle est initialisée de manière aléatoire, puis mise à jour dynamiquement et régulièrement pendant le temps d'exécution de l'agent en fonction de son environnement. L'équation traitant l'attraction la fait converger si les croyances des agents ne sont pas modifiées (*cf.* preuve [3]). Cette façon de traiter la valeur

Contribution scientifique

d'attraction présente deux avantages : premièrement, elle permet aux agents d'explorer des solutions avant de converger vers un état final, deuxièmement, elle permet à un observateur externe de suivre la progression du système dans la recherche de solutions.

L'attraction est une distance entre deux points virtuels dans un espace euclidien à une dimension. Le premier représente l'agent qui traite la valeur, le second représente l'agent pour lequel on veut connaître la valeur d'attraction. L'équation suivante permet de mettre à jour l'attraction entre les deux agents en calculant le mouvement du second point dans l'espace :

$$\dot{x}^i = f(|x^i - x^j|) \quad (1)$$

$$f(y) = -y \left(\frac{a}{\|y\|} - \frac{b}{\|y\|^4} \right) \quad (2)$$

où x^i et x^j sont la position des individus i et j dans l'espace euclidien et $i \neq j$. L'agent i , qui traite la valeur, ne se déplace pas et se trouve toujours au point 0. $f(y)$ étant la fonction d'attraction/répulsion où a et b sont deux constantes et $\|y\|$ la norme euclidienne donnée par $\|y\| = \sqrt{y^T y}$ qui est la distance entre deux agents dans l'espace euclidien. Nous avons également créé deux fonctions paramétriques pour choisir les bonnes constantes a et b .

$$f_a(z) = \frac{0.05^z}{2} \quad f_b(z) = 20^{(3+z)} * z^3 \quad (3)$$

z étant un biais qui intègre les facteurs sociaux décrit dans les parties suivantes. f_a et f_b ont été établis empiriquement. Ils permettent de conserver un bon rapport entre a et b et gèrent bien la fonction d'attraction/répulsion quelle que soit la valeur de z . Cependant, ces fonctions n'ont pas été conçues pour être optimales.

Dans la *dynamique de groupe*, le sous-domaine *formation de groupe* se concentre sur les processus qui génèrent des liens d'attraction entre les membres des groupes. Le processus de formation de groupe est un phénomène complexe qui implique de nombreuses dimensions. Parmi ces dimensions, les principes d'attraction prennent une grande part [14, 5, 7] :

- principe de similarité (s) : les individus aiment les personnes qui leur ressemblent [19]. Dans notre système, la similarité est une distance entre les désirs de deux agents.

- principe de complémentarité (c) : les individus aiment les personnes qui leur ressemblent et dont les qualités sont complémentaires aux leurs. Nous le représentons par la complémentarité des caractéristiques des agents.
- principe d'attractivité physique (a) : les individus sont plus attirés par les personnes qui sont physiquement attractives. Dans notre système, l'attractivité physique est considérée comme l'adéquation entre les caractéristiques d'un individu et les désirs de l'autre.
- principe du minimax (m) : les individus sont attirés par les personnes qui leur offrent une récompense maximale et un coût minimal.

Les agents doivent être capables d'évaluer l'attrait qu'ils ont pour d'autres agents afin de former des groupes dans le système. Pour ce faire, nous construisons l'équation 4 qui intègre ces principes.

$$z = \frac{\frac{4}{3} * s * a}{c} + m \quad (4)$$

4.3 Module de décision

Prise de décision. La prise de décision du modèle ABSG est basée sur deux facteurs : (1) les règles de comportement issues de la mémoire de production, (2) la valeur d'attraction traitée pour les accointances ou les groupes. Les comportements sont un ensemble de règles qui indiquent comment un agent doit agir dans des situations spécifiques. Ils sont similaires à une déclaration "si/alors".

Règle 1. Si l'agent ne connaît pas les caractéristiques d'une de ses accointances, alors il les lui demande.

Règle 2. Si l'agent reçoit une demande de caractéristiques, alors il les envoie à l'accointance qui les demande.

Règle 3. Si l'agent a une valeur d'attraction inférieure à un seuil α pour une de ses accointances, alors il lui envoie une requête de formation de groupe.

Règle 4. Si l'agent reçoit une demande de formation de groupe et que la valeur de l'attraction pour l'émetteur de la demande est inférieure à un seuil α , alors il accepte la requête, sinon il la refuse.

Contribution scientifique

Règle 5. Si l'accointance possédant la plus faible valeur d'attraction n'est pas dans le groupe qui possède la plus faible valeur d'attraction de l'agent, alors il lui envoie une requête à son accointance pour lui demander de rejoindre son groupe.

Règle 6. Si l'agent reçoit une demande pour rejoindre un groupe et que son attraction pour l'accointance émettrice est inférieure à un seuil β , alors il accepte, sinon il refuse.

Règle 7. Si l'attraction de l'agent pour un groupe est supérieure à un seuil γ , alors il le quitte en notifiant tous les autres membres de son départ.

Règle 8. Si un agent change de caractéristiques, alors il renvoie les nouvelles à toutes ses accointances.

Ces règles font partie du modèle ABSG et sont utilisées par les agents pour savoir quelle action exécuter en fonction de leur état actuel. Mais un agent doit choisir la bonne règle à appliquer. Pour cela, ils suivent un cycle de décision qui leur permet de sélectionner leur prochaine action.

Cycles de décision. Le cycle de décision est divisé en quatre phases :

- La phase d'élaboration : ABSG charge dans la mémoire de travail toutes les règles de production qui peuvent être déclenchées.
- La phase de décision : ABSG choisit une règle à lancer. Si plusieurs règles peuvent être lancées, il choisit la règle de plus haute priorité qui est l'ordre dans lequel elles sont présentées dans la partie précédente.
- La phase d'application : l'action de la règle sélectionnée est exécutée. La mémoire de travail et la mémoire sémantique sont modifiées.
- La phase de socialisation : les valeurs d'attractions des connaissances et des coalitions sont mises à jour en fonction du nouvel état de la mémoire sémantique.

4.4 Interactions

Les interactions permettent aux agents de s'auto-organiser en coalitions. Les messages transmettent des informations telles que les caractéristiques des agents ou des requêtes afin de former, rejoindre ou quitter un groupe. Nous utilisons le langage de communication des agents (FIPA-ACL) pour faire communiquer les agents.

La figure 2 illustre une instance du protocole interne d'un agent dans un scénario de formation de groupe. L'état 1 est le début, l'état 4 la fin, ! sont les messages émis, ? les messages attendus et x l'identifiant de la nouvelle coalition.

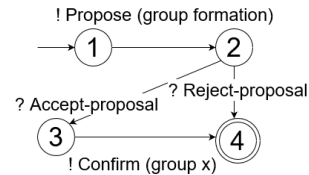


FIGURE 2 – Protocole de proposition de formation de groupe

5 Operationalisation

5.1 Facteurs sociaux

Nous caractérisons un agent par deux listes de décimaux. Elles sont la représentation des caractéristiques et des désirs des agents sous forme numérique.

- caractéristiques personnelles (C) : ce sont les caractéristiques physiques des agents (par exemple, poids, forme, dimensions, prix, etc.),
- désirs (D) : sont les caractéristiques des accointances avec lesquelles les agents aimeraient former un groupe (*e.g.* dimensions désirées, prix désiré, *etc.*);

Dans cette expérience, nous utilisons une liste de trois décimaux compris entre 0 et 1 pour représenter les informations de chaque module de nos agents. La taille des matrices n'a aucune importance pour le bon fonctionnement du système. Afin d'utiliser ces informations, nous devons intégrer les principes de l'attraction de la partie 4.2 à notre expérimentation.

Similarité (s), Attrait physique. Le principe de similarité est une distance entre les désirs de deux agents. Le principe de l'attractivité physique est une distance entre les désirs d'un agent et les caractéristiques d'un autre agent. Soit $[M]$ le nombre d'éléments d'une matrice M à N dimensions.

$$s = \frac{\sqrt{u^T u}}{|D|} \quad u = x_D^i - x_D^j \quad (5)$$

où x_D^i et x_D^j sont la matrice des désirs des agents i et j . Le principe de l'attractivité physique est traité avec la même équation en remplaçant les matrices utilisées par la matrice de désir d'un agent et les caractéristiques de l'autre.

Contribution scientifique

Complémentarité (c) est une distance entre les caractéristiques de deux agents :

$$c = 1 - \frac{\sqrt{u^T u}}{|C|} \quad u = x_C^i - x_C^j \quad (6)$$

où x_s^i et x_s^j sont les matrices des caractéristiques des agents i et j .

principe du minimax augmente l'attraction lorsqu'un individu rejette une proposition d'un autre. Il permet aux individus de cesser d'envoyer une demande lorsqu'elle a été rejetée plusieurs fois.

$$m = nbR * \epsilon \quad (7)$$

où nbR est le nombre de demandes rejetées par un agent et ϵ un paramètre à ajuster.

6 Expériences et analyses

6.1 Méthodologie

Dans ces expériences, pour simuler une demande de produit d'un utilisateur, le but des agents est de former des coalitions contenant un nombre spécifique de membres tout en trouvant les meilleurs agents en fonction de leurs désirs et de leurs caractéristiques. Nous avons défini la taille idéale des coalitions à 4 pour les rendre réalisables dans les expériences à plus petite échelle. La valeur d'une coalition $V(C)$ est une valeur comprise entre 0 et 1. Elle est utilisée comme mesure de la qualité pour un utilisateur extérieur. 0 représente la qualité la plus faible qu'une coalition peut avoir, 1 représente la qualité la plus élevée.

$$V(C) = \text{moyenne}(S_c, R_c) \quad (8)$$

où R_c est la distance moyenne entre les désirs de chaque agent d'une coalition et les caractéristiques de tous les autres membres. S_c est un ratio associé à la taille de la coalition défini comme

$$S_c = \frac{(|cs - O_s|)}{O_s} \quad (9)$$

où cs est la taille d'une coalition et O_s la taille optimale d'une coalition. O_s est fixé à 4 pour ces expériences.

La valeur d'attraction $A(C)$ représente l'attraction qu'un agent exerce sur son groupe. Elle est traitée avec une équation proche de $V(C)$. $A(C)$ est défini de telle sorte que

$$A(C) = \gamma_S S_c + \gamma_A A_c \quad (10)$$

où A_c est l'attraction moyenne que les agents ont pour les autres membres de la coalition.

Les seuils utilisés dans la section 4.3 sont choisis empiriquement : $\alpha : 5$, $\beta : 20$, $\gamma : 25$. Plus les α et les β sont petits, plus la qualité des coalitions est bonne et moins il y a de coalitions.

Dans les sections suivantes, nous évaluons cinq critères : (1) la durée des expériences, (2) la qualité des groupes formés, (3) la robustesse du système à la charge de communication, (4) la robustesse à l'ouverture, (5) la robustesse à la variabilité. Les caractéristiques et les désirs des agents sont générés de manière aléatoire pour les sections 6.2, 6.4 et 6.5 et de manière pseudo-aléatoire pour la section 6.3. Puisque nous avons besoin de connaître la meilleure solution pour évaluer la qualité de notre résultat, un script Python génère des agents afin que nous connaissions la valeur de la meilleure coalition possible. Les expériences sont effectuées cinq fois et moyennées. Elles sont exécutées sur un PC multi-cœur Intel i7-6600U CPU (2.60GHz), 16 Go, 64-bit, Manjaro Linux 20.1.

6.2 Durée des expériences

La figure 3 compare la durée d'exécution des algorithmes ODP et ODP-IP de [15] et celle du modèle ABSG. ODP-IP et ODP sont deux algorithmes exacts qui trouvent les meilleures coalitions possibles dans un système. Nous les exécutons sur la même machine qu'ABSG afin de comparer le temps d'exécution. ODP et ODP-IP sont plus rapides sur les premières expériences, le temps d'exécution augmente de façon exponentielle avec le nombre d'agents. Contrairement à ces deux algorithmes, le temps d'exécution d'ABSG est polynomial. ABSG devient plus rapide qu'ODP-IP à partir de 25 agents et plus rapide qu'ODP à partir de 23 agents. ABSG semble donc plus adapté pour traiter un nombre élevé d'agents comme notre système d'aide à la décision.

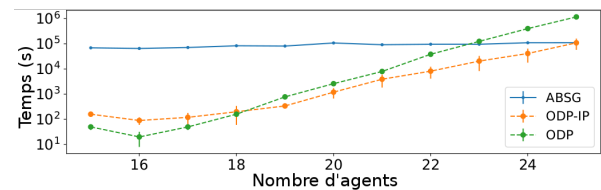


FIGURE 3 – Durée des expériences en fonction du nombre d'agents

Contribution scientifique

6.3 Qualité des groupes

Nous avons généré les caractéristiques des agents pseudo-aléatoirement afin de connaître la valeur de la meilleure coalition que nos agents peuvent construire pendant l'exécution. La figure 4 compare nos résultats à la méthode SACF et à la méthode de Janovsky et DeLoach. Le modèle d'agents produit de bonnes coalitions avec une qualité moyenne de 97,5% et un écart-type moyen de 1,1%. La méthode SACF et le modèle ABSG sont tous deux conçus pour traiter un problème de CSG dans un environnement ouvert. Dans [17], la valeur de la coalition est corrélée à la somme des valeurs des membres de la coalition tandis que dans [9], elle est définie comme la moyenne des gains que chaque membre des coalitions obtient de par leur participation à celles-ci. Mais comme ces méthodes expriment leur résultat en pourcentage de la solution optimale qu'elles peuvent trouver, la principale différence avec nos résultats est probablement due au fait que notre modèle d'agent est conçu pour un système d'aide à la décision et que nous ne considérons que la meilleure coalition comme une solution. Contrairement à ABSG, les solutions de la méthode SACF et de la méthode de Janovsky et DeLoach sont une structure de plusieurs coalitions.

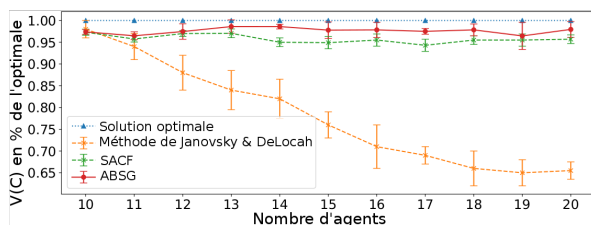


FIGURE 4 – Comparaison de la qualité des solutions pour l'ABSG, la méthode SACF et de la méthode de Jenovsky et DeLoach

6.4 Robustness to the communication load

La figure 5 montre que le nombre de messages suit une augmentation polynomiale. Cela peut s'expliquer par le fait que les agents ne se connaissent pas au début de l'expérience. Ils doivent donc émettre des messages pour connaître les caractéristiques et les désirs des autres agents afin d'estimer leur attirance pour eux et de prendre des décisions. Au contraire, dans [17], les agents ne s'échangent pas d'information sur eux. Les messages leur servent uniquement à créer, rejoindre ou quitter des coalitions ce qui explique que leur méthode soit moins émettrice de communications que ABSG.

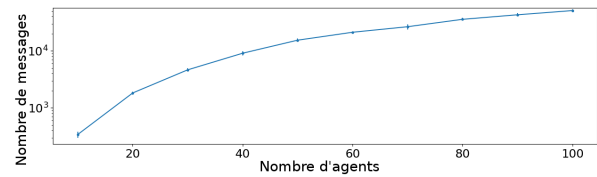


FIGURE 5 – Nombre de messages émis en fonction du nombre d'agents dans le système

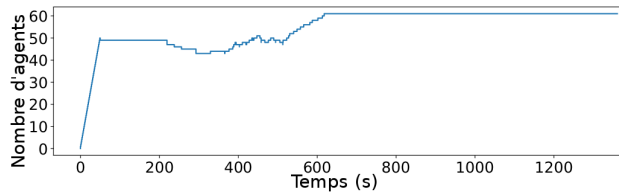
6.5 Robustness to the openness

Nous voulons voir comment le système réagit lorsque nous ajoutons ou retirons des agents. Notez que, contrairement à la section 6.3, nous ne connaissons pas la valeur optimale qu'une coalition peut avoir. La figure 6a montre le nombre d'agents dans le système. Deux phases se chevauchent. Pendant la première (200s-500s), nous retirons aléatoirement des agents du système puis, pendant la seconde (400s-650s), nous en ajoutons de nouveaux. Tout en retirant des agents du système au cours de la première phase, le nombre de messages émis reste faible (figure 6b). D'abord parce que le système converge naturellement vers des solutions et que les agents réduisent leurs interactions. Mais nous retirons également les agents du système, laissant les agents restants avec moins d'accointances avec lesquelles ils peuvent interagir. Au contraire, la suppression d'agents ne semble pas avoir d'impact sur la qualité moyenne des coalitions (figure 6c). Cela peut être dû au fait que tous les agents éliminés ne faisaient pas partie d'une coalition à ce moment-là. Dans la deuxième phase, on peut voir que le lancement de nouveaux agents augmente le nombre de messages émis. Lorsqu'ils entrent dans le système, ils sont inconnus et les agents doivent envoyer leurs caractéristiques et leurs désirs à d'autres. La première fois, l'introduction de nouveaux agents diminue la qualité moyenne des coalitions. En effet, la valeur d'attraction des anciens agents pour les nouveaux est initialisée de manière aléatoire. Ensuite, lorsque le nombre d'agents dans le système est stable, de nouveaux agents entrent dans les coalitions et les qualités des coalitions augmentent.

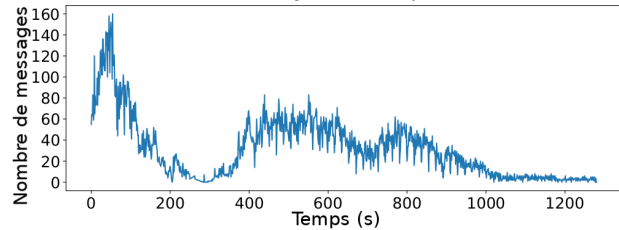
6.6 Robustesse à la variabilité

Dans cette expérience de 50 agents, nous voulons voir comment le système réagit si les agents changent aléatoirement leurs caractéristiques et leurs désirs. Dans cette expérience, nous ne connaissons pas la valeur optimale qu'une coalition peut avoir. À environ 200s, tous les agents modifient aléatoirement les valeurs de

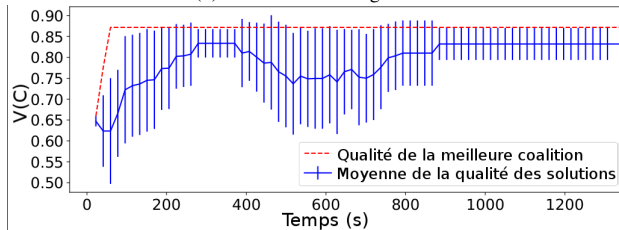
Contribution scientifique



(a) Nombre d'agents dans le système



(b) Nombre de messages émis



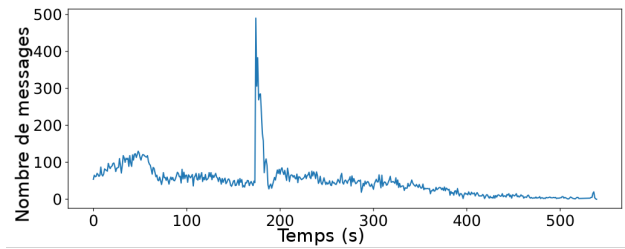
(c) Qualité des coalitions

FIGURE 6 – Openness experimentation

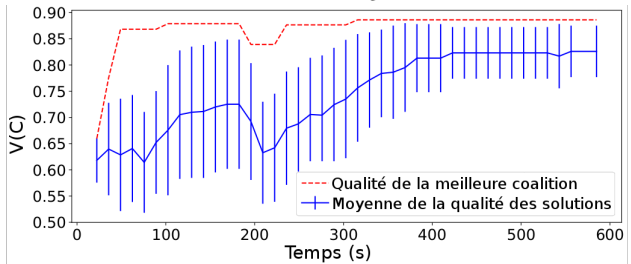
leurs caractéristiques et de leurs désirs, ce qui entraîne l'explosion des messages émis (figure 7a). Conformément à la règle 8, lorsque les agents reçoivent de nouvelles caractéristiques, ils doivent les envoyer à leurs accointances. La qualité des coalitions sont largement affectées par ces modifications (figure 7b). Lorsqu'ils reçoivent les nouvelles caractéristiques de leurs accointances, les agents mettent à jour toutes leurs valeurs d'attraction et se réorganisent, ce qui augmente la moyenne des qualités de coalition. Seules les coalitions détériorées par les changement de caractéristiques et de désirs des agents se réorganisent. Si une coalition reste de bonne qualité après la modification des agents, celle-ci n'est pas modifiée. Le système recherche des solutions de manière incrémentale et peut s'adapter à des modifications de ses agents sans avoir à reconstruire les coalitions en partant de zéro. En outre, on remarquera que le système est anytime, il est capable de délivrer une solution à tout moment.

7 Conclusion

Nous avons présenté le modèle d'agent ABSG conçu pour résoudre le problème de génération de structures de coalitions. Ce modèle est basé



(a) Nombre de messages émis



(b) Qualité des coalitions

FIGURE 7 – Expérimentation de la variabilité

sur les principes d'attraction des SHS et permet aux agents de former des coalitions. Les coalitions sont construites sur la base d'une valeur d'attraction que chaque agent du système estime pour ses accointances. Cette valeur est prise en compte par les agents pour former des coalitions, les rejoindre ou les quitter. Les résultats expérimentaux montrent qu'ABSG est plus adapté pour une application à grande échelle que les algorithmes d'optimisation. Nous montrons également que les coalitions formées par les agents d'ABSG sont proches de l'optimum lors d'expériences à petite échelle. En outre, ABSG produit de meilleures coalitions que certaines autres approches similaires de la littérature. Nous montrons également qu'ABSG est adapté à un système ouvert et variable où les agents peuvent être modifiés à tout moment dans des expériences à grande échelle. Les travaux futurs consisteront à intégrer pleinement le modèle d'agent dans un système d'aide à la décision et à le tester sur un cas d'étude réelle.

Références

- [1] Samir Aknine, Suzanne Pinson, and Melvin F Shakun. A multi-agent coalition formation method based on preference models. *Group Decision and Negotiation*, 13(6) :513–538, 2004.
- [2] Elliot Anshelevich and Wennan Zhu. Ordinal approximation for social choice, matching, and facility location problems gi-

Contribution scientifique

- ven candidate positions. *ACM Transactions on Economics and Computation (TEAC)*, 9(2) :1–24, 2021.
- [3] Mickael Bettinelli, Ocelllo Michel, and Genthial Damien. Coalition formation problem : a group dynamics inspired swarming method. *HAL preprint hal-02903531*, 2020.
- [4] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.
- [5] Donelson R Forsyth. *Group dynamics*. Cengage Learning, 2010.
- [6] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1) :9–15, 1962.
- [7] David Dryden Henningsen, Mary Lynn Miller Henningsen, and Paul Booth. Predicting social and personal attraction in task groups. *Groupwork*, 23(1) :73–93, 2013.
- [8] Fabrício R Inácio, Douglas G Macharet, and Luiz Chaimowicz. Pso-based strategy for the segregation of heterogeneous robotic swarms. *Journal of Computational Science*, 31 :86–94, 2019.
- [9] Pavel Janovsky and Scott A DeLoach. Multi-agent simulation framework for large-scale coalition formation. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 343–350. IEEE, 2016.
- [10] M. Kumar, D. P. Garg, and V. Kumar. Segregation of heterogeneous units in a swarm of robotic agents. *IEEE Transactions on Automatic Control*, 55(3) :743–748, 2010.
- [11] David F Manlove. Hospitals/residents problem. 2008.
- [12] Tomasz Michalak, Talal Rahwan, Edith Elkind, Michael Wooldridge, and Nicholas R Jennings. A hybrid exact algorithm for complete set partitioning. *Artificial Intelligence*, 230 :14–50, 2016.
- [13] Michael Mrissa, Lionel Médini, Jean-Paul Jamont, Nicolas Le Sommer, and Jérôme Laplace. An avatar architecture for the web of things. *IEEE Internet Computing*, 19(2) :30–38, 2015.
- [14] Theodore M Newcomb. Some varieties of interpersonal attraction. 1960.
- [15] Talal Rahwan, Tomasz P Michalak, Edith Elkind, Michael Wooldridge, and Nicholas R Jennings. An exact algorithm for coalition structure generation and complete set partitioning. *DCS*, 2013.
- [16] Talal Rahwan, Tomasz P Michalak, Michael Wooldridge, and Nicholas R Jennings. Coalition structure generation : A survey. *Artificial Intelligence*, 229 :139–174, 2015.
- [17] Gabriel De O. Ramos, Burguillo Juan C., and Ana L.C. Bazzan. Self-adapting coalition formation among electric vehicles in smart grids. In *2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems*, pages 11–20, 2013.
- [18] Hong Shi and Guangming Xie. Collective dynamics of swarms with a new attraction/repulsion function. *Mathematical Problems in Engineering*, 2011, 2011.
- [19] Elaine Walster, Vera Aronson, Darcy Abrahams, and Leon Rottman. Importance of physical attractiveness in dating behavior. *Journal of personality and social psychology*, 4(5) :508, 1966.
- [20] D Yun Yeh. A dynamic programming approach to the complete set partitioning problem. *BIT Numerical Mathematics*, 26(4) :467–474, 1986.

Planification multi-utilisateurs et multi-satellites de tâches d'observation dans des constellations avec portions d'orbites exclusives

Gauthier Picard
gauthier.picard@onera.fr

ONERA-DTIS, Toulouse, France

Résumé

Nous étudions des techniques distribuées de planification sur des scénarios d'observation de la Terre avec utilisateurs et satellites multiples. Nous nous concentrons sur la coordination des utilisateurs ayant réservé des portions d'orbites exclusives et d'un planificateur central ayant plusieurs demandes qui peuvent utiliser certains intervalles de ces portions exclusives. Nous définissons le problème de planification de constellations de satellites d'observation de la Terre (EOSCSP). Pour le résoudre, nous proposons des schémas de résolution distribués, notamment l'optimisation sous contraintes distribuée, où les agents coopèrent pour répartir les demandes sans partager leurs propres plans. Ces contributions sont évaluées expérimentalement sur des instances EOSCSP générées sur la base de carnets d'observation réels grande échelle ou très conflictuels.

Mots-clés : Constellation de satellites, planification, DCOP, allocation de ressources

Abstract

We investigate the use of distributed scheduling techniques on problems related to Earth observation scenarios with multiple users and satellites. We focus on the problem of coordinating users having reserved exclusive orbit portions and one central planner having several requests that may use some intervals of these exclusives. We define this problem as Earth Observation Satellite Constellation Scheduling Problem (EOSCSP) and map it to a Mixed Integer Linear Program. As to solve EOSCSP, we propose distributed solving schemes, notably Distributed Constraint Optimization, where agents cooperate to allocate requests without sharing their own schedules. These contributions are experimentally evaluated on EOSCSP instances based on real large-scale or very conflicting observation order books.

1. Traduction de l'article en cours d'évaluation à IJCAI 2021 [14]

Keywords: Satellite constellation, scheduling, DCOP, resource allocation

1 Introduction

Ces dernières années ont vu une forte augmentation du développement des constellations de satellites. Au lieu de considérer des satellites individuels, elles tirent parti d'un groupe de satellites, dont certains partagent souvent les mêmes plans orbitaux, pour fournir des services plus riches comme le positionnement, les télécommunications ou l'observation de la Terre [21]. Avec peu de satellites dans une constellation (e.g. deux dans le projet PLEIADES [8]), et en orbite terrestre basse ou moyenne (altitude inférieure à 35 000 km), toute région de la Terre n'est pas couverte par la constellation à tout moment. Ainsi, la principale motivation pour augmenter la taille de ces constellations est de permettre de capturer avec une grande réactivité n'importe quel point sur Terre, comme le fait la société Planet avec plus de 150 satellites d'observation de la Terre (EOS) [16]. Mais l'exploitation de nombreux EOS nécessite une meilleure coopération entre les ressources et une autonomie à bord afin d'utiliser au mieux le système, ce qui devient une tâche hautement combinatoire. Outre leur nombre croissant, la composition des constellations évolue également. Les récentes avancées technologiques permettent la production et le déploiement d'EOS agiles capables de changer leur orientation, et de fournir de multiples types de prises de vue avec de multiples capteurs. Tout en offrant des services plus riches à de multiples utilisateurs, cela ajoute de nombreux degrés de liberté et des variables de décision pour programmer l'activité des EOS, et ouvre de nombreux défis [22].

Parmi ces défis, nous nous concentrons sur la planification collective d'observations sur un ensemble de satellites pour lesquels certains utilisateurs ont un accès exclusif à certaines por-

tions d'orbite, en utilisant des techniques distribuées, de manière à répartir les décisions entre les différents utilisateurs de la constellation. La spécificité découlant de la gestion des exclusions et des exigences de confidentialité des tâches programmées dans les fenêtres exclusives entraîne la nécessité de recourir à des méthodes de résolution distribuées. Si la littérature sur la planification multi-satellite est riche, comme le confirme un récent article de synthèse [22] et des travaux récents [1, 12, 19, 17, 2, 3, 6, 18, 23, 16, 11, 22, 7, 20], considérer les constellations de satellites comme des ressources partagées nécessitant la coordination d'utilisateurs multiples pour la répartition des tâches dans des portions d'orbite exclusives est un problème totalement nouveau, que nous abordons dans cet article.

La section 2 illustre et définit le problème de planification de constellation de satellites d'observation de la Terre (EOSCSP). La section 3 se concentre sur les méthodes de résolution centralisées : un programme linéaire et une approche gloutonne pour EOSCSP. La section 4 expose certaines approches distribuées pour résoudre EOSCSP, en utilisant différents schémas de communication entre les utilisateurs, et la section 5 ajoute la coordination entre les utilisateurs exclusifs en utilisant des techniques d'optimisation distribuée sous contraintes (DCOP). Nous évaluons expérimentalement ces différents algorithmes en utilisant des instances générées de façon aléatoire dans la section 6. Enfin, la section 7 conclut l'article avec quelques perspectives.

2 Le modèle EOSCSP

Cette section illustre le problème que nous étudions à l'aide d'un exemple de scénario, et fournit ensuite quelques définitions de base.

2.1 Scénario illustratif

La figure 1 illustre un scénario, où nous considérons : 3 satellites, chacun ayant une période de planification donnée (par exemple, planification sur la prochaine orbite, ou sur les horizons en fonction des fenêtres de communication entre le satellite et les stations au sol); 1 utilisateur u_0 sans portion d'orbite exclusive; 2 utilisateurs ayant des portions d'orbite exclusives telles que u_1 possède des exclusivités sur le satellite s_0 et sur le satellite s_1 (rouge hachuré), u_2 possède des exclusivités sur le satellite s_0 et sur le satellite s_2 (bleu hachuré); plusieurs requêtes à effectuer avant une date d'échéance, noté $r_{i,j}$ pour la

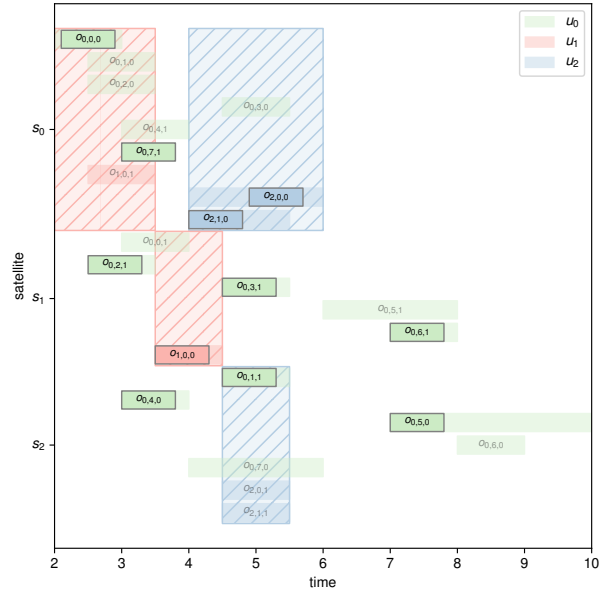


FIGURE 1 – Un exemple avec 3 satellites, 2 utilisateurs (rouge et bleu) avec des exclusives (zones hachées), et 1 utilisateur non exclusif (vert). Les fenêtres de temps d'observation apparaissent comme des surfaces transparentes. Une solution optimale est représentée par des surfaces opaques.

j ième requête pour l'utilisateur i ; plusieurs possibilités d'observation par requête, notées $o_{i,j,k}$ pour la k ième observation de la j ième requête du i ième utilisateur. Une seule observation doit être planifiée pour répondre à la requête sur des créneaux temporels en fonction des orbites des satellites et de la position des zones d'intérêt (les créneaux sont représentés sous forme de zones transparentes). Plus précisément, nous considérons 2 observations par requête, de sorte que les observations $o_{1,0,0}$ et $o_{1,0,1}$ sont personnelles à u_1 (en rouge), les observations $o_{2,0,0}$, $o_{2,0,1}$, $o_{2,1,0}$ et $o_{2,1,1}$ sont personnelles à u_2 (en bleu), les observations $o_{0,j,k}$ (en vert) sont directement demandées au planificateur central u_0 par d'autres clients sans fenêtre exclusive. La solution de la figure 1, représentée par des observations surlignées, répond à toutes les requêtes, en permettant à l'utilisateur non exclusif u_0 de positionner des observations sur des portions d'orbite exclusives (par exemple $o_{0,0,0}$ sur le satellite s_0). Une contrainte énergétique simplifiée stipule qu'un satellite ne peut pas effectuer plus de t_{\max} minutes d'observation sur sa période de programmation (ici, un maximum de 4 observations est autorisé par satellite), des temps de transition minimaux entre deux observations o et p , en fonction de o et p et de la date à laquelle la transition est déclenchée sur un satellite donné.

Au niveau global, chaque utilisateur exclusif (u_1

ou u_2) peut avoir son propre système de planification pour gérer ses périodes exclusives, et un système de planification central (u_0) gère les observations $o_{0,j,k}$. En fin de compte, chaque utilisateur et le planificateur central ont un problème de planification local à résoudre. Résoudre ces problèmes séparément peut conduire le planificateur central à ne pas pouvoir réserver de créneaux sur des portions d'orbite exclusives, alors que cela pourrait améliorer la solution. Sans coordination, et avec une gestion non coopérative des créneaux exclusifs, la planification globale pourrait ne pas être optimale, du point de vue du nombre d'observations programmées possibles. Nous proposons donc ici de coordonner les processus de planification entre les utilisateurs.

2.2 Définitions et notations

Présentons maintenant les concepts fondamentaux de ce problème de planification.

Définition 1. Un *problème de planification de la constellation de satellites d'observation de la Terre avec des exclusivités* (ou EOSCSP) est défini par un tuple $P = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$, tel que \mathcal{S} est un ensemble de satellites, \mathcal{U} est un ensemble d'utilisateurs, \mathcal{R} est un ensemble de requêtes, et \mathcal{O} est un ensemble d'observations à programmer pour répondre aux requêtes de \mathcal{R} .

Définition 2. Un *satellite* est défini comme un tuple $s = \langle t_s^{\text{start}}, t_s^{\text{end}}, \kappa_s, \tau_s \rangle$ avec $t_s^{\text{start}} \in \mathbb{R}$ l'heure de début de son plan d'orbite, $t_s^{\text{end}} \in \mathbb{R}$ l'heure de fin de son plan d'orbite, $\kappa_s \in \mathbb{N}^+$ sa capacité (i. e. le nombre maximum d'observations pendant son plan d'orbite), $\tau_s : \mathcal{O} \times \mathcal{O} \rightarrow \mathbb{R}$ la fonction définissant le temps de transition entre deux observations données.

Définition 3. Un *utilisateur* est défini comme un tuple $u = \langle e_u, p_u \rangle$ avec un ensemble (éventuellement vide) de fenêtres temporelles exclusives $e_u = \{ (s, (t^{\text{start}}, t^{\text{end}})) \mid s \in \mathcal{S}, [t^{\text{start}}, t^{\text{end}}] \subseteq [t_s^{\text{start}}, t_s^{\text{end}}] \} \subset (\mathcal{S} \times (\mathbb{R} \times \mathbb{R}))$, et une priorité $p_u \in \mathbb{N}^+$ (utilisée en cas de conflit). On note \mathcal{U}^{ex} (resp. \mathcal{U}^{nex}) l'ensemble des utilisateurs possédant (resp. ne possédant pas) des exclusivités.

Nous supposons ici qu'un seul utilisateur n'a pas de portion d'orbite exclusive : le planificateur central, noté u_0 , c'est-à-dire $\mathcal{U}^{\text{nex}} = \{u_0\}$.

Définition 4. Une *requête* est définie comme un tuple $r = \langle t_r^{\text{start}}, t_r^{\text{end}}, \Delta_r, \rho_r, p_r, u_r, \theta_r \rangle$, avec une fenêtre temporelle de validité définie par $t_r^{\text{start}} \in \mathbb{R}$ et $t_r^{\text{end}} \in \mathbb{R}$, une durée $\Delta_r \in \mathbb{R}$, une récompense $\rho_r \in \mathbb{R}$ si r est réalisée, une position

GPS pour observer p_r , un émetteur $u_r \in \mathcal{U}$ et une liste $\theta_r \in 2^{\mathcal{O}}$ d'opportunités d'observation pour valider la requête.

θ_r est calculée dynamiquement sur la configuration actuelle de la constellation et la position GPS demandée p_r , puisque plusieurs satellites agiles, en changeant leur orientation peuvent acquérir la même position, générant ainsi plusieurs opportunités d'observation.

Définition 5. Une *observation* est définie comme un tuple $o = \langle t_o^{\text{start}}, t_o^{\text{end}}, \Delta_o, r_o, \rho_o, s_o, u_o, p_o \rangle$, avec une fenêtre temporelle de validité définie par $t_o^{\text{start}} \in \mathbb{R}$ et $t_o^{\text{end}} \in \mathbb{R}$, une requête r_o à laquelle elle contribue, une durée $\Delta_o \in \mathbb{R}$, une récompense $\rho_o \in \mathbb{R}$ (héritée de r_o), un satellite s_o sur lequel cette observation peut être planifiée, un émetteur $u_o \in \mathcal{U}$ (héritée de r_o), et une priorité $p_o \in \mathbb{N}^+$ (héritée de r_o).

Définition 6. Une *solution* à un EOSCSP est une allocation $\mathcal{M} = \{ (o, t) \mid o \in \mathcal{O}, t \in [t_o^{\text{start}}, t_o^{\text{end}}] \}$ associant une heure de début à au plus une observation par requête de sorte que les utilisateurs exclusifs aient leurs observations planifiées sur leurs fenêtres exclusives respectives, et que la récompense globale soit maximisée (somme des récompenses des observations planifiées) : $\arg \max_{\mathcal{M}} \sum_{(o,t) \in \mathcal{M}} \rho_o$.

Définition 7. Un *EOSCSP pour l'utilisateur* u , noté $P[u] = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}[u], \mathcal{O}[u] \rangle$ (ou EOSCSP $[u]$), est un EOSCSP, sous-problème d'un autre EOSCSP $P = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$ limité aux requêtes et observations appartenant à l'utilisateur u , où $\mathcal{R}[u] = \{ r \mid r \in \mathcal{R}, u_r = u \} \subseteq \mathcal{R}$ et $\mathcal{O}[u] = \{ o \mid o \in \mathcal{O}, u_o = u \} \subseteq \mathcal{O}$.

Plus généralement, on note $P[x]$ (resp. $\overline{P[x]}$) le problème P limité aux seules composantes liées à x , x étant une requête, une observation ou un satellite. Plus tard, nous utiliserons également les notations $P[\emptyset | \mathcal{M}]$ (resp. $P[u_1, \dots, u_m | \mathcal{M}]$) pour définir le problème (resp. sous-problème pour les utilisateurs u_1, \dots, u_m) étant donné une allocation prédéfinie \mathcal{M} de certaines observations. En outre, nous utiliserons la notation \overline{P} pour désigner le EOSCSP P , où seules les requêtes et les observations relatives qui peuvent être planifiées en dehors de toute fenêtre d'exclusivité sont prises en compte (c'est-à-dire les observations dont les fenêtres temporelles croisent des portions d'orbite non exclusives). Enfin, on note l'union de deux problèmes $P = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$ et $P' = \langle \mathcal{S}', \mathcal{U}', \mathcal{R}', \mathcal{O}' \rangle$, $P \cup P' = \langle \mathcal{S} \cup \mathcal{S}', \mathcal{U} \cup \mathcal{U}', \mathcal{R} \cup \mathcal{R}', \mathcal{O} \cup \mathcal{O}' \rangle$.

3 Approches centralisées à EOCSPP

Nous présentons ici des approches centralisées pour résoudre EOCSPP. Chaque utilisateur ou planificateur a pour objectif de planifier certaines observations sur les satellites. Ce problème de planification des observations peut être modélisé sous la forme d'un programme linéaire en nombres mixtes (MILP). Les variables de décision sont les suivantes. $x_{s,o} \in \{0, 1\}$ est la décision d'effectuer l'observation o sur le satellite s , $t_{s,o} \in \mathbb{R}$ est la date de début de l'observation o sur le satellite s , $\beta_{s,o,p} \in \{0, 1\}$ est la précedence entre deux observations sur le même satellite, qui est égale à 1 si o est avant p sur s , sinon 0.

Nous définissons donc le programme suivant :

$$\max_{x_{s,o}} \sum_{o \in \mathcal{O}, s \in \mathcal{S}} \rho_o x_{s,o} \quad (1)$$

$$\text{t.q. } \forall s \in \mathcal{S}, \forall r \in \mathcal{R}, \forall o \in \mathcal{O}, \forall p \in \mathcal{O}$$

$$2 - \beta_{s,o,p} - \beta_{s,p,o} \geq x_{s,o} \quad (2)$$

$$2 - \beta_{s,o,p} - \beta_{s,p,o} \geq x_{s,p} \quad (3)$$

$$\beta_{s,o,p} + \beta_{s,p,o} \leq 3 - x_{s,o} - x_{s,p} \quad (4)$$

$$\beta_{s,o,p} + \beta_{s,p,o} \leq 1 \quad (5)$$

$$t_{s,p} - t_{s,o} \geq \tau_s(o, p) + \Delta_o - \Delta_{s,o,p}^{\max} \beta_{s,o,p},$$

$$\Delta_{s,o,p}^{\max} > 0 \quad (6)$$

$$t_{s,o} - t_{s,p} \geq \tau_s(p, o) + \Delta_p - \Delta_{s,p,o}^{\max} \beta_{s,p,o},$$

$$\Delta_{s,p,o}^{\max} > 0 \quad (7)$$

$$\sum_{o \in \mathcal{O}} x_{s,o} \leq \kappa_s \quad (8)$$

$$\sum_{o \in \theta(r)} x_{s,o} \leq 1 \quad (9)$$

$$x_{s,o} \in \{0, 1\} \quad (10)$$

$$t_{s,o} \in [t_o^{\text{start}}, t_o^{\text{end}}] \subset \mathbb{R} \quad (11)$$

$$\beta_{s,o,p} \in \{0, 1\} \quad (12)$$

$$\text{avec } \Delta_{s,o,p}^{\max} = t_o^{\text{end}} - t_p^{\text{start}} + \Delta_o + \tau^s(o, p)$$

(2) à (7) assurent la précedence des observations et que leur distance est au moins le temps de transition nécessaire sur leur satellite. (8) fait en sorte que le nombre d'observations planifiées sur un satellite ne dépasse pas sa capacité. (9) vérifie qu'au maximum une observation par requête est planifiée. (10) à (12) sont des définitions de domaines. Ce MILP peut être résolu en utilisant des solveurs standards comme CPLEX ou Gurobi, mais ils ne s'adapteront guère à des problèmes de grande taille (par exemple, plus de

100 observations avec 3 satellites et 3 utilisateurs). Pour que les observations des utilisateurs exclusifs aient la priorité sur les observations des utilisateurs non exclusifs, leur récompense doit être fixée à une valeur élevée. Ainsi, le solveur préférera programmer des observations exclusives sur sa fenêtre temporelle plutôt que de planifier une autre observation moins prioritaire. Bien que la solution à ce problème soit optimale, elle exige que chaque utilisateur exclusif *divulgue intégralement les informations sur les requêtes* au planificateur central.

Pour résoudre de plus grands problèmes, une approche consiste à appliquer une allocation gloutonne qui planifie d'abord les observations exclusives des utilisateurs, puis les observations plus urgentes, comme décrit dans l'algorithme 1. En pratique, c'est la technique utilisée par la plupart des opérateurs de satellites/constellations et c'est un étalon classique pour les méthodes de résolution [3, 22]. Pour ce faire, les observations sont triées par ordre croissant selon les critères de priorité et d'heure de début (ligne 2). Ensuite, pour chaque observation de cette liste triée, on trouve le premier créneau libre sur son plan d'orbite de satellite (ligne 4-8). Cet algorithme n'est pas optimal, mais fournit des solutions très rapidement. Cependant, comme pour le MILP, cette solution nécessite de *partager toutes les contraintes et informations* avec un planificateur central.

Algorithme 1 : Solveur greedy

Données : Un EOCSPP $P = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$

Résultat : Une allocation \mathcal{M}

```

1  $\mathcal{M} \leftarrow \{\}$ 
2  $\mathcal{O}^{\text{sorted}} \leftarrow \text{sort}(\mathcal{O})$ 
3  $R \leftarrow \{(s, \square) \mid s \in \mathcal{S}\}$ 
4 pour chaque  $o \in \mathcal{O}^{\text{sorted}}$  faire
5    $t \leftarrow \text{first\_slot}(o, P, R)$ 
6   si  $t \neq \emptyset$  alors
7      $\mathcal{M} \leftarrow \mathcal{M} \cup \{(o, t)\}$ 
8      $\mathcal{O}^{\text{sorted}} \leftarrow \mathcal{O}^{\text{sorted}} \setminus \theta(r_o)$ 
9 retourner  $\mathcal{M}$ 

```

4 Approches distribuées à EOCSPP

Nous étudions ici les schémas de coordination pour résoudre le problème EOCSPP de manière collective et discutons de la mesure dans laquelle les informations sur les utilisateurs doivent être divulguées pour résoudre le problème EOCSPP.

4.1 Résolution simple par communication d'exclusifs à non-exclusif

Une approche simple pour envisager de résoudre EOSCSP de manière distribuée consiste à mettre en œuvre une version distribuée de l'algorithme glouton susmentionné, où chaque utilisateur exclusif planifie ses observations privées dans ses exclusions, puis le planificateur central rassemble les observations planifiées afin de positionner les observations restantes demandées par les autres utilisateurs. Cette approche est distribuée et rapide, mais les observations des utilisateurs exclusifs doivent être communiqués au planificateur central afin de construire le plan collectif final. L'algorithme 2 esquisse ce processus de résolution distribuée, appelé *ex2nex*. La fonction *solve* est un raccourci vers n'importe quel solveur EOSCSP (par exemple, basé sur le MILP ou glouton) utilisé d'abord par chaque utilisateur exclusif pour obtenir une solution locale (ligne 1), et ensuite par le planificateur central u_0 pour positionner les observations restantes avec la connaissance des solutions locales (ligne 2).

Algorithme 2 : Solveur *ex2nex*

Données : An EOSCSP $P = \langle S, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$

Résultat : An assignment \mathcal{M}

- 1 **pour chaque** $u \in \mathcal{U}^{\text{ex}}$ **faire en parallèle**
 $\mathcal{M}_u \leftarrow \text{solve}(P[u])$
 - 2 **retourner** $\text{solve}(P[u_0] \cup \bigcup_{u \in \mathcal{U}^{\text{ex}}} \mathcal{M}_u)$
-

4.2 Résolution simple par communication non-exclusif à exclusifs

Afin de limiter la divulgation des plans des utilisateurs exclusifs, une approche symétrique à *ex2nex* consiste à considérer que le planificateur central planifie autant d'observations que possible en dehors des fenêtres exclusives, puis à demander aux utilisateurs exclusifs de planifier les observations restantes dans leurs fenêtres respectives (conjointement avec leurs propres observations). Ainsi, les utilisateurs exclusifs ne partagent pas leurs plans, seules les requêtes et observations non exclusives sont partagées avec les utilisateurs exclusifs intéressés. L'algorithme 3 esquisse cette approche, appelée *nex2ex*. Tout d'abord, le planificateur central positionne les observations non exclusives en dehors des fenêtres exclusives (ligne 1), puis chaque utilisateur exclusif planifie ses propres observations et les observations non exclusives restantes, en

fonction des observations déjà planifiées (lignes 2-3). Enfin, le planificateur central rassemble toutes les observations non exclusives planifiées (lignes 4-5). *nex2ex* a l'avantage de *de sécuriser les informations sur les horaires des utilisateurs exclusifs*. Seules les observations des utilisateurs non exclusifs dont les fenêtres temporelles chevauchent des portions d'orbite exclusives sont partagées par le solveur central. Toutefois, cela peut conduire à des *requêtes surbookées*, c'est-à-dire des requêtes avec plus d'une observation planifiée sur différentes fenêtres exclusives. De plus, la contrainte de respect de la capacité des satellites n'est pas garantie.

Algorithme 3 : Solveur *nex2ex*

Données : Un EOSCSP $P = \langle S, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$

Résultat : Une allocation \mathcal{M}

- 1 $\mathcal{M} \leftarrow \text{solve}(P[u_0])$
 - 2 **pour chaque** $u \in \mathcal{U}^{\text{ex}}$ **faire en parallèle**
 - 3 $\mathcal{M}_u \leftarrow \text{solve}(P[u, u_0] | \mathcal{M})$
 - 4 $\mathcal{M}'_u \leftarrow \{(o, t) \in \mathcal{M}_u | u_o \in \mathcal{U}^{\text{nex}}\}$
 $\quad // \text{ send } \mathcal{M}'_u \text{ to } u_0$
 - 5 **retourner** $\mathcal{M} \cup \bigcup_{u \in \mathcal{U}^{\text{ex}}} \mathcal{M}_u$
-

4.3 Résolution itérative

Pour éviter la surréservation de l'approche *nex2ex*, on peut considérer l'allocation comme un processus itératif, où les requêtes et les observations ne sont pas considérées par lots, mais une par une et envoyées à certains utilisateurs candidats. Cette approche, que nous appelons *itnex2ex*, est décrite dans l'algorithme 4.

Algorithme 4 : Solveur *itnex2ex*

Données : Un EOSCSP $P = \langle S, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$

Résultat : Une allocation \mathcal{M}

- 1 $\mathcal{M} \leftarrow \text{solve}(P[u_0])$
 - 2 $\mathcal{O}^{\text{sorted}} \leftarrow \text{sort}(\mathcal{O} \setminus \{o | (o, t) \in \mathcal{M}\})$
 - 3 **pour chaque** $o \in \mathcal{O}^{\text{sorted}}$ **faire**
 - 4 **pour chaque** $u \in \text{candidates}(o, \mathcal{U}^{\text{ex}})$ **faire**
 - 5 $\mathcal{M}_u \leftarrow \text{solve}(P[u, u_0] | \mathcal{M} \cup P[o])$
 - 6 $\mathcal{M}'_u \leftarrow \{(o, t) \in \mathcal{M}_u | u_o \in \mathcal{U}^{\text{nex}}\}$
 $\quad // \text{ envoyer } \mathcal{M}'_u \text{ à } u_0$
 - 7 **retourner** $\mathcal{M} \cup \bigcup_{u \in \mathcal{U}^{\text{ex}}} \mathcal{M}'_u$
-

Comme pour *nex2ex*, une première solution est obtenue pour les observations qui peuvent

être planifiées en dehors des fenêtres exclusives (ligne 1). Ensuite, les observations restantes sont triées, selon l'heure de début, la priorité ou tout autre critère (ligne 2). Cette liste triée est ensuite parcourue, et pour chaque observation, des utilisateurs exclusifs candidats sont choisis (ligne 4). Ce choix peut être arbitraire ou basé sur certains indicateurs comme la récompense ou la probabilité que l'observation soit de bonne qualité (si un tel modèle existe ou peut être appris). Dans cet article, nous allons considérer l'ordre lexicographique arbitraire. Chaque candidat essaie ensuite d'intégrer l'observation dans son plan (ligne 5), et envoie le résultat au planificateur central (ligne 6). Pour améliorer les performances, cela peut se faire en deux étapes : (1) essayer d'ajouter o de manière gloutonne et conservatrice, sans déplacer les observations déjà planifiées de u ; (2) réviser le plan de u pour intégrer o au cas où la première étape échouerait. Cela évite de recalculer le plan de u à chaque o interrogé. Enfin, la solution est construite en agrégeant les sous-solutions provenant d'utilisateurs exclusifs. *itnex2ex empêche la surréservation* par rapport à *nex2ex* en considérant itérativement certains utilisateurs candidats. La divulgation d'informations est équivalente à *nex2ex*, mais le calcul est synchrone et ne bénéficie pas de la distribution. Encore une fois, la contrainte de respect de la capacité des satellites n'est pas garantie.

5 Coordination par DCOP

Afin d'améliorer *itnex2ex*, nous étudions une approche orientée multi-agents et concevons un mécanisme de coopération entre utilisateurs exclusifs pour coordonner la planification.

5.1 À propos des DCOPs

Une façon de modéliser les problèmes de coordination entre agents consiste à les formaliser dans le cadre des problèmes d'optimisation distribuée sous contraintes (DCOP) [13].

Définition 8. Un *problème d'optimisation sous contraintes distribuées* discret (ou DCOP) est un tuple $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu \rangle$, où : $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$ est un ensemble d'agents ; $\mathcal{X} = \{x_1, \dots, x_n\}$ sont des variables appartenant aux agents ; $\mathcal{D} = \{\mathcal{D}_{x_1}, \dots, \mathcal{D}_{x_n}\}$ est un ensemble de domaines finis, tel que la variable x_i prend ses valeurs dans $\mathcal{D}_{x_i} = \{v_1, \dots, v_k\}$; $\mathcal{C} = \{c_1, \dots, c_m\}$ est un ensemble de contraintes, où chaque c_i définit un coût $\in \mathbb{R}^+ \cup \{+\infty\}$ pour chaque combinaison de l'affectation à un sous-ensemble de variables

(une contrainte est initialement connue seulement des agents impliqués) ; $\mu : \mathcal{X} \rightarrow \mathcal{A}$ est une fonction associant chaque variable avec l'agent qui la gère ; $f : \prod \mathcal{D}_{x_i} \rightarrow \mathbb{R}$ est un objectif représentant le coût global d'une affectation complète de valeurs aux variables. L'objectif d'optimisation est représenté par la fonction f , qui, en général, est considérée comme la somme des coûts : $f = \sum_i c_i$. Une *solution* à un DCOP P est une affectation complète à toutes variables. Une solution est *optimale* si elle minimise f .

Les DCOP ont été largement étudiés et appliqués dans de nombreux domaines de référence [5]. Ils ont de nombreuses propriétés intéressantes, comme : (i) une approche décentralisée où les agents négocient par des échanges locaux de messages ; (ii) une structuration du domaine (en l'encodant dans des contraintes) pour résoudre des problèmes complexes ; (iii) une grande variété de méthodes de résolution allant des méthodes exactes à des techniques heuristiques ou approchées ; comme, par exemple, ADOPT [10], DPOP [13], MaxSum [4], DSA [24] ou MGM [9], pour ne citer que les plus célèbres.

5.2 Étendre *itnex2ex* avec des DCOPs

Le problème avec *itnex2ex* est que la prise en compte itérative des observations (i) nécessite de choisir les candidats et leur ordre, et (ii) empêche d'exploiter la distribution de certains calculs. Nous allons donc examiner ici les requêtes de manière itérative, et laisser les utilisateurs exclusifs se coordonner pour choisir celui qui y répondra en planifiant une observation dans ses fenêtres temporelles exclusives. Ainsi, pour chaque requête provenant du planificateur central u_0 , un nouveau DCOP doit être résolu par l'ensemble des utilisateurs exclusifs intéressés, afin de choisir celui qui planifiera ou non une observation. L'algorithme 5 esquisse cette méthode, appelée *itnex2ex_DCOP*. Tout d'abord, le planificateur central tente de positionner autant d'observations que possible en dehors des fenêtres temporelles exclusives (ligne 1). Les utilisateurs exclusifs résolvent également leur propre sous-problème local simultanément (ligne 2). Ensuite, pour chaque requête r dans la liste ordonnée des requêtes restantes (ligne 3-4), une nouvelle instance DCOP est construite collectivement entre les utilisateurs exclusifs (ligne 5), puis résolue (ligne 6) en utilisant n'importe quel solveur DCOP disponible. Une fois que toutes les requêtes ont été examinées, le planificateur central rassemble les sous-solutions pour construire sa propre solution finale (ligne 7-8).

Algorithme 5 : Solveur itnex2ex_DCOP**Données** : Un EOSCSP $P = \langle \mathcal{S}, \mathcal{U}, \mathcal{R}, \mathcal{O} \rangle$ **Résultat** : Une allocation \mathcal{M}

```

1  $\mathcal{M} \leftarrow \text{solve}(\overline{P[u_0]})$ 
2 pour chaque  $u \in \mathcal{U}^{\text{ex}}$  faire en parallèle
    $\mathcal{M}_u \leftarrow \text{solve}(P[u])$ 
3  $\mathcal{R}^{\text{sorted}} \leftarrow \text{sort}(\mathcal{R} \setminus \{r \mid (o, t) \in \mathcal{M}, o \in \theta_r\})$ 
4 pour chaque  $r \in \mathcal{R}^{\text{sorted}}$  faire
5    $p \leftarrow$ 
      $\text{build\_DCOP}(\theta_r, \mathcal{M}, \mathcal{M}_{u_1}, \dots, \mathcal{M}_{u_n}, P)$ 
6    $\mathcal{M}_{u_1}, \dots, \mathcal{M}_{u_n} \leftarrow \text{solve\_DCOP}(p)$ 
7   pour chaque  $u \in \mathcal{U}^{\text{ex}}$  faire en parallèle
8      $\mathcal{M}'_u \leftarrow \{(o, t) \in \mathcal{M}_u \mid u_o \in \mathcal{U}^{\text{nex}}\}$ 
     // send  $\mathcal{M}'_u$  to  $u_o$ 
9 retourner  $\text{solve}(\overline{P[u_0]}) \mid \mathcal{M} \cup \bigcup_{u \in \mathcal{U}^{\text{ex}}} \mathcal{M}_u$ 

```

5.3 Modèle DCOP

Spécifions maintenant l'instance DCOP à construire à la ligne 5 de l'algorithme 5 pour une requête donnée r , et un plan courant $(\mathcal{M}, \mathcal{M}_{u_1}, \dots, \mathcal{M}_{u_n})$, comme requis dans la définition 8. L'ensemble des agents est l'ensemble des utilisateurs exclusifs qui peuvent potentiellement planifier la requête actuelle r :

$$\mathcal{A} = \{u \in \mathcal{U}^{\text{ex}} \mid \exists (s, (t_u^{\text{start}}, t_u^{\text{end}})) \in e_u, \exists o \in \theta_r$$

$$\text{t.q. } s_o = s, [t_u^{\text{start}}, t_u^{\text{end}}] \cap [t_o^{\text{start}}, t_o^{\text{end}}] \neq \emptyset\} \quad (13)$$

On note $\mathcal{O}[u]^r = \{o \in \theta_r \mid \exists (s, (t_u^{\text{start}}, t_u^{\text{end}})) \in e_u, \text{t.q. } s_o = s, [t_u^{\text{start}}, t_u^{\text{end}}] \cap [t_o^{\text{start}}, t_o^{\text{end}}] \neq \emptyset\}$ ces observations liées à la requête r qui peuvent être planifiées sur les fenêtres exclusives u .

La fonction μ associe chaque variable $x_{e,o}$ à son propriétaire e .

Les contraintes doivent vérifier qu'au maximum une observation par requête est planifiée (14), que les satellites ne sont pas surchargés (15), et qu'au maximum un agent sert la même observation (16).

$$\sum_{e \in \bigcup_{u \in \mathcal{A}} e_u} x_{e,o} \leq 1, \forall u \in \mathcal{A}, \forall o \in \mathcal{O}[u]^r \quad (14)$$

$$\sum_{o \in \{o \in \mathcal{O}[u]^r \mid u \in \mathcal{X}, s_o = s\}, e \in \bigcup_{u \in \mathcal{A}} e_u} x_{e,o} \leq \kappa_s^*, \forall s \in \mathcal{S} \quad (15)$$

avec κ_s^* la capacité courante du satellite s étant données les observations déjà planifiées $\mathcal{M}, \mathcal{M}_{u_1}, \dots, \mathcal{M}_{u_n}$.

$$\sum_{e \in \bigcup_{u \in \mathcal{A}} e_u} x_{e,o} \leq 1, \quad \forall o \in \mathcal{O} \quad (16)$$

En outre, le coût de l'intégration d'une observation dans l'emploi du temps de l'utilisateur actuel doit être évalué pour guider le processus d'optimisation. Nous ajoutons donc une contrainte souple à chaque $x_{e,o}$:

$$c(x_{e,o}) = \pi(o, \mathcal{M}_{u_o}), \quad \forall x_{e,o} \in \mathcal{X} \quad (17)$$

où π évalue le meilleur coût obtenu lors de la planification de o et toute combinaison d'observations de \mathcal{M}_{u_o} , afin de prendre en compte toutes les révisions possibles du plan actuel de u_o . En pratique, au lieu de calculer π à chaque fois, une compilation des contraintes peut être utilisée pour évaluer toutes ces combinaisons une seule fois. En pratique, le coût de chaque alternative parmi cet ensemble de taille exponentielle est calculé à l'aide d'un algorithme glouton polynomial.

Pour résumer, l'ensemble des contraintes du DCOP relatif à la requête r est :

$$\mathcal{C} = \{(14), (15), (16), (17)\} \quad (18)$$

6 Évaluation expérimentale

Les expérimentations visent à analyser les performances des algorithmes étudiés avec un nombre croissant de requêtes (et donc d'observations). Elles ont été développées en Python 3.7 et exécutées sur un processeur Intel(R) Xeon(R) E5-2660 v3 à 20 cœurs à 2,60 GHz et 62 Go RAM sous Ubuntu 18.04.5 LTS. Nous avons exécuté 30 instances d'EOSCSP générées de manière aléatoire avec une graine dans $[0 : 29]$ pour chaque taille de problème, et nous avons tracé la moyenne, avec un intervalle de confiance $[0.05, 0.95]$. La procédure `solve` utilisée dans `ex2nex`, `nex2ex`, `itnex2ex` et `itnex2ex_DCOP` est l'algorithme glouton greedy. L'algorithme DCOP utilisé par `itnex2ex_DCOP` est l'implémentation de DPOP [13] présente dans `pyDCOP` [15]. Les valeurs générées de manière aléatoire sont choisies uniformément dans les intervalles fournis.

6.1 Problèmes fortement conflictuels

Nous évaluons les algorithmes sur des problèmes très conflictuels à petite échelle (horizon

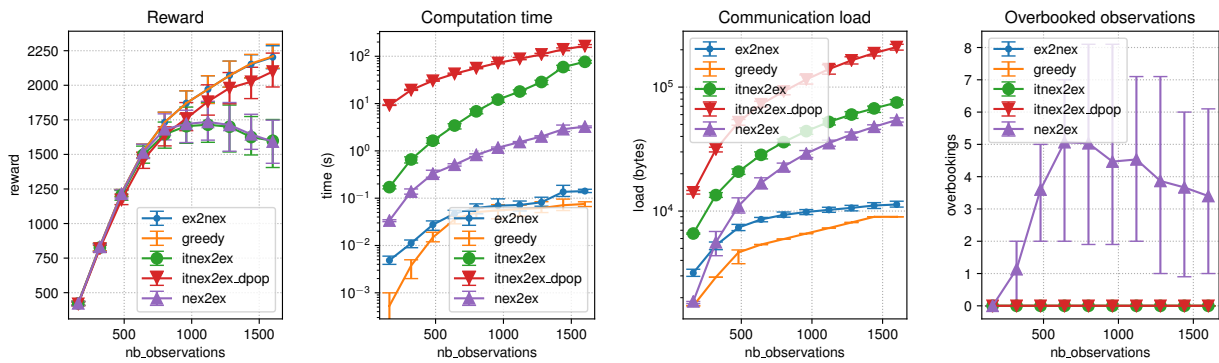


FIGURE 2 – Résultats des algorithmes étudiés sur des problèmes fortement conflictifs de petite taille.

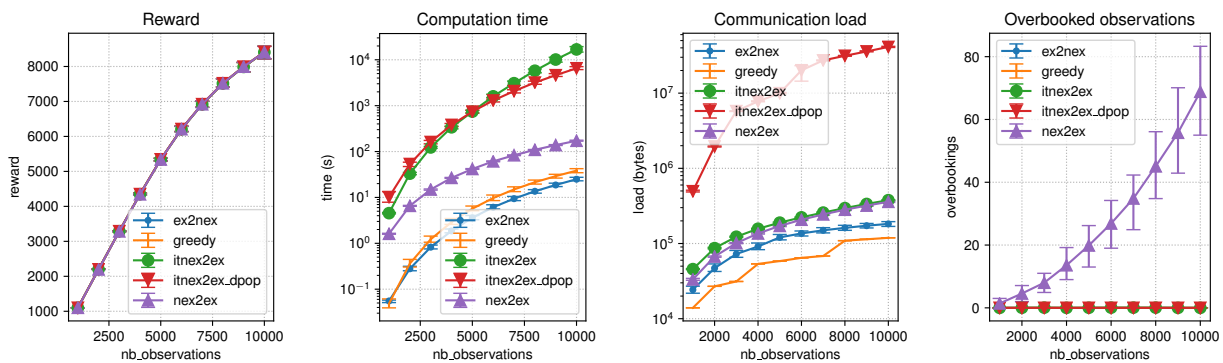


FIGURE 3 – Résultats des algorithmes étudiés sur des problèmes réalistes de grande taille.

de planification de 5 min). Nous générons des EOCSPP avec 3 satellites d'une capacité de 20 observations, 4 utilisateurs exclusifs émettant 2 à 20 requêtes chacun, 8 portions exclusives chacun d'une durée aléatoire dans [15 :20], un planificateur central émettant 8 à 80 requêtes, 10 opportunités d'observation par requête d'une durée égale à 5 qui peuvent être planifiées dans une fenêtre de temps d'une durée dans [10 :20], et une récompense dans [10 :50 :10] pour un utilisateur exclusif, et dans [1 :5] pour le planificateur central. La fenêtre temporelle des satellites est [0, 300]. Les temps de transition entre les observations sont uniformément égaux à 1. Les fenêtres exclusives sont positionnées de manière aléatoire, tout en veillant à ce qu'elles ne se chevauchent pas. Les fenêtres temporelles d'observation sont positionnées de manière aléatoire, de manière à garantir qu'elles sont soit incluses dans une fenêtre exclusive, soit en dehors de toute fenêtre exclusive. Il y a beaucoup de chevauchements d'observations, et autant de requêtes provenant du planificateur central que toutes les requêtes des utilisateurs exclusifs.

La figure 2 montre les résultats pour cette confi-

guration. En ce qui concerne les récompenses, itnex2ex_dpop offre des performances légèrement en-deça (environ 5.6%) des approches sans confidentialité (greedy et ex2nex). Les autres algorithmes avec confidentialité (nex2ex et itnex2ex) affichent des performances se dégradant avec une augmentation du nombre de requêtes. Ceci est dû au manque de coordination qui implique une augmentation du nombre de surbookings et de surcharge des satellites. Lorsque de tels cas surviennent, les observations en question sont retirées du plan, et donc les récompenses sont réduites. Les performances de itnex2ex_dpop sont au prix d'un temps de calcul supplémentaire, tout en restant raisonnable, contrairement au solveur optimal (par exemple CPLEX) qui ne peut pas résoudre les instances avec plus de observations (non affiché ici). Le temps de calcul plus élevé de itnex2ex_dpop résulte de la fonction de pré-calcul π et de la procédure de résolution DPOP sous-jacente. itnex2ex_dpop génère une charge de communication supplémentaire (environ 100kB pour les instances les plus grandes) en raison du processus itératif et de l'échange de messages DPOP. En résumé, itnex2ex_dpop est un candidat pertinent qui fournit de bonnes

solutions sans divulguer d'informations sur les utilisateurs exclusifs.

6.2 Problèmes réalistes

Ici, nous générons des EOSCSP de grande taille, avec des paramètres réalistes, dans le respect des carnets de commande fournis par nos partenaires, pour programmer des milliers d'observations dans un horizon de planification de 6 heures. Nous générons des instances comme précédemment mais avec 8 satellites d'une capacité de 500 observations, 5 utilisateurs exclusifs avec 10 à 150 requêtes chacun, 10 portions d'orbite exclusives chacun d'une durée dans [300 :600], 1 planificateur central avec 500 à 1000 requêtes, 5 opportunités d'observation par requête d'une durée égale à 20 qui peuvent être planifiées dans une fenêtre de temps d'une durée dans [40 :60]. L'horizon de planification est [0, 21600].

La figure 2 montre les résultats pour ce paramètre. Tous les algorithmes fournissent des solutions de bonne qualité, équivalentes à greedy, sauf nex2ex qui est environ 1,5% en dessous. ex2nex fournit même des résultats légèrement plus rapidement que greedy. Au niveau du temps de calcul, tous les algorithmes restent en dessous de 180 minutes de calcul, ce qui permet de les utiliser entre chaque fenêtre de planification (planifier les 6 heures suivantes pendant que le plan actuel de 6 heures est exécuté), même avec un code Python non entièrement optimisé. Dans ce contexte, itnex2ex_dpop est la meilleure approche pour fournir de bonnes solutions sans divulguer les informations des utilisateurs exclusifs, mais nécessite l'échange de 30 Mo pour des instances plus grandes. Ces instances ne sont pas trop conflictuelles, ce qui explique la progression quasi-linéaire de la récompense avec l'augmentation du nombre de requêtes. Cependant, les carnets de commande des années à venir devraient être de plus en plus conflictuels en raison du nombre croissant de clients de constellations EOS. Dans un tel cas, itnex2ex_dpop devient encore plus pertinent.

7 Conclusions

Ce papier étudie pour la première fois l'utilisation de techniques distribuées et multi-agents pour résoudre le nouveau EOSCSP, en gardant à l'esprit la nécessité de limiter la divulgation d'informations entre les utilisateurs. Nous avons défini les composantes essentielles de l'EOSCSP et proposé un codage MILP simple pour résoudre

de manière optimale ces problèmes. Ce système est malheureusement inutilisable dans la pratique, même seulement sur de petites instances. Nous avons donc proposé un algorithme glouton et rapide pour résoudre l'EOSCSP. Nous avons conçu plusieurs algorithmes distribués (ex2nex, nex2ex, itnex2ex et itnex2ex_dcop) avec différentes propriétés, notamment concernant la divulgation d'informations. itnex2ex_dcop offre des solutions équivalentes aux meilleurs algorithmes sur des problèmes très conflictuels, tout en préservant la confidentialité des informations d'observation des utilisateurs exclusifs. Ceci est dû au schéma de communication et à l'utilisation de DCOPs pour coordonner les utilisateurs exclusifs lorsqu'ils décident des observations à planifier. Cela a un coût : une charge de communication et un temps de calcul plus élevés pour évaluer la récompense que représente l'intégration d'une observation dans un plan donné. Pourtant, cette technique est entièrement distribuable et peut tirer profit d'une exécution simultanée. Sur des problèmes réalistes à grande échelle, la qualité de la solution est également bonne, bien que ces problèmes nécessitent moins de coordination car la probabilité de chevauchement des observations est plus faible. Les instances DPOP impliquent toujours de nombreuses observations d'utilisateurs exclusifs, ce qui rend le calcul de la fonction d'évaluation π et l'exécution de DPOP encore coûteux.

Ce travail nous permet d'identifier plusieurs perspectives, notamment le développement de solveurs DCOP dédiés et adaptés à la spécificité des EOSCSP, par exemple l'utilisation de la fonction d'évaluation π , qui peut résulter d'un processus d'apprentissage, au lieu d'une évaluation systématique de chaque alternative. Deuxièmement, nous envisageons de considérer un EOSCSP comme un problème d'optimisation de consensus où les utilisateurs exclusifs et le planificateur central se coordonnent en utilisant des méthodes de décomposition duale pour converger vers un accord sur la planification des observations. Enfin, nous travaillons actuellement à l'intégration des incertitudes sur le succès des observations dans le processus de décision.

Remerciements

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

Références

- [1] J. Bonnet, M.P. Gleizes, E. Kaddoum, S. Rainjoneau, and G. Flandin. Multi-satellite mission planning using a self-adaptive multi-agent system. In *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems*, pages 11–20, Boston, 2015. IEEE.
- [2] Kerri L. Cahoy and Andrew K. Kennedy. Initial Results from ACCESS : An Autonomous CubeSat Constellation Scheduling System for Earth Observation. In *Proc. of the 31st Annual AIAA/USU Conference on Small Satellites*, 2017.
- [3] Doo-Hyun Cho, Jun-Hong Kim, Han-Lim Choi, and Jaemyung Ahn. Optimization-based scheduling method for agile earth-observing satellite constellation. *Journal of Aerospace Information Systems*, 15(11) :611–626, 2018.
- [4] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*, pages 639–646, 2008.
- [5] F. Fioretto, E. Pontelli, and W. Yeoh. Distributed constraint optimization problems and applications : A survey. *Journal of Artificial Intelligence Research*, 61 :623–698, 2018.
- [6] Lei He, Liu Xiaolu, Gilbert Laporte, Ying-Wu Chen, and Yingguo Chen. An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. *Computers and Operations Research*, 100 :12–25, 07 2018.
- [7] M. Johnston. Scheduling nasa's deep space network : Priorities, preferences, and optimization. 2020.
- [8] Michel Lemaître, Gérard Verfaillie, Frank Jouhaud, Jean-Michel Lachiver, and Nicolas Bataille. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6(5) :367 – 381, 2002.
- [9] R.T. Maheswaran, J.P. Pearce, and M. Tambe. Distributed algorithms for dcop : A graphical-game-based approach. In *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems (PDCS)*, pages 432–439, 2004.
- [10] P.J. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT : Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence Journal*, 2005.
- [11] Sreeja Nag, Alan Li, Vinay Ravindra, Marc Sanchez Net, Kar-Ming Cheung, Rod Lammers, and Brian Bledsoe. Autonomous Scheduling of Agile Spacecraft Constellations with Delay Tolerant Networking for Reactive Imaging. In *Proceedings of the International Conference on Automated Planning and Scheduling SPARK Workshop*, 2019.
- [12] Peng Feng, Hao Chen, Shuang Peng, Luo Chen, and Longmei Li. A method of distributed multi-satellite mission scheduling based on improved contract net protocol. In *2015 11th International Conference on Natural Computation (ICNC)*, pages 1062–1068, Zhangjiajie, China, 2015. IEEE.
- [13] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 266–271, 2005.
- [14] G. Picard. Distributed and Multi-Agent Approaches for Scheduling Observations in Satellite Constellations with Exclusive Orbit Portions. In *International Conference on Artificial Intelligence (IJCAI-21)*, page 7, 2021. in submission.
- [15] Pierre Rust, Gauthier Picard, and Fano Ramparany. pyDCOP, a DCOP library for IoT and dynamic systems. In *International Workshop on Optimisation in Multi-Agent Systems (OptMAS@AAMAS 2019)*, 2019.
- [16] Vishwa Shah, Vivek Vittaldev, Leon Stepan, and Cyrus Foster. Scheduling the world's largest earth-observing fleet of medium-resolution imaging satellites. *IWPSS*, 2019.
- [17] P. K. Sinha and A. Dutta. Multi-satellite task allocation algorithm for earth observation. In *2016 IEEE Region 10 Conference (TENCON)*, pages 403–408, New York, New York, US, 2016. IEEE.
- [18] C. Sun, X. Wang, and X. Liu. Distributed satellite mission planning via learning in games. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4381–4386, New York, New York, US, 2018. IEEE.
- [19] Panwadee Tangpattanakul, Nicolas Jozefowicz, and Pierre Lopez. A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite. *European Journal of Operational Research*, 245(2) :542–554, September 2015.
- [20] Alex Elkjær Vasegaard, Mathieu Picard, Florent Hennart, Peter Nielsen, and Subrata Saha. Multi criteria decision making for the multi-satellite image acquisition scheduling problem. *Sensors*, 20(5) :1242, 2020.
- [21] J. G. Walker. Satellite Constellations. *Journal of the British Interplanetary Society*, 37 :559, December 1984.
- [22] Xinwei Wang, Guohua Wu, Lining Xing, and Witold Pedrycz. Agile earth observation satellite scheduling over 20 years : formulations, methods and future directions. *CoRR*, abs/2003.06169, 2020.
- [23] W. Yang, Y. Chen, R. He, Z. Chang, and Y. Chen. The Bi-objective Active-Scan Agile Earth Observation Satellite Scheduling Problem : Modeling and Solution Approach. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–6, July 2018.
- [24] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. Distributed stochastic search and distributed breakout : Properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2) :55–87, January 2005.

Un modèle agent générique pour la comparaison d'approches d'allocation de ressources dans le domaine du transport à la demande

Alaa Daoud^a
alaa.daoud@emse.fr

Flavien Balbo^a
flavien.balbo@emse.fr

Paolo Gianessi^a
paolo.gianessi@emse.fr

Gauthier Picard^b
gauthier.picard@onera.fr

^aMines Saint-Étienne, CNRS, UMR 6158, LIMOS
Institut Henri Fayol, Saint-Étienne, France

^bONERA-DTIS, Université de Toulouse

Résumé

Les problèmes d'allocation sont d'importance majeure dans la gestion des systèmes de transport à la demande (ODT). Ils ont été étudiés pendant des décennies, et diverses solutions ont été proposées. Les approches de résolution peuvent être classées en deux catégories : centralisées et décentralisées. Dans la pratique, chacune a ses avantages et désavantages. Dans ce travail, nous visons à fournir un modèle générique pour le problème du transport à la demande en ligne avec des véhicules autonomes et un modèle multi-agent dédié à l'allocation des ressources et à la planification d'une flotte de véhicules. Ce dernier considère des véhicules autonomes qui communiquent dans un réseau inter-véhiculaire pour satisfaire les demandes de course dans un système ODT. Nous évaluons la généricité de ce modèle en appliquant plusieurs approches d'allocation (programmation mathématique, heuristique gloutonne, optimisation par contraintes distribuées et enchères) et comparons leurs performances sur des scénarios synthétiques générés à partir de données réelles.

Mots-clés : Transport à la demande, véhicules autonomes, modèle agent

Abstract

Allocation problems are major issues in managing On-Demand Transport (ODT) systems. They have been studied for decades, and a variety of solutions were proposed. The Approaches to these problems can be classified as centralized and decentralized. Each has its pros and cons in practice. In this work, we aim to provide a generic model for the problem of online on-demand transport with autonomous vehicles and a multi-agent model dedicated to resource allocation and scheduling in vehicle fleets. This model considers autonomous vehicles that com-

municate via an inter-vehicular network satisfy trip requests in an online ODT system. We validate this model's genericity by applying several allocation approaches (mathematical programming, greedy heuristic, distributed constraint optimization, and auctions) and compare their performance on synthetic scenarios in a real-world city map.

Keywords: On-demand transport, autonomous vehicles, agent model

1 Introduction

Un véhicule autonome (AV) est un véhicule sans conducteur qui peut avoir d'autres capacités que la conduite, comme par exemple le choix de son itinéraire en fonction de l'état du trafic, la coordination et la coopération avec d'autres véhicules ou encore la décision de ses propres horaires de voyage dans le cas d'un taxi autonome.

Parmi les questions importantes dans la gestion des systèmes de transport à la demande (ODT) nous retrouvons les problèmes d'allocation de demandes de courses aux véhicules qui soient réalisables et efficaces. Pour des taxis autonomes, cela signifie qu'ils peuvent être responsables de leur choix d'affectation aux demandes (en prenant des décisions décentralisées) ou suivre les plannings qui sont décidés de manière centralisée par un répartiteur. En pratique, la faisabilité et l'efficacité du choix de centraliser/décentraliser la solution dépendent de la complexité du problème, de ses contraintes et de la dynamique de l'environnement.

Dans cet article, (i) nous proposons AV-OLRA, un modèle générique pour le problème d'allocation des ressources en ligne avec des véhicules autonomes. Ce modèle définit les données du problème (composantes, contraintes) et les indicateurs permettant d'évaluer les différentes

1. Version étendue du papier court AAMAS [2]

stratégies d'allocation; (ii) nous proposons à ce problème un modèle multi-agent générique support de solution au problème ODT, où les véhicules autonomes (agents) communiquent avec leurs voisins via une communication pair-à-pair par le biais d'ensembles connectés; (iii) nous classons les différentes méthodes d'allocation en fonction du comportement de coordination des agents (sans coordination, passif, coopératif); (iv) et nous évaluons et comparons expérimentalement différentes méthodes de résolution (centralisée, gloutonne, enchères et optimisation sous contraintes distribuée).

Le papier est structuré comme suit. La section 2 présente quelques travaux relatifs à l'allocation de ressources multi-agents et à l'ODT. Sur la base de cette analyse de l'état de l'art, nous exposons le problème AV-OLRA dans la section 3, et un modèle multi-agent générique pour le résoudre dans la section 4. La section 5 examine plus en détail les différents mécanismes de coordination investigués dans cette étude, qui sont ensuite évalués expérimentalement dans la section 6. Enfin, le document se termine par quelques perspectives dans la section 7.

2 Allocation de ressources multi-agents et ODT

Ces dernières années, le nombre d'articles consacrés à l'application des technologies fondées sur les agents dans le domaine des transports a considérablement augmenté.

Les simulations et modèles ODT existants ont été décrits dans [14]. Dans ce domaine, mieux répondre à la demande est considéré comme un défi et doit prendre en compte les personnes avec leur comportement et leur interaction avec un environnement de transport complexe. La vision MARA (*Multi-Agent Resource Allocation*) est pertinente pour la résolution des problèmes ODT et a été mise en oeuvre selon différentes approches. Ainsi, la centralisation du processus d'allocation avec un répartiteur automatique est encore assez courante dans les approches multi-agents [3, 11, 17]. D'autre part et pour réaliser une planification en temps réel des services ODT, plusieurs modèles décentralisés ont été proposés [6, 16]. Un modèle théorique de système de transport est développé dans [8] pour étudier le comportement de coopération des véhicules, avec une perspective globale; la meilleure efficacité des véhicules coopérants devrait être de partager les connaissances dans un

transport public flexible. Au contraire, en l'absence de communication entre les agents, l'applicabilité de la programmation génétique pour développer des systèmes multi-agents décentralisés qui résolvent les problèmes dynamiques de ODT a également été étudiée [19]. Ils ont conclu que la planification à long terme n'est pas bénéfique dans de tels contextes en raison de la très forte dynamique; ainsi, les agents ne devraient examiner qu'une seule demande à l'avance.

L'un des principaux défis que pose l'utilisation des approches MARA, plus généralement multi-agents, pour résoudre les problèmes liés à l'ODT est le goulot d'étranglement en matière de communication. Une solution est une organisation spatiale des agents, par exemple, dans [7] il est proposé que chaque agent dispose d'une zone de planification limitée et n'ait pas besoin de communiquer avec tous les autres. Un second défi concerne la dynamique du système. Les solutions aux problèmes d'allocation de ressources ODT dans des environnements dynamiques doivent remettre en question les plannings des véhicules en temps réel. Cette remise en cause rend la réalisation d'une solution optimale un objectif non réalisable dans la pratique. Cependant, la conception d'approches itératives pour obtenir des solutions réalisables en temps raisonnable est une alternative appropriée pour aborder l'aspect dynamique; cela nécessite de considérer le besoin de communication et donc de fournir des schémas de communication et de coordination solides et efficaces.

De même, un cadre de modélisation générique, c'est-à-dire indépendant de la solution/stratégie, du problème d'allocation de ressources localisées en ligne (OLRA), et un système multi-agent pour résoudre le problème de la gestion du stationnement urbain ont été proposés dans [21]. La solution repose sur une communauté de conducteurs qui partagent leurs connaissances locales sur la disponibilité des places de stationnement. Notre travail s'appuie sur ce dernier modèle pour proposer un modèle spécifique au problème de l'allocation des ressources à la demande dans les flottes de véhicules autonomes.

3 Modèle de problème AV-OLRA

Dans cet article, nous définissons le problème AV-OLRA comme une spécialisation du modèle OLRA pour l'allocation de ressources en ligne avec des véhicules autonomes, et une extension pour la prise en compte de la communication et la modélisation de contraintes temporelles sup-

plémentaires.

Nous formulons donc le problème AV-OLRA comme suit :

$$\text{AV-OLRA} := (\mathcal{R}, \mathcal{V}, \mathcal{G}, \mathcal{T}) \quad (1)$$

$$\mathcal{R} = \{r_i | i \in \mathbb{N}\} \quad (2)$$

$$\mathcal{V} = \{v_i | i \in \mathbb{N}\} \quad (3)$$

$$\mathcal{G} = (\mathcal{N}, \mathcal{E}, \omega) \quad (4)$$

$$\mathcal{T} := \{t_0, t_1, \dots, t_{end}\} \quad (5)$$

où \mathcal{R} définit un ensemble dynamique de demande des passagers (les ressources dans le modèle OLRA) qui sont disponibles sur une fenêtre temporelle spécifique au moment de l'exécution; l'ensemble des consommateurs/véhicules \mathcal{V} (l'ensemble des consommateurs \mathcal{C} dans le modèle OLRA) représente une flotte de m véhicules autonomes qui sont mobiles et ne peuvent communiquer qu'à une portée limitée; \mathcal{G} est un graphe dirigé, avec \mathcal{N} l'ensemble des nœuds, et \mathcal{E} l'ensemble des arcs, $e_{ij} \in \mathcal{E}$ est l'arc entre les nœuds i et j , ω est une fonction d'évaluation qui associe à chaque arc $e \in \mathcal{E}$ une valeur ω_e sur la base d'une mesure de distance temporelle (par exemple, le temps de conduite moyen en minutes), qui sera utilisée pour calculer les coûts opérationnels des déplacements des véhicules. Enfin \mathcal{T} , est l'horizon temporel du problème.

Definition 1 *Un véhicule autonome $v \in \mathcal{V}$ est caractérisé par sa capacité $c : \mathcal{V} \rightarrow \mathbb{N}$, son coût de trajet par distance parcourue $cpd : \mathcal{V} \rightarrow \mathbb{R}^+$ et une portée de communication limitée $rng : \mathcal{V} \rightarrow \mathbb{R}^+$ qui ne change jamais avec le temps*

$$v := (c_v, cpd_v, rng_v)$$

ainsi qu'un ensemble de propriétés dépendantes du temps qui sont sa localisation actuelle $loc : \mathcal{V} \times \mathcal{T} \rightarrow \mathcal{N} \cup \mathcal{E}$, sa destination actuelle, $dest : \mathcal{V} \times \mathcal{T} \rightarrow \mathcal{N}$ et le nombre de sièges actuellement disponibles $seats : \mathcal{V} \times \mathcal{T} \rightarrow \mathbb{N}$.

La communication entre deux composants du système est réalisée si la distance qui les sépare est inférieure ou égale à leur portée de communication. Cependant, comme la portée de communication des véhicules est limitée, et pour maximiser leur connectivité, deux véhicules peuvent être connectés par transitivité. Cela conduit à la définition suivante d'un ensemble connecté :

Definition 2 *Un ensemble connecté (noté CS pour connected set) est un ensemble d'entités connectées directement ou par transitivité.*

Les CSs sont des entités dynamiques; ils sont créés, divisés, fusionnés en cours d'exécution en fonction du mouvement des véhicules. Ainsi, selon les définitions précédentes, un véhicule ne peut communiquer au temps t qu'avec les membres de son CS par des messages directs ou diffusés. La portée de communication limitée divise implicitement la flotte en plusieurs ensembles connectés.

Definition 3 *Une solution à un problème AV-OLRA est définie pour chaque ensemble connecté comme une agrégation des allocations de tous les consommateurs de cet ensemble qui évite tous les conflits.*

Cette définition implique qu'une solution à un problème AV-OLRA défini pour des véhicules et des demandes peut être sous-optimale parce que plusieurs véhicules considèrent les mêmes demandes ou parce que la solution optimale n'est pas l'union des sous-solutions optimales de chaque CS. En outre, toute solution dépend du temps à cause de l'aspect *en ligne* du problème.

La valeur de la portée de communication dépend uniquement de la technologie de communication utilisée, et la considérer dans notre modèle ajoute une autre dimension de généralité. Plus la portée de communication est faible, plus il existe d'ensembles connectés; cela signifie que pour une communication à courte portée, même avec des approches centralisées, le calcul de la solution est décentralisé vers plusieurs répartiteurs. Une flotte ayant une portée de communication suffisamment longue pourrait revenir à un seul ensemble connecté à l'échelle d'une ville avec un partage global des connaissances. Une approche centralisée conduit alors à un répartiteur central pour déterminer la solution globale.

La qualité d'une allocation est caractérisée par des indicateurs fonctionnels et techniques dont le calcul est indépendant des approches de résolution mais qui permet de comparer leur faisabilité et leur qualité. Les indicateurs fonctionnels sont des mesures de l'optimalité du processus d'allocation défini par sa fonction objective, tandis que les indicateurs techniques sont utilisés pour évaluer la faisabilité et l'applicabilité du processus d'allocation et pour prévoir ses coûts opérationnels dans différents contextes.

Dans ce papier, nous caractérisons la qualité d'une solution AV-OLRA dans les scénarios ODT par les indicateurs suivants : **Quality** est le pourcentage de demandes satisfaites (consommées) sur toutes les demandes annoncées. Par

conséquent, cet indicateur indique le niveau de qualité de service (QoS). **Utility** est l'*utilité totale* des plannings des véhicules, dérivée des distances des voyages réalisés (effectués avec un passager à bord, de la source à la destination), qui définit le **gain** pour l'entreprise. **Cost** est le *coût opérationnel*, dérivé des distances totales parcourues par les véhicules. La relation entre les indicateurs **Utility** et **Cost** définit la qualité des affaires (QoB). **MsgCount** est le *nombre total de messages* échangés au cours du processus d'allocation. **MsgSize** est la *taille moyenne des messages* échangés pendant l'allocation. Ces deux derniers indicateurs de communication estiment le coût technique de la solution et permettent de prédire si elle est applicable en termes de charge de communication, c'est-à-dire si elle pourrait provoquer des goulets d'étranglement critiques.

4 Approche multi-agent pour AV-OLRA

Dans cette section, nous décrivons notre modèle multi-agent au problème AV-OLRA. L'environnement du problème AV-OLRA représente la topologie de l'infrastructure urbaine \mathcal{G} et le modèle de communication des agents tel que décrit par la définition 2.

Il n'y a qu'un seul type d'agents dans notre modèle. Un agent véhicule autonome (AV) est associé à chaque véhicule du système. Nous pouvons distinguer trois sous-comportements différents (*acting*, *communicating* et *planning*). Comme nous modélisons AV-OLRA dans un espace de temps discret, l'horizon temporel est défini comme un ensemble de pas d'exécution. À chaque pas, chaque agent effectue les actions suivantes, comme le montre la figure 1(a) :

1. lire les messages reçus et mettre à jour le contexte (sous-comportement de communication, *communicating*);
2. choisir les lieux à visiter (sous-comportement de planification, *planning*);
3. agir en effectuant une action de conduite (sous-comportement d'action, *acting*);
4. diffuser ses informations contextuelles (sous-comportement de communication, *communicating*).

4.1 Sous-comportement d'action

En fonction de la présence de passagers à bord, de la localisation du véhicule et de sa connais-

sance des demandes à venir, un agent AV peut se trouver dans l'un des états suivants (figure 1(b)) :

Marauding : le véhicule n'a pas de passager à bord et cherche sa prochaine destination;

Moving : le véhicule a une destination et se déplace dans la topologie urbaine vers cette destination;

Picking up : le véhicule est situé à l'emplacement d'origine de la demande du passager p afin d'effectuer l'action $\text{pick_up}(p)$ puis de reprendre son déplacement;

Dropping off : le véhicule est situé à l'emplacement de destination de la demande du passager p afin d'effectuer l'action $\text{drop_off}(p)$ puis de rechercher une nouvelle destination.

Les transitions entre ces états sont illustrées dans la figure 1(b).

4.2 Sous-comportement de communication

En tant qu'agents communicants, les AV ont un comportement de communication avec les autres entités environnantes; ils peuvent rejoindre/quitter des ensembles connectés (CS), diffuser, envoyer et recevoir des messages.

- $\text{join}(c)$: l'agent rejoint le CS c du fait qu'il se trouve dans le rayon de communication d'au moins un de ses membres;
- $\text{leave}(c)$: l'agent quitte le CS c car il n'est plus dans le rayon de communication d'au moins un membre;
- $\text{send}(m, a)$: l'agent envoie un message m à un autre agent a à condition qu'ils soient dans le même CS;
- $\text{receive}(m)$: l'agent reçoit un message m d'un autre agent de son CS (une fois reçu et lu, le message est stocké dans la base de croyances de l'agent);
- $\text{broadcast}(m)$: similaire à $\text{send}(m, a)$ mais ici l'agent ne spécifie pas l'agent récepteur, il diffuse plutôt le message à l'ensemble des membres de son CS.

4.3 Sous-comportement de planification

Le comportement des AVs en matière de planification dépend du mécanisme d'allocation choisie (centralisée/décentralisée, coopérative/compétitive, avec/sans modèle de coordination). La figure 1(c) illustre le comportement de planification abstrait et générique des AVs. Pour mettre à jour son planning, un AV recherche en permanence des options de planification. Si une

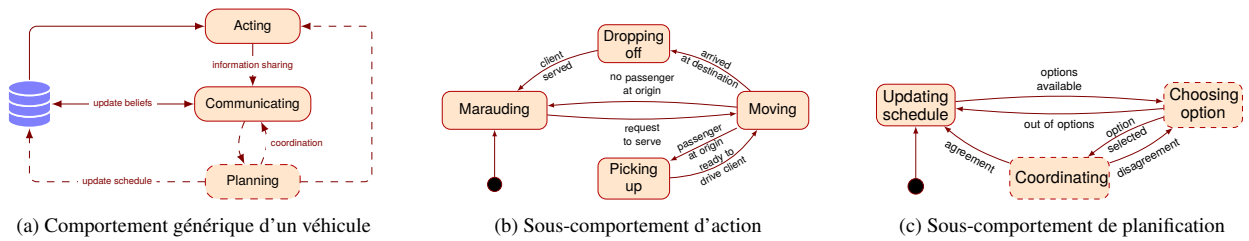


FIGURE 1 – Comportements des AV (en pointillés pour les génériques, à instancier pour toute stratégie spécifique)

option est trouvée, l’AV en sélectionne une et communique (ou pas selon le modèle de coordination) sa décision à ses voisins (les autres agents de son CS). Le voisinage parvient à un accord ou à un désaccord, selon le mécanisme de coordination et l’option choisie. En cas d’accord, l’AV met à jour son planning et recherche l’option suivante et ce, jusqu’à ce qu’aucune option ne soit disponible. La nature des options de planification dépend également du mécanisme de coordination.

Bien que nous soutenions plusieurs modèles de coordination, nous souhaitons étudier chacun et non les conséquences de leurs interactions. Par conséquent, nous considérons que l’ensemble des agents de la flotte est homogène. Dans la section 5, nous présentons en détail différents mécanismes de coordination que nous utilisons pour valider notre modèle, y compris certaines approches coopératives avancées comme l’usage d’algorithmes DCOP et un mécanisme d’enchères pour le comportement de coordination des agents. Ce dernier répond aux exigences du problème AV-OLRA en fournissant une solution utilisant des calculs légers, dynamiques et continuellement sujets à amélioration.

4.4 Utilité, contraintes et objectif

Dans tout problème MARA, la fonction d’utilité représente le degré de satisfaction d’un agent pour une allocation donnée [1]. Chaque agent a une valeur d’utilité exprimée sous la forme d’une valeur explicite ou d’une relation qui révèle la solution la plus satisfaisante (optimale). Une procédure d’allocation tente de fournir aux agents des ressources qui correspondent autant que possible à leur exigence. Dans notre modèle, nous définissons la fonction d’utilité des agents AV sur la base des indicateurs de la qualité de solution décrits dans la section 3. Nous considérons que plus un agent satisfait de demandes, plus il doit gagner en valeur d’utilité. Ainsi, l’indicateur **Quality**, s’il est considéré individuellement pour chaque agent $a \in \mathcal{V}$, définit

son utilité :

$$u_a = \frac{|R_a|}{|K_a|} \quad (6)$$

où R_a est l’ensemble des demandes qui sont attribuées à a , i.e choisie par l’agent ou le répartiteur, et K_a est l’ensemble de toutes les demandes connues par a . Bien entendu, l’attribution de la demande r à l’agent a est contrainte par la disponibilité spatiale et temporelle de a et de r . Nous considérons que l’origine et la destination de chaque demande sont des constantes, et qu’une demande n’est disponible pour être prise en charge qu’à son origine pendant sa fenêtre temporelle définie $w_r[l_r, u_r]$. Ainsi, l’affectation de r à a exige que a puisse arriver au point d’origine de r à un moment t situé entre la limite inférieure l_r et la limite supérieure u_r de la validité temporelle de r .

Dans ce document, nous n’envisageons pas de scénarios de partage de véhicules. Nous supposons qu’un trajet en véhicule est consacré à une seule demande, mais nous devons tout de même tenir compte de la taille de la demande (nombre de sièges requis) et de la capacité du véhicule. Cela implique que la définition de la disponibilité doit également inclure la contrainte de capacité :

$$\exists t \in w_r[l_r, u_r] : \text{seats}(a, t) \geq s_r \ \& \ \text{loc}(a, t) = o_r \quad (7)$$

Le fait d’être membre d’une flotte impose aux AV d’être coopératifs et de suivre le mécanisme de coordination prédéfini pour atteindre leur objectif global. Dans une perspective globale d’ODT en tant que modèle commercial, l’objectif principal des prestataires de services ODT est de gagner la satisfaction des utilisateurs. Cela signifie que leur objectif est de réduire les coûts et d’augmenter les gains. De ce point de vue, nous pouvons définir la fonction objective \mathcal{F} à maximiser par le processus d’allocation fondé sur la

relation entre les indicateurs **Utility** et **Cost** :

$$\mathcal{F} = \sum_{r \in \mathcal{R}_s} (P + p * \text{dist}(r)) - \sum_{v \in \mathcal{V}} cpd_v * \text{driven}(v) \quad (8)$$

où $\mathcal{R}_s \subseteq \mathcal{R}$ est l’ensemble de toutes les demandes satisfaites, P est un prix fixe (frais de service) par demande, p est un facteur de tarification par unité de distance parcourue, $\text{dist}(r)$ est la distance totale parcourue pour une demande r et $\text{driven}(v)$ est la distance totale parcourue par v .

5 Mécanismes de coordination

Cette section illustre certains comportements de coordination que les flottes de véhicules suivent habituellement pour atteindre un objectif global d’allocation. Pour chacun, nous présentons le modèle de coordination correspondant. Un mécanisme de coordination est défini par trois composantes $\langle DA, AC, AM \rangle$, où DA indique le niveau d’autonomie de décision qui est soit centralisé (C) soit décentralisé (D); AC indique si il y a coopération des agents avec (S) ou sans partage (N) des informations sur les plannings, et AM est le nom du processus d’allocation.

5.1 Comportement égoïste

Le mécanisme de coordination noté $\langle D, N, Greedy \rangle$ est fondé sur un processus d’allocation décentralisé avec des agents compétitifs et sans coordination explicite. Dans ce mécanisme, les agents ne s’appuient pas sur les décisions des autres et n’échangent jamais leurs plans. Dans les scénarios du monde réel, une stratégie de ce modèle est basée sur l’avidité, dans laquelle le véhicule ne considère qu’une seule demande à l’avance (par exemple, la plus proche, afin de raccourcir la distance de conduite à vide). [19] est un exemple de mise en œuvre de ce mécanisme. Lorsqu’un véhicule ne transporte pas déjà des clients, il doit décider quelle demande il traitera en premier, en fonction des informations dont il dispose sur les demandes disponibles. Une heuristique calcule une valeur de priorité pour chaque demande. Ensuite, l’agent traite en premier la demande ayant la valeur de priorité la plus élevée. Des conflits peuvent survenir, mais ils sont résolus simplement en appliquant la politique du « premier arrivé, premier servi ». L’ensemble

des options d’un agent est son ensemble de demandes connues et réalisables, si l’agent n’a pas de passager à bord, sinon aucune option n’est prise en compte. L’état *coordinating* est ignoré (comme s’il parvenait à un accord pour toute option choisie), de sorte que la qualité de la solution dépend de la stratégie de l’agent pour choisir la demande suivante.

5.2 Comportement avec répartiteur

Ici, le mécanisme de coordination est centralisé, le rôle de l’agent est de mettre à jour son planning en fonction de ce qu’il reçoit du répartiteur. $\langle C, S, MILP \rangle$ est un exemple de ce type de mécanisme avec MILP comme processus d’allocation. Dans notre modèle, nous avons besoin d’un répartiteur par ensemble connecté (CS). Ainsi, lors de la création (ou de la mise à jour), un membre d’un CS (par exemple, celui qui a l’indice le plus bas dans l’ensemble) devient le répartiteur qui sera responsable de collecter les informations des autres agents et sur demande. Il doit aussi faire le calcul d’allocation par lui-même, ou en appelant un service externe pour obtenir une allocation optimale (résolvant un MILP), puis envoyer à chaque autre véhicule son planning potentiel, comme dans [4, 10, 20]. Dans ce modèle centralisé, le rôle du comportement de planification des AVs est de demander à un portail (l’agent responsable) de mettre à jour leur planning en permanence. Dans ce cas, la seule option disponible est de requêter le portail, et l’état *coordinating* consiste en un protocole de demande/réponse qui enverra le nouveau planning sous forme d’accord.

5.3 Comportements réellement coordonnés

Dans ce cas, de type $\langle D, S, PC \rangle$, le mécanisme de décision est décentralisé, les agents sont coopératifs et un protocole de coordination (PC) est appliqué pour l’allocation. Dans cette catégorie de mécanismes de coordination, les agents échangent des informations et coopèrent pour atteindre un objectif commun, en évitant les conflits et en optimisant la qualité de la solution. Il existe plusieurs approches pour atteindre ce comportement, comme l’optimisation sous contraintes distribuées (DCOP) [5], les protocoles de négociation [16, 3] et les enchères [15, 12]. Nous instancions ici des solutions par enchères et par DCOP.

Coordination par enchères. Les enchères sont très courantes dans les situations quotidiennes

et fournissent une base conceptuelle générale pour comprendre les problèmes d'allocation des ressources au sein d'ensembles d'agents [18]. Nous présentons ici un exemple de mécanisme collaboratif de construction de plannings de véhicules, noté $\langle D, S, \text{Auction} \rangle$, fondé sur des enchères pour coordonner de manière pair-à-pair les décisions de planification de flottes de véhicules autonomes. Ce mécanisme est proposé pour fonctionner dans un cadre dynamique, entre des agents véhicules qui appartiennent à un ensemble connecté dans lequel ils peuvent recevoir et envoyer des messages directs ou diffusés. Les agents intéressés par une demande donnée lancent des enchères au premier prix pour cette demande, et le gagnant l'ajoute à son planning. La détermination du gagnant est un processus complètement décentralisé. Afin d'améliorer l'efficacité de la planification dans des contextes dynamiques, les agents sont autorisés à échanger leurs demandes planifiées au moment de l'exécution, avec des tours d'enchères supplémentaires pour décider si cet échange augmente la valeur de la fonction objectif au sein du CS. Les agents communiquent entre eux par des messages directs ou indirects pour partager des informations ou coordonner leurs décisions.

Coordination par DCOP. Dans $\langle D, S, \text{DCOP} \rangle$, les agents décident seuls mais se coordonnent avec les agents du même ensemble connecté en utilisant un algorithme d'optimisation sous contraintes distribuée afin d'éviter les conflits au sein du CS. À chaque fois qu'un ensemble connecté change, un DCOP $P = \langle A, X, D, C \rangle$ est généré à partir de l'instance AV-OLRA pour maximiser la fonction objectif dans l'équation 8, comme suit. A définit l'ensemble des agents dans l'ensemble connecté. X définit l'ensemble des variables de décision dans trois sous-ensembles (x_{ij} 's, y_{ij} 's et z_{ij} 's) : $x_{ij} \in X$ est une variable binaire égale à 1 si le véhicule v_i sert la requête r_j ; y_{ij} est une variable binaire égale à 1 seulement si la requête r_j est la première requête à être servie par v_i . Enfin, z_{ij} est une variable entière qui définit à quel moment une requête r_j est visitée par v_i . D définit les domaines des variables : $\{0, 1\}$ pour les x_{ij} et les y_{ij} , et un ensemble de domaines de plages de temps définissant la fenêtre $[l_j, u_j]$ pour chaque z_{ij} . C définit l'ensemble des contraintes, qui se compose de contraintes dures (capacité, disponibilité spatio-temporelle et fenêtres temporelles) et de contraintes souples définissant le coût et l'utilité de la décision d'allocation (utilisées pour calculer la valeur de la fonction objectif).

Les algorithmes DCOP sont variés, et le choix dépend de l'objectif de la solution et du contexte du problème. Les caractéristiques d'exécution de l'algorithme (temps d'exécution, nombre/taille des messages et besoin en mémoire par agent) sont des facteurs essentiels pour traiter les problèmes dynamiques en ligne. Compte tenu des caractéristiques de notre problème, nous avons choisi de mettre en œuvre des algorithmes de recherche locale pour résoudre l'AV-OLRA, à savoir MGM [13] et DSA [22].

6 Évaluation expérimentale

Dans cette section, nous présentons les résultats expérimentaux de l'instanciation du modèle AV-OLRA avec le modèle multi-agent décrit dans la section 4, en prenant en charge les différents types de mécanismes de coordination de la section 5. Le modèle est mis en œuvre en tant que système multi-agent avec un simulateur de transport en temps discret.

Cadre expérimental. Nous utilisons la même carte de réseau urbain pour toutes nos expériences. Plus de 1400 arcs ont été extraits d'Open Street Map (OSM)¹ et post-traités pour un quartier situé entre (45.4325,4.3782) et (45.437800,4.387877) pour produire un graphe formé de 71 arcs. Les requêtes des passagers sont générées aléatoirement avec des lieux de ramassage et de livraison appartenant à un ensemble spécifique de lieux appelés *sources*. 40 emplacements uniformément répartis sur la carte ont été sélectionnés pour être des sources d'émission de la demande. Lorsque les véhicules doivent échanger directement des messages, nous considérons qu'ils communiquent via DSRC² avec une portée de communication réaliste de 250 mètres. Le nombre de demandes générées et le nombre de véhicules sont des paramètres de la simulation. Tous les scénarios ont une durée de 1000 cycles et, à chaque cycle, 0 ou 1 demande est générée. Nous évaluons la performance de 5 mécanismes de coordination vue en section 5 dont celui fondé sur un DCOP avec l'algorithme DSA, variante A, $p = 0.5$ [22] $\langle D, S, \text{DCOP}(\text{dsa}) \rangle$, et celui fondé sur un DCOP avec l'algorithme MGM-2 [13] $\langle D, S, \text{DCOP}(\text{mgm-2}) \rangle$.

Le système multi-agent et le simulateur, implantés en Java, ont été exécutés sur un processeur

1. <https://www.openstreetmap.org>

2. La communication de véhicule à véhicule via la communication dédiée à courte portée (DSRC) offre une connectivité réseau rapide et à faible latence dans un rayon de communication allant jusqu'à 300 mètres.

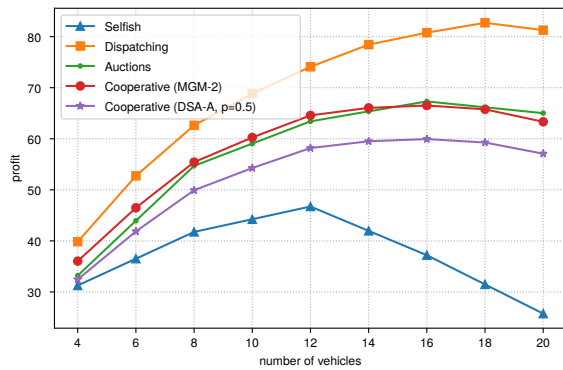


FIGURE 2 – Évolution de la QoB en fonction de la taille de la flotte de véhicules

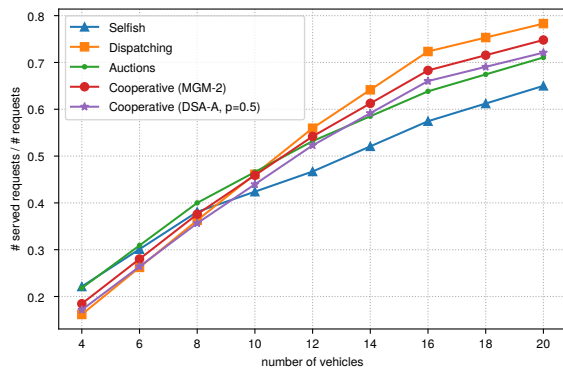


FIGURE 3 – Évolution de la QoS en fonction de la taille de la flotte de véhicules

Intel(R) Core(TM) i7-8650U à 1.90 GHz, avec 32 Go de RAM DDR4. Les algorithmes DCOPs ont été mis en œuvre en utilisant la bibliothèque FRODO [9].

Qualité des solutions. Les figures 2 et 3 illustrent les performances des cinq approches en termes d'indicateurs de qualité de service (QoS) et de qualité des affaires (QoB). Chaque point de ce diagramme représente la valeur de l'indicateur agrégée sur 1000 cycles de simulations. Ces deux figures montrent comment la qualité des solutions évolue avec l'augmentation de la taille de la flotte. Nous pouvons remarquer l'augmentation de la QoS et de la QoB avec l'augmentation du nombre de véhicules dans la flotte jusqu'à atteindre un seuil de remplissage, après lequel il n'est plus possible d'améliorer la qualité en ajoutant des véhicules supplémentaires.

Les valeurs obtenues par le comportement *avec répartiteur* représentent en quelque sorte une limite supérieure pour la QoB car le répartiteur central calcule pour chaque cas la solution localement optimale compte tenu du contexte

	max msg size	avg msg size	msg per agent	comm. load	reschedule rate
Egoïste	140	88	6	2.21 MB	2.0
Répartiteur	3500	168	21	11.2 MB	3.0
Enchères	140	112	53	37.7 MB	1.5
MGM-2	210	25	5040	297.6 MB	12.0
DSA	236	20	5015	75.1 MB	13.0

TABLE 1 – Statistiques des coûts de communication et de prise de décision pour les différents mécanismes de coordination étudiés pour un scénario avec 10 véhicules.

de l'ensemble connecté. Les performances des quatre autres approches varient selon les indicateurs. Ainsi, si l'approche basée sur les enchères est plus performante en terme de QoB que celles basées sur des algorithmes DCOP, elle est surpassée par les deux autres en termes de QoS.

L'approche *égoïste* est très efficace en termes de temps de prise de décision. La raison en est qu'elle ne nécessite pas beaucoup de calculs pour sélectionner la demande la plus proche. L'inconvénient est ici l'ignorance de ce qui précède les décisions prises et le fait que les conflits de décisions entre différents véhicules (par exemple, deux véhicules vont vouloir prendre le même passager) ne sont résolus que tardivement, ce qui réduit la QoS. Avec un faible nombre de véhicules, les ensembles connectés sont peu nombreux et, par conséquent, la quantité d'informations partagées est réduite.

Les performances des approches coopératives et de l'approche avec répartiteur dépendent fortement de la quantité d'informations, de sorte qu'il n'y a pas de différence de qualité entre les quatre approches. Avec des flottes de taille plus importante, davantage d'informations sont partagées dans les ensembles connectés. En outre, les véhicules passent plus fréquemment d'un ensemble connecté à un autre. Les trois approches coopératives ont des performances presque similaires. Pour atteindre les mêmes valeurs de QoS avec une approche égoïste, il faut davantage de véhicules dans la flotte.

Charge réseau. Le tableau 1 présente des indicateurs liés à la communication obtenus en simulant un scénario sur 1000 cycles, avec 10 véhicules, pour les différents comportements étudiés. Ici, les deuxième et troisième colonnes indiquent la taille maximale et moyenne des messages échangés (en octets) représentant l'indicateur **MsgSize**. La quatrième colonne rapporte l'indicateur **MsgCount** en termes de nombre moyen de messages reçus par un agent par cycle de simulation.

Même sans coordination, les agents échangent des messages d'information sur les nouvelles

demandes annoncées. Ce type de message dépend du mécanisme de coordination. De nouveaux types de messages sont utilisés dans le mécanisme avec répartiteur : les messages de requête et de réponse échangés entre les véhicules et le répartiteur central. Les messages de requête contiennent simplement le contexte global de l'ensemble des véhicules connectés qui demandent au répartiteur de construire leurs plannings. Les messages de réponse sont envoyés par le répartiteur aux véhicules de manière individuelle et contiennent le planning potentiel de chacun. Ces messages peuvent être volumineux, en fonction de la taille du sous-problème. Les messages d'offre et de réponse utilisés par le mécanisme de coordination basé sur les enchères sont légers, de sorte que les valeurs de l'indicateur **MsgSize** restent proches de l'approche égoïste, tandis que la valeur **MsgSize** devient polynomiale dans le nombre d'agents dans l'ensemble connecté et le nombre de leurs demandes connues. Dans les deux mécanismes de coordination basés sur un DCOP (DSA et MGM-2), les agents d'un ensemble connectéinstancient un DCOP entre eux chaque fois qu'ils doivent décider d'une mise à jour du planning. L'obtention d'une solution par l'un de ces algorithmes nécessite l'échange d'un grand nombre de messages. Ces deux algorithmes ne sont pas complets, ce qui signifie qu'ils poursuivent leurs essais pour améliorer la solution jusqu'à atteindre le *timeout* ou un optimum (local). Ceci accroît le nombre de messages échangés. D'autre part, la taille des messages échangés par ces deux approches est très faible par rapport aux autres approches.

Stabilité des plannings. Le tableau 1 présente également la fréquence des reprogrammations de plannings en considérant l'intervalle moyen entre deux cycles de simulation au cours desquels les véhicules mettent à jour leur plannings. Plus cette valeur est élevée, plus les plannings des véhicules sont stables. Dans ces contextes dynamiques, le fait d'avoir des plannings stables pendant une longue période signifie qu'aucune nouvelle demande n'est insérée, ce qui affecte la qualité de service. D'autre part, lorsque les horaires des véhicules changent fréquemment, les véhicules peuvent changer de destination et osciller pendant un certain temps avant d'effectuer un trajet réussi, ce qui peut diminuer la qualité de service. Dans nos scénarios, la coordination fondée sur des DCOPs permet d'obtenir des horaires très stables et de bonne qualité au détriment d'une charge de communication plus importante. Si la stabilité n'est pas une contrainte, mais que la communication est limitée, une ap-

proche utilisant des enchères est une très bonne alternative de stratégie d'allocation.

7 Conclusion

Ce document propose un modèle pour un problème d'allocation des ressources rencontré dans la gestion de flottes de véhicules autonomes. Notre modèle est bien adapté au domaine de l'ODT, où les flottes répondent en ligne aux demandes des passagers dans des environnements dynamiques. Notre modèle peut gérer différents types de contraintes et permet l'usage de différentes approches pour trouver des solutions et coordonner les véhicules. Nous avons mis en place un système multi-agent qui réalise ce modèle dans lequel les agents peuvent communiquer entre eux par des canaux radio en utilisant des messages pair-à-pair. Le modèle de communication prend en charge la transmission directe, par diffusion de messages et transitivité, et repose sur le concept d'ensembles connectés. Le modèle mis en œuvre offre une généralité à la fois sur les dimensions de communication et de coordination. D'une part, la portée limitée de la communication définit un attribut du problème qui affecte le niveau de connectivité et limite ainsi la centralisation réalisable. D'autre part, étant donné qu'il dépend du processus d'attribution, le choix du sous-comportement de planification des véhicules définit le mécanisme de coordination qui affecte le contexte spatio-temporel dynamique des instances du problème.

Pour expérimenter et évaluer ce modèle générique, nous avons mise en œuvre plusieurs mécanismes de coordination et les comparons brièvement suivant des indicateurs fonctionnels et techniques. Le recours aux DCOPs ou aux enchères en ligne pour coordonner les décisions décentralisées permet d'obtenir des allocations de qualité raisonnable par rapport à une allocation optimale ou à une flotte de taxis non coordonnés. Les stratégies d'allocation basées sur les DCOPs ne modifient pas trop fréquemment les plans des véhicules mais induisent tout de même plus de communication que la stratégie basée sur les enchères.

Nous prévoyons d'analyser en profondeur la relation entre la stabilité, l'exhaustivité et la faisabilité des solutions à l'avenir. Pour ce faire, nous devons mettre en œuvre d'autres approches et comparer systématiquement leurs performances, leur qualité, leur faisabilité, la stabilité et les questions techniques pour leur application pratique.

Références

- [1] Yann Chevaleyre, Paul E. Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaître, Nicolas Maudet, Julian A. Padget, Steve Phelps, Juan A. Rodr, and Paulo Sousa. Issues in Multiagent Resource Allocation. *Informatica*, 30 :37, 2006.
- [2] Alaa Daoud, Flavien Balbo, Paolo Gianessi, and Gauthier Picard. A Generic Multi-Agent Model for Resource Allocation Strategies in Online On-Demand Transport with Autonomous Vehicles. In U. Endriss, A. Nowé, F. Dignum, and A. Lomuscio, editors, *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*. International Foundation for Autonomous Agents and Multiagent Systems, 2021. Extended abstract.
- [3] Malcolm Egan and Michal Jakob. Market mechanism design for profitable on-demand transport services. *Transportation Research Part B : Methodological*, 89 :178–195, 2016.
- [4] Mohamad El Falou, Mhamed Itmi, Salah El Falou, and Alain Cardon. On demand transport system's approach as a multi-agent planning problem. In *2014 International Conference on Advanced Logistics and Transport (ICALT)*, pages 53–58, Tunis, Tunisia, 2014. IEEE, IEEE.
- [5] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. Distributed Constraint Optimization Problems and Applications : A Survey. *Journal of Artificial Intelligence Research*, 61 :623–698, March 2018.
- [6] Andrey Glaschenko, Anton Ivaschenko, George Rzevski, and Petr Skobelev. Multi-Agent Real Time Scheduling System for Taxi Companies. *AAMAS*, page 8, 2009.
- [7] Xu Jin and Luo Jie. A Study Of Multi-Agent Based Model For Urban Intelligent Transport Systems. *International Journal of Advancements in Computing Technology*, 4(6) :126–134, April 2012.
- [8] Adrien Lammoglia, Roger Marcelin Faye, and Didier Josselin. A dynamic cooperation modelling for improving taxi fleet efficiency. In *proceeding of AGILE'2012*, page 6, Avignin, 2012. Jérôme Gensel, Didier Josselin and Danny Vandenbroucke.
- [9] Thomas Léauté, Brammert Ottens, and Radoslaw Szymanek. FRODO 2.0 : An open-source framework for distributed constraint optimization. In *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*, pages 160–164, Pasadena, California, USA, July 13 2009. <https://frodo-ai.tech>.
- [10] Der-Horng Lee, Hao Wang, Ruey Cheu, and Siew Teo. Taxi Dispatch System Based on Current Demands and Real-Time Traffic Conditions. *Transportation Research Record : Journal of the Transportation Research Board*, 1882 :193–200, January 2004.
- [11] Michał Maciejewski and Kai Nagel. The influence of multi-agent cooperation on the efficiency of taxi dispatching. In *International conference on parallel processing and applied mathematics*, pages 751–760. Springer, 2013.
- [12] Ruben Van Parys, Maarten Verbandt, Marcus Kotzé, Jan Swevers, Herman Bruyninckx, Johan Philips, and Goele Pipeleers. Flexible Multi-Agent System for Distributed Coordination, Transportation & Localisation. page 3, 2018.
- [13] Jonathan P. Pearce and Milind Tambe. Quality guarantees on k-optimal solutions for distributed constraint optimization problems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07*, page 1446–1451, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
- [14] Nicole Ronald, Russell Thompson, and Stephan Winter. Simulating demand-responsive transportation : a review of agent-based approaches. *Transport Reviews*, 35(4) :404–421, 2015.
- [15] Sergey Satunin and Eduard Babkin. A multi-agent approach to Intelligent Transportation Systems modeling with combinatorial auctions. *Expert Systems with Applications*, 41(15) :6622–6633, November 2014.
- [16] Kiam Tian Seow, Nam Hai Dang, and Der-Horng Lee. A Collaborative Multiagent Taxi-Dispatch System. *IEEE Transactions on Automation Science and Engineering*, 7(3) :607–616, July 2010.
- [17] Wen Shen and Cristina Lopes. Managing Autonomous Mobility on Demand Systems for Better Passenger Experience. *arXiv :1507.02563 [cs]*, 9387 :20–35, 2015. arXiv : 1507.02563.
- [18] Yoav Shoham. *Multiagent Systems : Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.
- [19] Rinde R.S. van Lon, Tom Holvoet, Greet Vanden Berghe, Tom Wenseleers, and Juergen Branke. Evolutionary synthesis of multi-agent systems for dynamic dial-a-ride problems. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion - GECCO Companion '12*, page 331, Philadelphia, Pennsylvania, USA, 2012. ACM Press.
- [20] Li Yang, Zhao Jieru, Chen Jingxin, and Tang Zhiyong. Central Decision Intellective Taxi System and Multi Ride Algorithm. In *Proceedings of the 2017 International Conference on Artificial Intelligence, Automation and Control Technologies, AIACT '17*, pages 5 :1–5 :6, New York, NY, USA, 2017. ACM. event-place : Wuhan, China.
- [21] Mahdi Zargayouna, Flavien Balbo, and Khadim Ndiaye. Generic model for resource allocation in transportation. application to urban parking management. *Transportation Research Part C : Emerging Technologies*, 71 :538 – 554, 2016.
- [22] Weixiong Zhang, Guandong Wang, Zhao Xing, and Lars Wittenburg. Distributed stochastic search and distributed breakout : properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1) :55 – 87, 2005. Distributed Constraint Satisfaction.

Coopération généralisée et attracteurs au dilemme itéré du prisonnier

J.P. Delahaye^a P. Mathieu^a
 jean-paul.delahaye@univ-lille.fr philippe.mathieu@univ-lille.fr

^a Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRISAL (équipe SMAC)
 Centre de Recherche en Informatique Signal et Automatique de Lille,
 F-59000 Lille, France

Résumé

Pour comparer des ensembles de stratégies il existe un grand nombre de méthodes. Parmi celles-ci les modèles évolutionnaires offrent une pertinence et une robustesse remarquable. Nous étudions dans cet article deux modèles évolutionnaires parmi les plus simples et naturels possibles au dilemme itéré du prisonnier : le modèle individualiste dans lequel un individu se confronte à tout le monde et le modèle communautaire dans lequel un individu ne se confronte pas aux membres de sa propre famille. À l'aide de simulations massives utilisant des classes complètes de stratégies nous mettons en évidence des évolutions typiques. Pour le modèle individualiste, nous observons avec une grande fréquence une convergence vers un état de coopération généralisée. Pour le modèle communautaire, dont nous défendons la pertinence, nous montrons qu'il se produit de manière quasi-systématique un phénomène de convergence vers un attracteur unique et indépendant de la distribution initiale des effectifs. Des résultats statistiques sur la fréquence de ces attracteurs sont calculés et analysés.

Mots-clés : Théorie des jeux, Dilemme du prisonnier, stratégies d'agents, évolution

Abstract

To compare sets of strategies, there are a large number of different methods. Among these, evolutionary models offer a relevance and remarkable robustness. We study in this article two of the simplest and most advanced evolutionary models : the individualistic model in which an individual meets himself with everyone else and the community model in which an individual does not meet members of his own family. With the help of simulations using complete classes of strategies, we highlight two typical evolutions. For the individualistic model, we observe with great frequency a convergence towards a state of widespread cooperation. For the community model, for which we defend rele-

vance, we highlight a convergence phenomenon towards a unique attractor and independent of the initial distribution of population. Statistical results on the frequency of these attractors are calculated and analyzed.

Keywords: Game Theory, Iterated prisoner's Dilemma, Agent's Strategy, Behaviour

1 Introduction

La théorie des jeux computationnelle ou théorie algorithmique des jeux est une discipline qui permet d'étudier formellement le comportement d'agents qui entrent en interaction. Contrairement aux applications SMA pratiques, il est dans ce cadre formel, beaucoup plus facile de concevoir des stratégies comportementales et de fournir des outils permettant de les évaluer. Il s'agit donc d'un domaine de recherche très utile pour la communauté SMA puisqu'il permet de concevoir des outils adaptables aux situations pratiques. Dans ce cadre, le dilemme itéré du prisonnier (IPD) fait référence. Depuis sa description par Axelrod et Hamilton en 1981 [5] ce modèle a fait l'objet d'un grand nombre d'études et de travaux de recherche ([2, 3, 4, 7, 13, 14, 18, 22, 25, 30]). Un renouveau des recherches s'est produit à partir de l'article de Press et Dyson de 2012 [29] qui a provoqué une large discussion ([1, 15, 16, 19, 23, 31, 32], etc).

Plusieurs stratégies de jeu ont été identifiées comme réussissant de manière robuste dans des rencontres ou s'affrontent une multitude de stratégies. Les principales stratégies connues sont *tft*, *spiteful*, *pavlov*, *gradual* ([5, 6, 14, 18, 21, 22, 25]). D'autres stratégies ont été mises en évidence plus récemment [14, 29, 21, 22].

Le problème principal pour l'évaluation et la comparaison des stratégies est celui du mode de confrontation utilisé. Le tournoi fournit des résultats dont la pertinence a été discutée dès les premiers travaux de Axelrod car il donne

un poids équivalent à toutes les stratégies mises dans l'arène initiale de jeu, y compris à celles peu robustes qui disparaissent dès qu'on envisage des opérations de sélection naturelle. Bien plus intéressantes sont les compétitions évolutives, parfois appelées compétitions écologiques ([5, 3, 4, 6, 8, 9, 33, 18, 20, 22, 25, 26]), où les effectifs initiaux des stratégies mises dans l'arène évoluent de génération en génération en fonction de leur réussite dans des tournois instantanés (complets ou partiels, aléatoires ou non, soumis à du bruit ou non, avec possibilité de mutations ou non) qui se déroulent de manière répétée faisant émerger les stratégies les plus robustes et éliminant à la fois les stratégies faibles et celles qui les exploitaient.

Cette "compétition écologique" qui est le modèle de la compétition évolutive le plus simple, est utilisé dès les premiers travaux sur le dilemme itéré du prisonnier par Axelrod et Hamilton [5]. D'une génération à la suivante les résultats d'un tournoi général entre tous les individus présents dans l'arène déterminent les effectifs de la génération suivante. Sa version de base est déterministe, sans bruit, sans localisation dans un espace géométrique, et sans mutation. C'est le modèle évolutif le plus fondamental. Nous l'appellerons le "modèle évolutif individualiste" car chaque stratégie est évaluée à la génération n en fonction de son propre comportement face à tous les individus de l'arène, y compris ceux appartenant à la même catégorie que lui (c'est-à-dire jouant la même stratégie). Le second modèle évolutif fondamental est lui aussi déterministe, sans bruit, sans localisation et sans mutation. Il fonctionne comme le premier mais cette fois en supposant qu'un individu appartenant à une communauté — par exemple celle des individus jouant τft — a une attitude neutre vis à vis des individus de sa communauté et est donc évalué en considérant un tournoi dans l'arène de la génération n où il ne les rencontre pas. L'étude et la comparaison de ces deux modèles évolutifs fondamentaux semblent n'avoir jamais été réalisées dans la littérature. C'est le sujet de cet article. Cette étude nous semble importante car il est nécessaire d'avoir une compréhension aussi fine et complète que possible de ces modèles basiques avant d'envisager des modèles évolutifs plus complexes avec bruits, avec localisation topologique des individus (sur des graphes, le plan ou l'espace, etc.), avec mutations, etc. ([4, 33, 18] etc.). Les deux modèles évolutifs fondamentaux retenus engendrent des dynamiques sensiblement dif-

férentes, ce qui n'avait pas été noté jusqu'à présent. Le classique phénomène de convergence vers un état stable de coopération généralisée ([3, 7, 8, 9, 10, 20, 21, 22]) pour le modèle évolutif individualiste ("écologique") est étudié section 2. L'élimination presque systématique des stratégies agressives (prenant l'initiative de trahir) est intéressante à remarquer mais n'est pas satisfaisante pour la modélisation des écosystèmes réels, où on observe au contraire la survie à l'équilibre d'un grand nombre d'organismes variés dont certains sont agressifs. Nous fournissons des précisions statistiques sur les dynamiques qu'engendre ce modèle. L'étude menée section 3, du modèle évolutif communautaire montre qu'on n'y observe beaucoup plus rarement la convergence vers un état général de coopération. En revanche la compétition évolutive communautaire engendre fréquemment des dynamiques d'attracteurs qui ne sont que rarement observées dans le cas individualiste. Nous étudions ce modèle à nouveau en menant des expériences massives. Ce phénomène est remarquable et a été sous-estimé alors qu'il correspond à une modélisation des mécanismes évolutifs parfaitement naturelle et que ces attracteurs sont des points d'équilibre autorisant la survie d'une grande variété de stratégies, y compris agressives. La diversité des stratégies présentes dans ces points attracteurs correspond mieux à la grande variété d'organismes vivants présents simultanément dans les systèmes écologiques réels. Le modèle communautaire apparaît donc comme un intéressant outil de modélisation, et il est plus fidèle à la réalité du monde vivant que le modèle individualiste.

2 Le modèle individualiste

2.1 Les limites des classements par tournoi

Dans le dilemme itéré du prisonnier (voir [5, 4]) une infinité de stratégies est possible et aucune stratégie n'est meilleure que toutes les autres dans l'absolu. Certaines stratégies, les ALLD ou les Extorqueurs [29, 12] notamment, ne perdent jamais contre personne, mais cela a pour conséquence qu'elles gagnent peu de points et se comportent donc très mal dans une compétition où se rencontrent de nombreuses stratégies.

L'évaluation d'un ensemble de stratégies mises en compétition les unes avec les autres pour déterminer un classement entre elles est un problème délicat. Même s'il est évident que le clas-

sement trouvé dans un ensemble donné de stratégies est toujours dépendant de cet ensemble, il est essentiel pour un ensemble fixé S de disposer de moyens de comparaison réalistes et robustes aboutissant à des classements reproductibles. La méthode du tournoi (chaque stratégie rencontre toutes les autres, et le cumul des points qu'elle obtient fixe sa valeur et sert de base au classement) est la première à laquelle on pense. Elle possède un défaut grave. Si une stratégie s exploite certaines stratégies au comportement déraisonnable — par exemple celles non réactives, se laissant systématiquement exploiter comme la stratégie `allC` — et que s est bien classée dans un tournoi, ce bon résultat est illusoire, car dans un monde plus réaliste où les comportements déraisonnables disparaissent, s serait moins bien classée et peut-être très mal. C'est pourquoi Axelrod a envisagé une méthode de classement simulant la sélection naturelle.

2.2 Mise en équations vectorielles

Le modèle le plus simple dont il faut avoir une compréhension aussi précise que possible avant d'envisager des méthodes plus complexes est le suivant. Ce modèle a été appelé compétition écologique ([5, 3, 4, 6, 6, 8, 9, 33, 18, 20, 22, 24, 26]), certains auteurs parlent de PFR pour "proportional fitness rule" ([8, 9, 17, 27, 28]). Nous utiliserons le terme de "compétition évolutionnaire déterministe individualiste" pour le désigner, ou plus simplement de "modèle individualiste". L'adjectif individualiste sera opposé à communautaire (voir section 3) car dans ce modèle chaque individu joue pour lui-même et gagne des points en rencontrant tous les individus présents dans l'arène de jeu, alors que dans le modèle communautaire un individu ne gagne de points que face aux individus n'appartenant pas à sa communauté.

Pour classer les stratégies d'un ensemble E de stratégies, dans le modèle individualiste on procède en faisant évoluer une arène d'individus, chacun jouant une stratégie de l'ensemble E et rencontrant tous les autres individus présents de l'arène.

- a) Pour chaque stratégie S de E , on place dans l'arène plusieurs individus jouant S , ce qui constitue la génération 0. Les effectifs d'individus associés à chaque stratégie en ce point de départ peuvent être choisis égaux, mais il est intéressant d'envisager des effectifs inégaux afin de comparer les dynamiques qui en résultent ;

- b) On calcule le résultat d'un tournoi entre tous les individus de cette génération 0.
- c) Pour chaque stratégie S de E , on évalue le total des points gagnés par les individus jouant S . Ce total est le score de S . Les stratégies qui obtiennent les meilleurs scores auront dans la composition de la génération 1 des effectifs d'individus plus importants que celles ayant obtenu de moins bons scores.
- d) Le calcul de la génération 2 se fera à partir de la génération 1, selon la même idée. Etc.

La façon la plus simple et la plus naturelle de fixer précisément les effectifs de la génération $k + 1$ à partir de ceux de la génération k est de les prendre proportionnels pour la stratégie S aux nombres de points obtenus par les individus jouant S dans le tournoi entre individus de la génération k . L'idée est que la descendance d'un individu est directement proportionnelle pour la génération $k + 1$ au nombre de points qu'il tire de ses interactions avec les autres individus présents dans l'arène à la génération k . Pour rendre plus facile la comparaison des résultats d'une génération à la suivante, on suppose que l'effectif total reste constant. Si ce n'était pas le cas, en multipliant tous les effectifs par une constante à chaque génération on pourrait se ramener à ce cas. Le choix de fixer un même effectif total à chaque génération ne fait donc perdre aucune généralité au modèle.

Dans les simulations numériques¹, on peut mener les calculs avec des nombres réels indiquant par exemple pour chaque stratégie S le pourcentage d'individus jouant S dans l'arène. Écrivons les équations en utilisant un système de notation vectoriel et matriciel.

Soit P_k le vecteur donnant les effectifs des n stratégies qu'on cherche à classer pour la génération k : $P_k(s_i) =$ nombre d'individus jouant la stratégie s_i dans la génération k . Soit $Scores$ la matrice $n \times n$ des scores des rencontres deux à deux : $Scores(s_i, s_j) =$ score obtenu par la stratégie s_i quand elle rencontre s_j .

Le vecteur $fitInd$ de taille n indiquant le nombre de points gagnés par un individu jouant la stratégie s_i lors d'un tournoi entre les individus de la génération k est donné par le produit de la matrice $Scores(s_i, s_j)$ par le vecteur P_k :

$$fitInd_k = Scores.P_k$$

1. <https://github.com/cristal-smac/ipd>

où la notation $M.V$ désigne le produit usuel d'une matrice M par un vecteur V . Dans ce calcul des points gagnés pour le tournoi de la génération k , un individu joue contre lui-même. Si on mène le calcul avec un vecteur P_k qui correspond à des effectifs d'individus (et non pas à des pourcentages) on peut éviter cela en soustrayant à chaque composante du vecteur $fitInd_k$ la quantité $Scores(s_i, s_i)$. Lorsque nous prendrons en compte cette remarque nous parlerons de modèle strict. Le vecteur $fitStr_k$ de taille n indiquant pour chaque stratégie le nombre de points gagnés par l'ensemble des individus jouant la stratégie s_i est donné par la formule :

$$fitStr_k = P_k \bullet fitInd_k$$

où \bullet désigne le produit terme à terme entre deux vecteurs de taille n . Si on considère que le total des effectifs reste constant et égal au nombre $base$ (si on calcule avec des pourcentages on a $base = 1$), alors les nouveaux effectifs sont donnés par l'équation :

$$P_{k+1} = (base/\Sigma_k(fitStr_k)) * fitStr_k$$

où $\Sigma_k(fitStr_k)$ est la somme des composantes du vecteur $fitStr_k$ et où $r * V$ désigne le vecteur V dont chaque composante a été multipliée par le nombre réel r . Tous les calculs permettant de passer du vecteur des effectifs à la génération k au vecteur des effectifs à la génération $k + 1$ se résument donc en :

$$\begin{aligned} fitInd_k &= Scores.P_k \\ fitStr_k &= P_k \bullet fitInd_k \\ P_{k+1} &= (base/\Sigma_k(fitStr_k)) * fitStr_k \end{aligned}$$

Si le vecteur P_k est exprimé en pourcentage, on fera les calculs avec des nombres réels. Si P_k est exprimé avec des entiers et qu'on souhaite à chaque étape avoir un vecteur d'entiers, on prendra en compte la remarque pour éviter de faire jouer une stratégie contre elle-même dans le tournoi opéré à la génération k (modèle strict). Dans le cas de calculs avec des entiers, l'opération de normalisation pour maintenir constant le nombre total des individus présents à chaque génération devra être suivie d'un calcul d'arrondi (par exemple à l'entier inférieur) ce qui aura pour effet que la somme des effectifs pourra parfois être légèrement et temporairement inférieure à $base$. Ces variations du total ne changent pas fondamentalement les dynamiques observées, sauf sur un point : le modèle avec des nombres entiers arrive plus rapidement un état stable que le modèle avec des

nombres réels. Nous avons mené les deux types de calcul pour nous assurer de la cohérence et de l'universalité des résultats observés.

2.3 Convergence vers la coopération généralisée

Nous appellerons stratégies bienveillantes les stratégies qui lors d'une rencontre ne jouent jamais d'en premier, et agressives les stratégies qui, à l'opposé, prennent l'initiative de trahir même si l'adversaire a toujours joué c dans le passé. Un phénomène remarquable noté par Robert Axelrod dès ses premières simulations évolutionnaires utilisant le modèle individualiste (pour lui : *ecological evolution*) est que les stratégies agressives sont toutes repoussées à la fin des classements, y compris celles qui avaient d'assez bons résultats au tournoi. Le plus souvent, elles sont même totalement éliminées ce qui conduit donc à un ensemble de stratégies qui coopèrent toutes les unes avec les autres. Un tel état de l'arène de jeu est nommé état de "coopération généralisée". Il a été souvent mentionné dans la littérature en particulier par [5, 3, 6, 7, 8, 9, 20, 21, 22] mais aucune étude générale ne semble avoir été menée à son sujet. Le phénomène n'est pas absolument général et on trouve facilement des exceptions (voir plus loin). On ne peut donc pas énoncer et démontrer un théorème qui établirait qu'il se produit systématiquement. En revanche, on peut mener des expériences pour en évaluer la fréquence. C'est ce que nous avons fait à l'aide de plusieurs calculs massifs que nous décrivons maintenant. Partant d'une famille de stratégies F , mille fois de suite, nous prenons k stratégies au hasard dans F avec $k = 4, 5, \dots, 10$. Pour la génération 0 l'arène comporte 100 individus jouant chacune des k stratégies. La simulation évolutionnaire individualiste est effectuée (calcul avec des entiers, modèle strict, rencontres de 100 coups, points 0,3,1,5 pour la matrice des scores [3, 12]). Le nombre de simulations évolutionnaires individualistes menant à la coopération généralisée est évalué, ce qui donne un résultat entre 0 et 1000 que nous convertissons en pourcentage. Les familles F que nous avons considérées sont les suivantes :

- La famille BASIC de 20 stratégies simples et classiques (même famille que dans [12, 11] sauf la stratégie mem2 qui est sujette à des interprétations contradictoires.)
- Le même ensemble BASIC auquel sont ajoutées 5 stratégies probabilistes de type

égalisateur et extorqueur selon les définitions de [29]. Cet ensemble est noté BASIC+.

- Les ensembles $\text{Mem}(1, 1)$, $\text{Mem}(1, 2)$ et $\text{Mem}(2, 1)$ issus de [7] et comportant respectivement 32, 1024 et 1024 stratégies.

Par définition chaque stratégie de $\text{Mem}(n, m)$ mémorise les n derniers coups qu'elle a joués et les m derniers coups joués par l'adversaire et prend une décision — toujours la même — en fonction de cette information. Nous avons aussi repris la même expérience pour $\text{Mem}(1, 1)$, $\text{Mem}(1, 2)$, $\text{Mem}(2, 1)$ en ajoutant à chaque sous-ensemble de k stratégies les trois stratégies *tft*, *spiteful*, et *gradual* dont on sait qu'elles favorisent la coopération. Nous avons noté $\text{Mem}11+$, $\text{Mem}12+$, $\text{Mem}21+$ les lignes donnant les résultats pour ces familles. Au total nous avons donc calculé 56000 simulations évolutionnaires individualistes.

k	4	5	6	7	8	9	10	Avg
BASIC	67.0	69.0	74.5	76.9	78.9	84.2	83.1	76.2
BASIC+	67.4	71.9	74.2	72.7	76.3	81.7	82.9	75.3
Mem11	25.9	29.1	33.7	39.6	43.9	52.0	53.7	39.7
Mem12	14.9	15.8	14.8	16.2	15.5	16.5	18.0	16.0
Mem21	8.8	7.9	9.7	7.8	8.4	10.0	9.5	8.9
Mem11+	100	100	100	100	100	100	100	100
Mem12+	99.6	99.6	99.9	99.6	99.2	99.0	99.2	99.4
Mem21+	99.7	100.0	99.9	99.4	99.7	99.7	99.3	99.7
Avg	60.4	61.7	63.3	64	65.2	67.9	68.2	64.4

TABLE 1 – Probabilité de convergence vers la coopération généralisée lors d'une simulation évolutionnaire individualiste.

Les points suivants apparaissent.

- La convergence vers la coopération généralisée est un phénomène général qui se produit fréquemment dans une grande variété de contextes.
- Plus grand est le nombre de stratégies mises en compétition plus la probabilité de converger vers la coopération généralisée est grande.
- Globalement, la convergence vers la coopération généralisée se produit dans plus de 64,4% des cas.
- Pour les stratégies des classes complètes $\text{Mem}(1, 1)$, $\text{Mem}(1, 2)$, $\text{Mem}(2, 1)$ la coopération généralisée est fréquente mais plus rare pour les classes de stratégies plus complexes. L'explication provient en partie de ce que la proportion de stratégies agressives augmente de $\text{Mem}(1, 1)$ où elle est de 75% à $\text{Mem}(1, 2)$ et $\text{Mem}(2, 1)$ où elle est de 87,5%. Il est en effet évident que

si dans l'ensemble des k stratégies choisies au départ, il n'y a que des stratégies agressives, alors il est impossible que se produise une convergence vers la coopération généralisée.

- Les trois dernières lignes de la table 1 montrent que l'ajout de trois stratégies bienveillantes et réactives (*tft*, *spiteful*, *gradual*) provoque avec une forte probabilité une situation où la coopération généralisée s'imposera, et cela même quand c'était une éventualité rare avant l'ajout. Il est remarquable que l'ajout des trois stratégies *tft*, *spiteful* et *gradual* à des ensembles pris au hasard de 10 stratégies de $\text{Mem}(2, 1)$ fait passer la probabilité de convergence vers la coopération généralisée de 9,5% à 99,3% (avant-dernière colonne des lignes 5 et 8 de la table 1). Les stratégies bienveillantes et réactives sont des catalyseurs efficaces de pacification.

La figure 1 montre un exemple typique de convergence vers la coopération généralisée. Les stratégies prises pour cet exemple sont *prober*, *gradual*, *tft*, *mistrust*, *allC* et *allD*. Les points de départ sont : effectif 1000 pour une stratégie et 20 pour les autres (6 trajectoires), puis 1000 pour une stratégie et 50 pour les autres (6 trajectoires), puis 1000 pour une stratégie et 100 pour les autres (6 trajectoires). Cela donne 18 trajectoires évolutives correspondant à 18 distributions initiales différentes des effectifs des 6 stratégies. Les 18 trajectoires sont dessinées sur la même figure par la méthode barycentrique dont voici le principe. Pour k stratégies, nous plaçons k points sur un cercle aux sommets d'un polygone régulier à k sommets. Chaque sommet représente une stratégie. La répartition des effectifs d'une génération détermine un point à l'intérieur du polygone qui est le barycentre des k sommets affectés des effectifs respectifs des stratégies. En joignant les points successifs obtenus lors d'une simulation évolutionnaire, la variation des effectifs de l'arène produit une courbe à l'intérieur du polygone. Il s'agit d'une projection en dimension 2 d'une courbe dans un espace de dimension k . S'il y a stabilisation la courbe s'arrête en un point intérieur au polygone ou à sa frontière, qui est marqué par un rond. Dans un tel schéma, on peut dessiner plusieurs trajectoires évolutives correspondant à plusieurs distributions initiales différentes des effectifs. On voit alors en observant les trajectoires dessinées et leurs points d'arrivée, si l'état stabilisé est

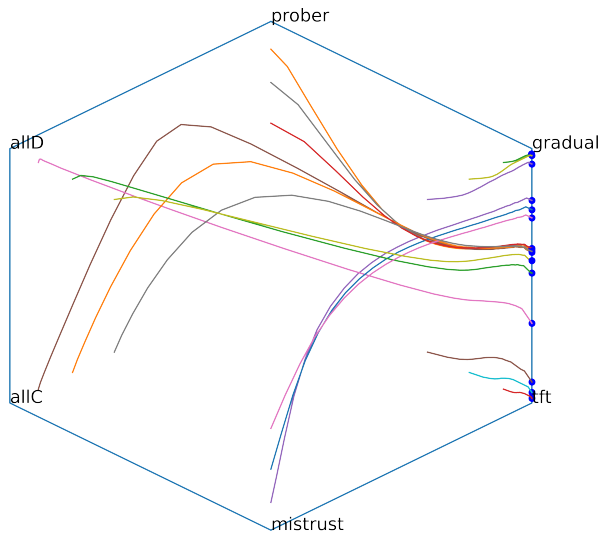


FIGURE 1 – Convergence vers la coopération généralisée. Dix-huit trajectoires évolutives du modèle individualiste conduisant à dix-huit états de coopération généralisée presque tous différents.

unique, ou s’il y en a plusieurs. La position des points d’arrêt des courbes donne des informations sur les états stables finaux. À la figure 1 on observe que toutes les trajectoires évolutives aboutissent sur le segment reliant `tft` et `gradual`, ce qui signifie qu’il ne reste dans l’état final stabilisé qu’un mélange de ces deux stratégies dans des proportions dépendant de la composition initiale de l’arène. À chaque fois il s’agit d’un état de coopération généralisée. Ce type de situations est fréquent comme le montre la table 1. Il est remarquable que même quand on part d’un point comme celui tout près de `allC` (qui correspond à une distribution des effectifs de 1000 pour `allC` et 20 pour les autres) la courbe d’évolution des effectifs se précipite sur la barre verticale à droite où `allC` a disparu et où ne subsistent que `tft` et `gradual`.

2.4 Attracteurs du modèle individualiste

Un phénomène plus rare que la convergence vers la coopération généralisée doit être remarqué : l’existence d’un état attracteur. Pour une large zone de points de départ, les trajectoires évolutives aboutissent exactement au même point d’équilibre.

Comme pour la table 1, nous avons mené des expériences massives avec une large variété des familles de stratégies pour évaluer la fréquence de ce phénomène. La table 2 donne les résultats.

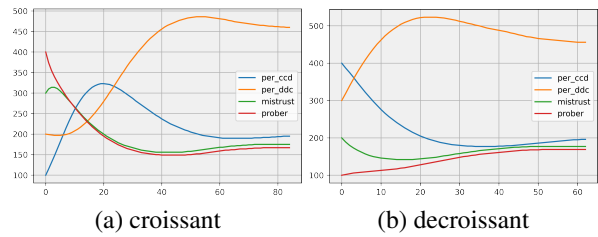


FIGURE 2 – Pour un ensemble de 4 stratégies agressives, on calcule la simulation évolutionnaire individualiste en partant de 2 répartitions initiales différentes des effectifs. Ces deux évolutions conduisent au même état stabilisé (même proportion de chacune des stratégies) qui est donc un attracteur.

tats. Précisons que nous parlons d’état attracteur dans le sens précis suivant : on opère la simulation avec des nombres réels pendant 100 itérations, en partant d’effectifs de départ de 1000 pour une stratégie et 100 pour les autres ce qui fait k points de départ différents s’il y a k stratégies. Pour chacune des stratégies, on calcule l’écart-type de l’effectif final (mesurée en pourcentage). Si cet écart-type est pour chaque stratégie inférieur à $1/1000$, on considère qu’il y a attracteur.

k	4	5	6	7	8	9	10	Avg
BASIC	4.6	1.2	0.1	0.0	0.0	0.0	0.0	0.8
BASIC+	6.9	2.4	0.7	0.2	0.1	0.0	0.0	1.5
Mem11	37.9	27.0	19.1	12.0	9.5	5.2	4.6	16.5
Mem12	39.6	27.9	19.9	13.8	12.6	9.2	6.4	18.5
Mem21	39.7	27.6	23.3	20.0	15.5	12.1	8.8	21.0
Avg	25.7	17.2	12.6	9.2	7.5	5.3	4.0	11.7

TABLE 2 – Probabilité d’existence d’un attracteur pour le modèle individualiste.

On note les points suivants.

- Avec les stratégies les plus simples et les plus performantes mises dans BASIC et BASIC+ la présence d’un attracteur est assez rare et décroissante quand on considère des familles de stratégies de taille croissante.
- La présence d’attracteurs devient plus importante quand la proportion de stratégies agressives devient importante et quand un faible nombre de stratégies est mis dans l’arène.

Un exemple de cette situation est donné aux figures 2 et 3. Cette fois toutes les trajectoires aboutissent au même point qui est donc un attracteur pour un large bassin d’attraction. Le

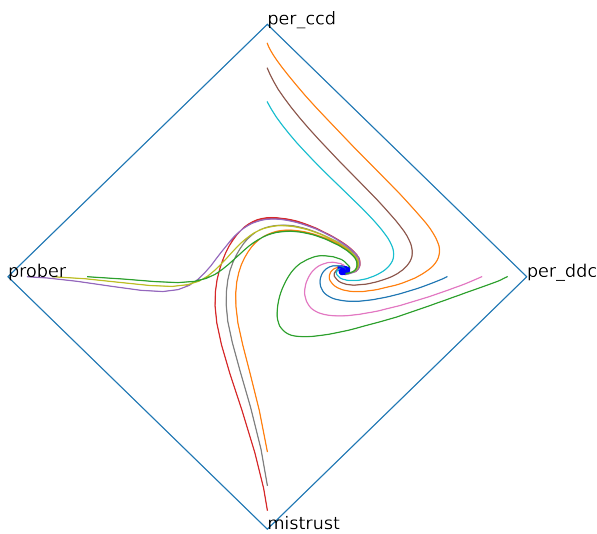


FIGURE 3 – Nous avons repris les 4 stratégies de la figure 2 et nous avons dessiné par la méthode de représentation barycentrique 12 trajectoires évolutives de la simulation évolutionnaire individualiste selon un schéma analogue à celui utilisé pour la figure 1.

point attracteur n’est pas un point de coopération généralisé mais un point où s’établit un équilibre entre chacune des quatre stratégies.

2.5 Attracteurs et coopération généralisée pour le modèle individualiste

Dans certains cas l’attracteur est dû à ce qu’on a aussi la convergence vers la coopération généralisée avec une seule stratégie bienveillante qui finit par occuper entièrement à elle seule toute l’arène. La table 3 donne la proportion de ces cas assez rares. La table a été calculée en opérant mille simulations évolutionnaires individualistes pour chaque case. Le critère utilisé pour savoir s’il y a coopération généralisée est le même que celui utilisé pour la table 1, et le critère utilisé pour savoir s’il y a un attracteur est le même que celui utilisé pour la table 2. La figure 4 fournit un exemple de cette situation. Les 5 points de départ sont les distributions d’effectifs avec 1000 individus pour une stratégie et 10 pour les autres. Les stratégies sont allC, allD, prober, mistrust et softmajo. Deux d’entre elles sont bienveillantes mais allC ne survit cependant jamais dans l’état final de coopération généralisée où seule softmajo occupe la totalité de l’arène. Le calcul a été fait avec des nombres réels, mais avec des entiers le résultat est identique. Bien noter ici encore la force de l’attrac-

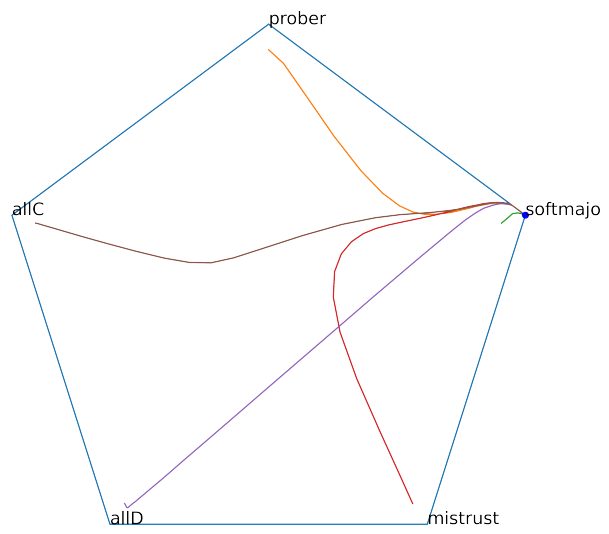


FIGURE 4 – Attracteur et coopération généralisée. Représentation barycentrique d’une situation où il y a convergence vers la coopération généralisée et en même temps attracteur.

teur : le point initial de la courbe qui part du haut est une répartition des effectifs de 1000 pour prober et 10 pour les autres. Malgré la surabondance des représentants de prober, tous disparaissent pour être remplacés uniquement par des softmajo.

k	4	5	6	7	8	9	10	Avg
BASIC	2.33	0.53	0.13	0.05	0.0	0.0	0.0	0.4
BASIC+	5.1	1.4	0.9	0.0	0.0	0.0	0.0	1.1
Mem11	8.3	8.0	5.8	4.9	4.0	3.1	2.2	5.2
Mem12	5.6	4.3	3.3	2.9	2.3	1.3	1.2	3.0
Mem21	3.7	3.2	1.9	1.9	1.7	1.7	1.4	2.0
Avg	5.0	3.5	2.4	2.0	1.6	1.2	1.0	2.4

TABLE 3 – Probabilité pour le modèle individualiste de l’existence d’un attracteur avec coopération généralisée.

2.6 Non convergence et oscillations

Pour certaines distributions initiales des stratégies dans l’arène la dynamique évolutive est chaotique avant d’aboutir à un schéma périodique. Les figures 5 et 6 donnent un exemple de cette situation délicate où le calcul avec des entiers produit une dynamique non convergente périodique alors que les calculs avec des réels donnent une dynamique d’oscillations amorties. Le calcul avec des entiers conduit à une situation périodique qui comme on le voit dépend des effectifs initiaux pour la forme exacte des oscillations, mais ne dépend pas des effectifs initiaux pour le niveau moyen auquel arrivent chaque effectif de stratégie. Le même cal-

cul avec des nombres réels conduit à des oscillations amorties où on retrouve les niveaux moyens des cas périodiques. On peut parler d'un attracteur qui ne se manifeste pleinement qu'avec le calcul avec des nombres réels, et que les problèmes d'arrondi dans le cas des calculs avec des nombres entiers transforme en ce qu'on pourrait appeler un attracteur périodique.

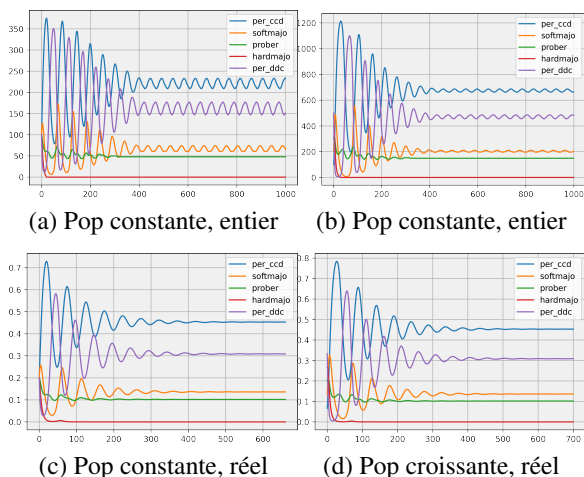


FIGURE 5 – Le modèle individualiste donne un état limite différent selon qu'on opère le calcul avec des entiers ou des réels.

3 Le modèle communautaire

3.1 Communautés solidaires

Le modèle déterministe évolutionnaire communautaire (ou modèle communautaire) prend en compte l'idée que les individus qui dans l'arène de rencontre jouent la même stratégie peuvent être considérés comme appartenant à un même groupe (ou communauté) et qu'en conséquence ils ne s'affrontent pas entre eux. Chaque membre de la communauté des individus qui par exemple jouent tft rencontrera tous les individus jouant d'autres stratégies (c'est-à-dire appartenant à d'autres communautés) mais ne rencontrera pas les individus jouant tft .

Pour les équations décrivant le modèle, le calcul de l'évolution se fait avec les mêmes méthodes que précédemment sauf qu'au lieu de prendre la matrice des rencontres $Scores$ comme base des calculs, on procède à la mise à 0 de la diagonale, ce qui a pour effet d'annuler les gains en points qu'un individu jouant une stratégie s obtient en rencontrant les membres de sa commu-

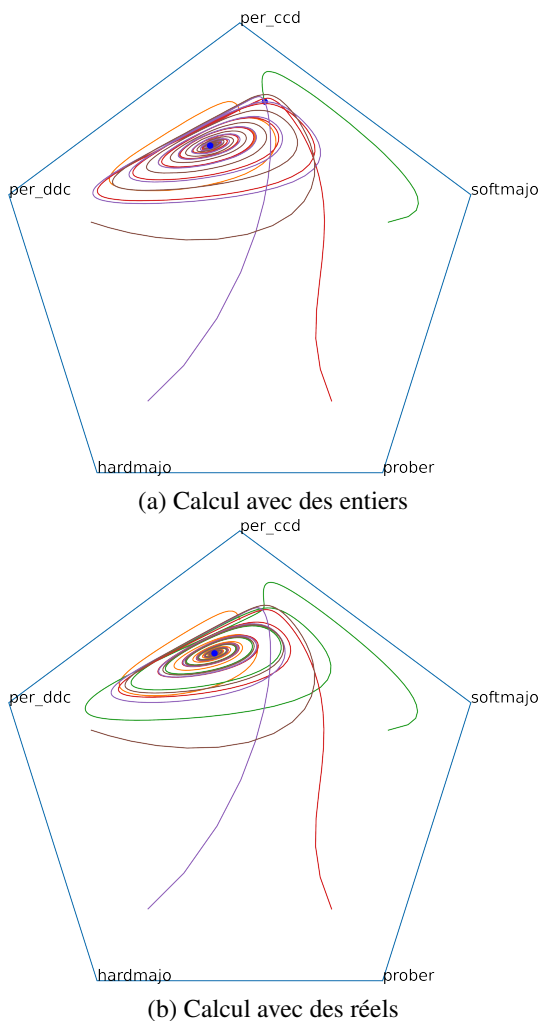


FIGURE 6 – Les mêmes stratégies que dans la figure précédente : sur la représentation barycentrique la différence entre le calcul avec des entiers et avec des réels est encore plus nette.

nauté (les autres individus jouant s). Cette annulation de la diagonale revient au même que de ne pas faire se rencontrer entre eux les membres d'une même communauté. Comme précédemment le calcul peut se faire avec des nombres réels ou avec des nombres entiers.

3.2 Justification du modèle communautaire

Le modèle communautaire représente d'une façon plus satisfaisante les situations rencontrées dans le monde vivant que le modèle individualiste. En effet, dans le monde vivant les individus d'une même espèce savent se reconnaître et composer des groupes solidaires, et c'est bien évidemment encore plus vrai des membres d'un même troupeau, d'une même famille, etc. Dans

le monde des sociétés humaines, le même type de remarques s’applique. Les individus humains s’opposent et sont en compétition entre eux, mais les membres d’une même communauté (tribu, armée, nation, etc.) sont solidaires et ne combattent que les individus des autres communautés. Les dynamiques observées avec les simulations évolutives communautaires sont assez différentes de celles observées dans le cas individualiste. On rencontre beaucoup moins souvent la convergence vers la coopération généralisée (voir le table 4), et bien plus fréquemment la présence d’un attracteur avec un vaste bassin d’attraction (voir la table 5).

Alors que dans le cas individualiste on a une convergence vers la coopération généralisée dans plus de 76% des cas sur les 7000 simulations effectuées avec BASIC et de 64% pour les 56.000 simulations effectuées au total, ces pourcentages sont maintenant seulement de 22% et de 21.3%. L’inverse se produit concernant l’existence d’attracteurs. Alors que dans le cas individualiste on ne trouve d’attracteur que dans 0.8% des cas sur les 7000 simulations effectuées avec BASIC et dans 11.7% pour les 56000 simulations, ces pourcentages sont maintenant de 81.4% et 90.8%

k	4	5	6	7	8	9	10	Avg
BASIC	25.8	24.1	23.3	20.3	19.4	19.9	21.5	22.0
BASIC+	21.8	22.3	23.5	21.2	23.6	24.2	24.1	23.0
Mem11	1.0	1.2	0.9	0.3	0.6	1.0	0.6	0.8
Mem12	0.2	0.2	0.3	0.1	0.1	0.0	0.1	0.1
Mem21	0.3	0.1	0.0	0.3	0.1	0.4	0.1	0.2
Mem11+	56.9	52.1	50.3	46.7	43.3	42.5	40.7	47.5
Mem12+	54.4	47.9	41.5	38.3	35.7	33.4	29.6	40.1
Mem21+	52.3	45.5	38.3	36.1	30.8	28.2	25.6	36.7
Avg	26.6	24.2	22.3	20.4	19.2	18.7	17.8	21.3

TABLE 4 – Probabilité de la convergence vers la coopération généralisée pour le modèle communautaire

k	4	5	6	7	8	9	10	Avg
BASIC	99.3	98.1	92.0	86.8	76.3	62.5	55.0	81.4
BASIC+	99.4	96.2	92.2	88.4	81.8	76.2	68.4	86.1
Mem11	99.7	99.5	98.6	96.8	95.4	92.8	90.6	96.2
Mem12	99.6	98.9	96.4	96.5	92.9	88.9	82.2	93.6
Mem21	99.9	99.8	99.0	97.8	96.3	93.9	90.6	96.8
Avg	99.6	98.5	95.6	93.3	88.5	82.9	77.4	90.8

TABLE 5 – Probabilité d’existence d’un attracteur pour le modèle communautaire

3.3 Pourquoi des attracteurs dans le cas communautaire ?

Les expériences montrent qu’apparaît de manière fréquente et presque systématique un phénomène d’attracteur puissant. Nous ne sommes pas en mesure de le démontrer et ce n’est

k	4	5	6	7	8	9	10	Avg
BASIC	27.0	24.4	23.7	23.2	20.3	18.6	17.1	22.0
BASIC+	23.8	21.5	21.5	23.7	22.7	23.1	24.8	23.0
Mem11	1.3	0.5	0.7	1.0	0.7	0.5	0.8	0.8
Mem12	0.6	0.2	0.2	0.3	0.1	0.1	0.1	0.2
Mem21	0.1	0.1	0.0	0.0	0.0	0.0	0.2	0.2
Avg	10.6	9.3	9.2	9.6	8.8	8.5	8.6	9.2

TABLE 6 – Probabilité de l’existence d’un attracteur avec coopération généralisée pour le modèle communautaire.

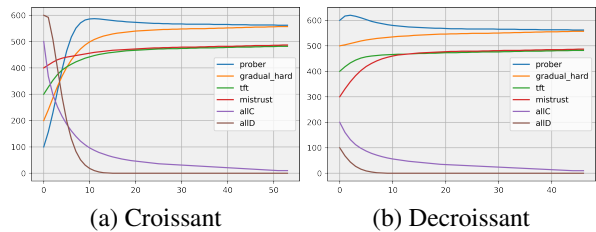


FIGURE 7 – Un exemple d’attracteur pour le cas communautaire. Les courbes et le schéma barycentrique des figures 8 et 9 montrent un cas typique de dynamique avec attracteur dans le cas de la simulation évolutive communautaire. Pour un très vaste bassin d’attraction toutes les évolutions conduisent exactement au même équilibre final.

d’ailleurs pas possible en toute généralité puisque parfois il n’y a pas d’attracteurs unique. Nous proposons cependant une explication heuristique de l’existence des attracteurs dans ce modèle. L’un des effets de la méthode de calcul des effectifs est que lorsqu’une communauté C accroit le nombre de ses individus, alors (a) la communauté C gagne plus de points, et donc maintient un effectif important, mais (b) chaque individu de la communauté C se trouve ne rencontrer qu’un nombre devenant plus faible d’individus des autres communautés car le total des effectifs est constant.

Devenir plus gros crée donc pour une communauté C un handicap, alors que devenir petit à l’inverse procure un avantage qui peut être perdu si la stratégie jouée est mauvaise ou médiocre, mais qui dans le cas d’une stratégie moyenne a pour conséquence de limiter son recul. Arrivé à un certain point de croissance l’avantage de la stratégie dont l’effectif s’accroît (car elle est performante) diminue et s’égalise avec l’avantage croissant de la communauté dont l’effectif décroît (du fait par exemple de sa médiocrité). Un point d’équilibre est donc possible et même probable ne dépendant que de la matrice Score.

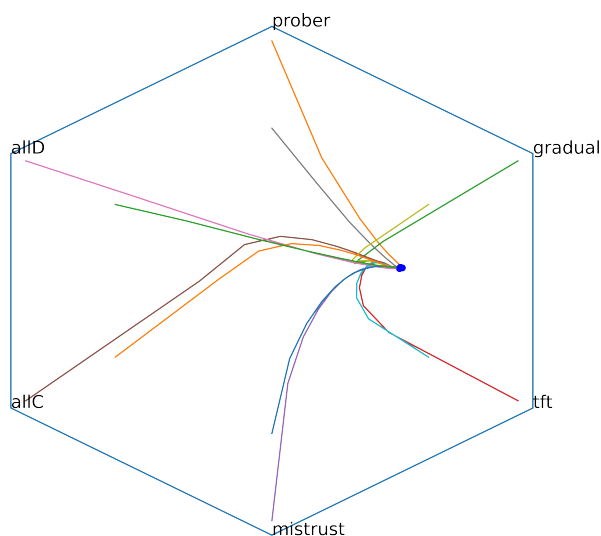


FIGURE 8 – Mise en évidence spectaculaire d'un attracteur au large bassin d'attraction.

4 Conclusion

Les simulations évolutives déterministes individualistes (où chaque individu rencontre tous les individus, même ceux jouant la même stratégie que lui) et les simulations déterministes évolutives communautaires (où les individus ne rencontrent pas les individus jouant la même stratégie qu'eux-mêmes) donnent des dynamiques sensiblement différentes.

Il y a très fréquemment convergence vers la coopération généralisée pour le modèle individualiste et nettement moins souvent pour le modèle communautaire. Le modèle individualiste est en quelque sorte intransigeant : les individus qui persistent à avoir une attitude agressive n'ont dans la majorité des cas pas leur place dans l'état final stabilisé. Le modèle communautaire permet lui plus facilement la mise en place d'états stabilisés où les groupes qui s'opposent ont les uns contre les autres des attitudes non totalement pacifiées. Le modèle communautaire apparaît donc plus proche à la fois dans sa définition et pour les états stables qu'il induit de la réalité du monde biologique où la persistance indéfinie de comportements agressifs des diverses espèces (identifiables à nos communautés) semble la règle. Cette observation peut sembler paradoxale puisque l'individualisme conduit avec une sévérité extrême à l'élimination de ceux qui n'acceptent pas de renoncer aux comportements agressifs. On peut au contraire la trouver naturelle : l'existence de communautés permet à des dynamiques plus subtiles de se mettre en place autorisant la sur-

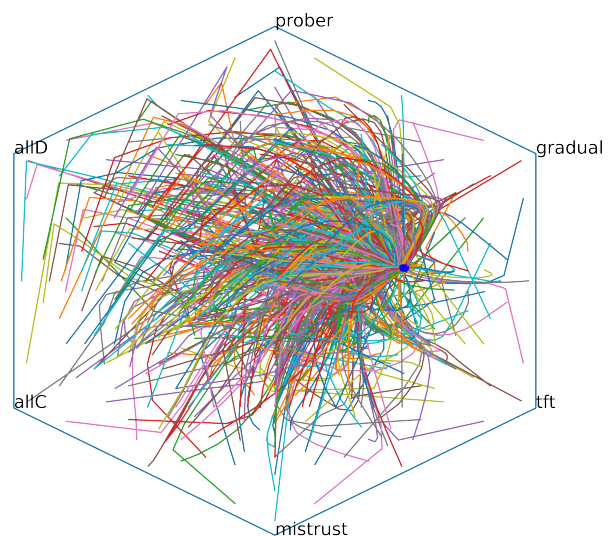


FIGURE 9 – Même ensemble de stratégies que pour la figure précédente. Cette fois nous avons dessiné 729 trajectoires qui arrivent néanmoins toutes au même point attracteur.

vie d'une variété plus grande de comportements et des équilibres plus complexes comme on les observe dans la nature.

Avec le modèle communautaire, l'existence très fréquente d'attracteurs est inattendue. Il y a comme un équilibre prédéterminé entre les différentes communautés de stratégies, et quels que soient les effectifs initiaux (ou en tout cas pour un large ensemble de distributions de départ) l'arène évolue vers cet équilibre prédéterminé. Cette observation est importante et avec la première observation sur la fréquence de la convergence vers la coopération généralisée, elle renforce l'idée que le modèle communautaire devrait être pris plus au sérieux qu'il ne l'a été jusqu'à présent. Les états stables vers lesquels tendent les simulations communautaires se reconstituent quand une force (aléatoire ou autre) les en écarte. Ces états stables sont donc bien plus résistants et robustes aux perturbations que ceux atteints par les simulations individualistes qui eux peuvent dériver si on les soumet à des perturbations mêmes légères.

Références

- [1] Christoph Adami and Arend Hintze. Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything. *Nature communications*, 4(2193), 2013.
- [2] Ethan Akin. The iterated prisoner's dilemma : good strategies and their dynamics. *Ergodic Theory, Advances in Dynamical Systems*, pages 77–107, 2016.

- [3] Robert Axelrod. *The evolution of cooperation : revised edition*. Basic books, 1984, 2006.
- [4] Robert Axelrod. *The complexity of cooperation : Agent-based models of competition and collaboration*, volume 3. Princeton University Press, 1997.
- [5] Robert Axelrod and William Donald Hamilton. The evolution of cooperation. *science*, 211(4489) :1390–1396, 1981.
- [6] Bruno Beaufils, Jean-Paul Delahaye, and Philippe Mathieu. Our meeting with gradual, a good strategy for the iterated prisoner’s dilemma. In *ALIFE’5*, pages 202–209. The MIT Press/Bradford Books, 1996.
- [7] Bruno Beaufils, Jean-Paul Delahaye, and Philippe Mathieu. Complete classes of strategies for the classical iterated prisoner’s dilemma. In *EP’7 proc.*, volume 1447 of *LNCS*, pages 33–41. Springer, 1998.
- [8] Jonathan Bendor and Piotr Swistak. Types of evolutionary stability and the problem of cooperation. *Proceedings of the National Academy of Sciences*, 92(8) :3596–3600, 1995.
- [9] Jonathan Bendor and Piotr Swistak. The evolutionary stability of cooperation. *American Political Science Review*, 91(2) :290–307, 1997.
- [10] Jean-Paul Delahaye and Philippe Mathieu. Random strategies in a two levels iterated prisoner’s dilemma : How to avoid conflicts. In *ECAI proc.*, volume 96, pages 68–72, 1996.
- [11] Jean-Paul Delahaye and Philippe Mathieu. Méta-stratégies pour le dilemme itéré du prisonnier. In *JFSMA*, pages 13–22. Cépaduès, 2016.
- [12] Jean-Paul Delahaye and Philippe Mathieu. Que valent les stratégies probabilistes au dilemme itéré des prisonniers ? In *JFSMA*, pages 41–51. Cépaduès, 2017.
- [13] Jeffrey A Fletcher and Martin Zwick. Unifying the theories of inclusive fitness and reciprocal altruism. *The American Naturalist*, 168(2) :252–262, 2006.
- [14] Nikoleta E Glynnatsi and Vincent A Knight. A meta analysis of tournaments and an evaluation of performance in the iterated prisoner’s dilemma. *arXiv preprint arXiv :2001.05911*, 2020.
- [15] Christian Hilbe, Torsten Röhl, and Manfred Milinski. Extortion subdues human players but is finally punished in the prisoner’s dilemma. *Nature communications*, 5(1) :1–6, 2014.
- [16] Christian Hilbe, Arne Traulsen, and Karl Sigmund. Partners or rivals ? strategies for the iterated prisoner’s dilemma. *Games and economic behavior*, 92 :41–52, 2015.
- [17] Luis R Izquierdo, Segismundo S Izquierdo, and J Gary Polhill. Evo-2x2 : a modelling framework to study the evolution of strategies in 2x2 symmetric games under various competing assumptions. 2008.
- [18] Graham Kendall, Xin Yao, and Siang Yew Chong. *The Iterated Prisoners’ Dilemma : 20 Years on*. World Scientific Publishing Co., Inc., 2007.
- [19] Jie Liu, Y Li, C Xu, and PM Hui. Evolutionary behavior of generalized zero-determinant strategies in iterated prisoner’s dilemma. *Physica A*, 430 :81–92, 2015.
- [20] Philippe Mathieu, Bruno Beaufils, and Jean-Paul Delahaye. Studies on dynamics in the classical iterated prisoner’s dilemma with few strategies : Is there any chaos in the pure dilemma ? In *AE proc.*, volume 1829 of *LNCS*, pages 177–190. Springer, 1999.
- [21] Philippe Mathieu and Jean-Paul Delahaye. New winning strategies for the iterated prisoner’s dilemma. In *AAMAS proc.*, pages 1665–1666, 2015.
- [22] Philippe Mathieu and Jean-Paul Delahaye. Experimental criteria to identify efficient probabilistic memory-one strategies for the iterated prisoner’s dilemma. *Simulation Modelling Practice and Theory*, 97 :101946, 2019.
- [23] Manfred Milinski, Christian Hilbe, Dirk Semmann, Ralf Sommerfeld, and Jochem Marotzke. Humans choose representatives who enforce cooperation in social dilemmas through extortion. *Nature communications*, 7(1) :1–9, 2016.
- [24] Martin Nowak and Karl Sigmund. Chaos and the evolution of cooperation. *Proceedings of the National Academy of Sciences*, 90(11) :5091–5094, 1993.
- [25] Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner’s dilemma game. *Nature*, 364(6432) :56–58, 1993.
- [26] Martin A Nowak. *Evolutionary dynamics : exploring the equations of life*. Harvard university press, 2006.
- [27] Brendan O’Connor, Noah Mark, and James Fearon. The social evolution of human cooperation via group competition and conflict. 2006.
- [28] Tadeusz PŁatkowski. Cooperation in two-person evolutionary games with complex personality profiles. *Journal of theoretical biology*, 266(4) :522–528, 2010.
- [29] William H. Press and Freeman J. Dyson. Iterated prisoner’s dilemma contains strategies that dominate any evolutionary opponent. *PNAS*, 109(26) :10409–10413, 2012.
- [30] Karl Sigmund. *The calculus of selfishness*. Princeton University Press, 2016.
- [31] Alexander J Stewart and Joshua B Plotkin. From extortion to generosity, evolution in the iterated prisoner’s dilemma. *Proceedings of the National Academy of Sciences*, 110(38) :15348–15353, 2013.
- [32] Attila Szolnoki and Matjaž Perc. Evolution of extortion in structured populations. *Physical Review E*, 89(2) :022804, 2014.
- [33] Jianzhong Wu and Robert Axelrod. How to cope with noise in the iterated prisoner’s dilemma. *Journal of Conflict resolution*, 39(1) :183–189, 1995.

État de l'art

De la diversité des jeux de coalitions à utilité transférable

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

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

Résumé

Dans cet article, nous présentons une synthèse des différents modèles de jeux de coalitions. Les jeux de coalitions ont pour objectif de partitionner les agents en groupes, appelés coalitions, pour leur permettre de coopérer. L'intérêt des extensions des jeux de coalitions classiques est alors de pouvoir modéliser des contextes divers (hétérogénéité des capacités des agents, interdépendance des coalitions, incertitude, résolution décentralisée) bien adaptés à des problématiques multi-agents. Nous proposons une classification des modèles selon trois axes ainsi qu'un tour d'horizon des protocoles de résolution décentralisés des jeux de coalitions.

Mots-clés : Coalitions, Théorie des jeux

Abstract

In this article, we present a synthesis of the different models of coalitional games. The aim of such games is to partition the agents into groups, called coalitions, in order to make them cooperate. The interest of extending the classic coalitional game is to model diverse contexts (heterogeneity of agents, interdependence of coalitions, uncertainty, decentralized resolution), which are well adapted to multi-agent contexts. We propose to classify all those models according to three axes as well as an overview of decentralized resolution protocols for coalitional games.

Keywords: Coalitions, Game Theory

1 Introduction

Dans les systèmes multi-agents composés d'agents hétérogènes et égoïstes, il est parfois nécessaire de coopérer, c'est-à-dire travailler de concert temporairement afin que les agents puissent réaliser leurs tâches. Nous pouvons penser aux réseaux électriques intelligents [12, 63], aux réseaux de capteurs [29], aux chaînes logistiques automatisées [27], ou toute autre application de systèmes cyber-physiques. Le cadre formel pour modéliser ces situations est la théorie des jeux coopératifs, et plus particulièrement

la formation de coalitions.

Dans un *jeu de coalitions*, les agents – égoïstes – d'un système sont amenés à former des groupes (que nous appelons coalitions) afin d'en tirer un gain, en sachant que leur égoïsme les rendent enclins à ne pas accepter de former n'importe quelle coalition. Deux grandes familles de jeux existent, les *jeux à utilité transférable* où les agents reçoivent une utilité quantifiée lorsqu'ils coopèrent et qu'ils doivent ensuite partager entre eux [49, 52]; et les *jeux à utilité non transférable* [49] où les agents reçoivent un gain individuel (exprimé sous forme de préférences) qu'ils ne peuvent partager. Les jeux à utilité non transférable étant un cas particulier de jeux à utilité transférable, nous nous concentrons dans cet article sur ces derniers.

L'objectif de cet article est alors de présenter une revue des différents modèles de formation de coalitions, en partant du cadre classique des jeux à utilité transférable puis en relâchant au fur et à mesure ses hypothèses les plus contraignantes. Nous n'avons pas ici la prétention d'être exhaustifs mais de mettre en lumière de la diversité des modèles et leur capacité à répondre aux besoins des systèmes multi-agents dans des contextes applicatifs. Cette diversité est représentée en figure 1. L'origine, marquée d'un point plus épais, représente les modèles classiques tandis que les axes représentent chacun le relâchement d'une hypothèse : agents homogènes (axe bleu), coalitions indépendantes (axe noir), gestion de l'incertitude (axe rouge).

Nous présentons ainsi dans la section 2 la formation de coalitions pour le jeux à utilité transférable dans sa forme classique, les concepts caractérisant les solutions acceptables par les agents, ainsi que la manière de représenter ces jeux. Dans la section 3, nous présentons des extensions du cadre classique pour modéliser les capacités hétérogènes des agents. Dans la section 4, nous nous intéressons aux cadres qui relâchent l'hypothèse d'indépendance entre les coalitions. La section 5 est consacrée aux cadres incluant

État de l'art

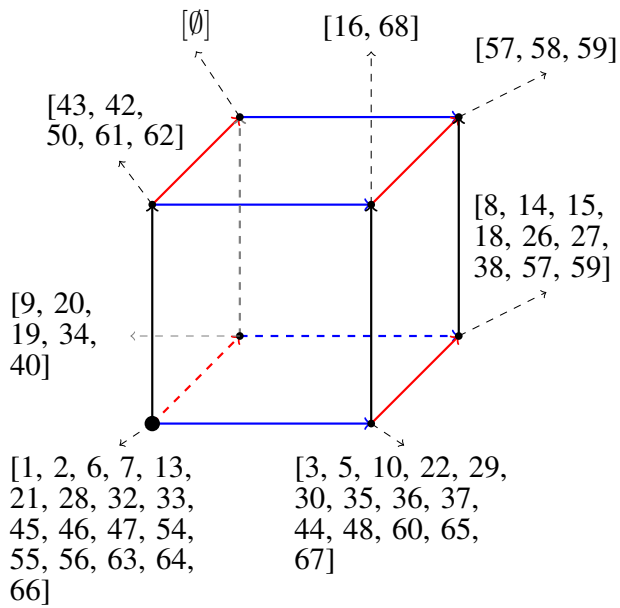


FIGURE 1 – Taxonomie des modèles : à partir du cadre classique (point noir épais), l'axe bleu indique la prise en compte des capacités des agents, l'axe noir indique la prise en compte des dépendances entre coalition et l'axe rouge indique la prise en compte de l'incertitude

de l'incertitude. Enfin, la section 6 propose une revue de protocoles de formation de coalitions permettant la décentralisation.

2 Jeux de coalitions classiques

2.1 Définition formelle

Les jeux de coalitions modélisent des situations où des agents doivent coopérer afin d'obtenir un gain supérieur à ce qu'ils gagneraient s'ils ne coopéraient pas. Lorsque ces agents coopèrent, ils forment des *coalitions*, c'est-à-dire des sous-ensembles d'agents travaillant de concert. Si l'ensemble de tous les agents du système forment une unique coalition, cette dernière est appelée la *grande coalition*. Si un agent reste seul, il forme sa *coalition singleton*.

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,
- $v : 2^N \rightarrow \mathbb{R}$ est une fonction caractéristique qui à chaque coalition associe une valeur réelle, appelée utilité de la coalition et notée $v(C)$ où $C \subseteq N$.

Dans les jeux de coalitions classiques [46, 49], les coalitions sont disjointes une à une, chaque agent ne faisant partie que d'une seule coalition. La partition de tous les agents en coalitions forme une *structure de coalitions* sur N .

Definition 2 (Structure de coalitions) Une structure de coalitions \mathcal{C} est une partition de l'ensemble des agents de N en k coalitions disjointes : $\mathcal{C} = \{C_1, \dots, C_k\}$ et $\bigcup_{C \in \mathcal{C}} C = N$.

Une fois les coalitions formées, les agents doivent se répartir l'utilité des coalitions entre les membres de celles-ci. Cette répartition, appelée *imputation*, est définie comme suit.

Definition 3 (Imputation) Une imputation dans une structure de coalitions \mathcal{C} au sein d'un jeu \mathcal{G} est un vecteur de gains tel que $\vec{x} = \{x_1, \dots, x_n\}$, où x_i est le gain de l'agent a_i , et $x_i \geq 0$.

Ainsi, un *solution* à un jeu de coalitions est une structure de coalitions associée à une imputation.

Definition 4 (Solution d'un jeu de coalitions) Une solution à un jeu de coalitions \mathcal{G} est un tuple $S_{\mathcal{G}} = \langle \mathcal{C}, \vec{x} \rangle$ où :

- \mathcal{C} est une structure de coalitions de N ,
- $\vec{x} = \{x_1, \dots, x_n\}$ est une imputation.

Comme il est fait l'hypothèse que les agents sont égoïstes, ceux-ci cherchent à obtenir un gain maximum. Cela a pour conséquence qu'une solution doit être unanimement acceptée par les agents, c'est-à-dire qu'aucun d'entre eux ne doit être capable de former ou rejoindre une autre coalition dans le but d'obtenir un gain plus important. Si une solution à un jeu de coalitions est acceptée par tous les agents, celle-ci est *stable* [49]. Ces solutions stables appartiennent à des *concepts de solutions*, qui représentent une caractérisation d'une notion de stabilité donnée. Par exemple, cela peut consister en le fait qu'aucun agent ne peut en influencer un autre contre une partie de son gain, ou bien alors qu'aucun ensemble d'agents ne peut former une autre coalition pour augmenter leurs gains globaux.

2.2 Concepts de solutions et valeurs

Les concepts de solutions définissent donc différentes notions de stabilité et d'optimalité. C'est pourquoi une solution stable dans un certain

État de l'art

concept peut ne pas l'être dans un autre. Les concepts de solutions se distinguent en deux catégories : les *concepts de solutions ensemblistes* et les *concepts de solutions singletons*.

Les concepts de solutions ensemblistes caractérisent les solutions par des contraintes représentant l'absence d'incitation des agents à dévier d'une coalition donnée. Le concept fondamental est celui de *cœur* [28, 49, 56] mais il peut être vide. C'est pourquoi la littérature propose des généralisations qui relâchent certaines contraintes pour obtenir au moins une solution, comme le *dernier cœur* [41, 45], le *nucléole* [41, 54], le *noyau* [17, 21] ou les *ensembles de marchandage* [17, 24].

Par exemple, le cœur caractérise l'ensemble des solutions où aucun groupe d'agent ne peut former une autre coalition qui lui rapporterait plus, tandis que le noyau caractérise les solutions dans lesquelles aucun agent ne peut rationnellement donner une partie de son gain à un autre pour l'inciter à rejoindre sa coalition.

Une des caractéristiques intéressantes des concepts de solutions ensemblistes est qu'il existe des relations d'inclusion entre eux. Ces relations sont présentées en figure 2. Si le cœur est non vide, alors le nucléole est dans le cœur et se trouve à l'intersection du noyau et du dernier cœur. Ces deux-ci sont à leur tour inclus dans les ensembles de marchandage [17, 24, 54].

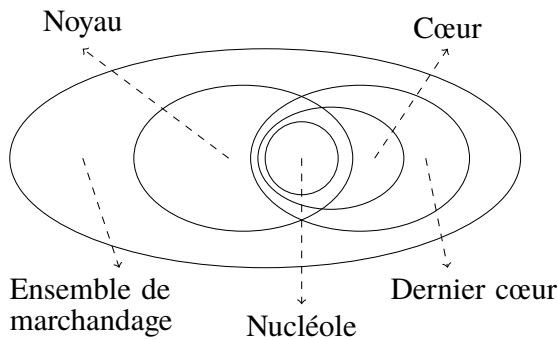


FIGURE 2 – Inclusions entre concepts

La seconde catégorie de concepts de solutions est celle des concepts de solutions singletons. Ces concepts ne produisent qu'une unique solution en définissant une règle de partage de la valeur produite. Ainsi, à une structure de coalitions donnée, il n'y a qu'une imputation possible. De manière générale, ces règles s'appuient sur la contribution marginale des agents, c'est-à-dire ce que les agents apportent en moyenne aux coalitions qu'ils forment. La valeur est ensuite

partagée équitablement, c'est-à-dire proportionnellement à la contribution marginale.

Les deux concepts singletons majeurs sont la *valeur de Shapley* [39, 55] et l'*indice de Banzhaf* [6]. La valeur de Shapley est calculée sur l'ensemble des ordres d'arrivée possibles des agents dans les coalitions tandis que l'indice de Banzhaf sur un unique ordre d'arrivée. Ainsi, l'indice de Banzhaf est moins complexe à calculer mais il perd certaines propriétés intéressantes comme l'additivité (la somme des indices de Banzhaf de deux jeux n'est pas égale à l'indice de Banzhaf de la somme de ces deux jeux).

Definition 5 (Répartitions équitables) Soit Π l'ensemble des ordres d'arrivée des agents dans les coalitions et $\pi(a_i)$ l'ensemble des agents précédant l'agent a_i dans un ordre donné. La valeur de Shapley ϕ_i et l'indice de Banzhaf ψ_i d'un agent a_i sont définis par :

$$\phi_i = \frac{1}{n!} \sum_{\pi(a_i) \in \Pi} v(\pi(a_i) \cup a_i) - v(\pi(a_i))$$

$$\psi_i = \frac{1}{2^{n-1}} \sum_{C \subseteq N \setminus a_i} v(C \cup a_i) - v(C)$$

Les concepts de solutions précédents se fondent sur l'équité entre agents, d'autres concepts singletons proposent d'introduire de l'égalité dans les règles de répartition. C'est le cas de la famille des *valeurs de solidarité* [13, 47, 66]. Alors que les répartitions équitables s'appuient sur les contributions marginales individuelles des agents, les répartitions égalitaires s'appuient sur les contributions marginales moyennes au sein d'une coalition donnée.

Definition 6 (Répartitions égalitaires) La contribution marginale moyenne d'une coalition C est donnée par :

$$\mathcal{A}(C) = \frac{1}{|C|} \sum_{k \in C} [v(C) - v(C \setminus k)]$$

La valeur de solidarité χ_i de l'agent a_i est alors :

$$\chi_i = \sum_{C \ni a_i} \frac{(n - |C|)! (|C| - 1)!}{n!} \mathcal{A}(C)$$

2.3 Représenter la fonction caractéristique

Au-delà des questions d'algorithmique pour le calcul des solutions, que nous aborderons en

État de l'art

section 6, les jeux de coalitions posent aussi des questions en termes de représentation. En effet, le nombre de coalitions possibles augmentant exponentiellement en fonction du nombre d'agents, la fonction caractéristique peut atteindre une taille conséquente. Des travaux ont donc été menés afin de proposer des représentations compactes de la fonction caractéristique.

Parmi celles-ci, nous pouvons trouver les *induced subgraph games* qui utilisent des graphes non-orientés pondérés [23, 31], les *algebraic decision diagrams* qui utilisent des arbres binaires dont les nœuds sont des agents et les feuilles des valeurs [1, 32] ou encore les *marginal contribution networks* [31, 33] qui utilisent des règles d'agrégation.

À titre d'exemple, détaillons les *marginal contribution networks*, ou *MC-nets* [31, 33], pour lesquels des algorithmes efficaces ont été proposés [11, 25, 62], comme dans le cas du calcul de la valeur de Shapley [33]. Les *MC-nets* représentent donc la fonction caractéristique par un ensemble de règles de la forme *modèle* \rightarrow *valeur*. Le modèle est une formule booléenne (sous forme de conjonction dans la majorité des cas) représentant les agents. Dans une formule, un littéral positif désigne la présence de l'agent associé, ou son absence pour les littéraux négatifs. Une coalition *satisfait* une règle si la formule booléenne est satisfaite par l'ensemble représentant la coalition, c'est-à-dire si la coalition contient tous les agents correspondants aux littéraux positifs, et ne contient pas les agents décrits par les littéraux négatifs. La valeur d'une coalition est alors égale à la somme des valeurs des règles qui satisfont cette coalition. Considérons par exemple les trois règles suivantes :

$$\begin{aligned} a_1 &\rightarrow 1 \\ a_1 \wedge a_2 &\rightarrow 2 \\ a_1 \wedge a_2 \wedge \neg a_3 &\rightarrow 3 \end{aligned}$$

Avec ces trois règles, nous pouvons déduire les valeurs des sept coalitions suivantes : $v(\{a_1\}) = 1, v(\{a_2\}) = 0, v(\{a_3\}) = 0, v(\{a_1, a_2\}) = 6, v(\{a_1, a_3\}) = 1, v(\{a_2, a_3\}) = 0, v(\{a_1, a_2, a_3\}) = 3$.

3 Jeux à capacités

Dans les jeux de coalitions classiques, les tâches à réaliser sont abstraites : elles ne sont pas représentées et n'existent qu'à travers un ensemble d'agents et l'utilité qu'ils tirent des coalitions.

Or, dans une situation plus complexe, les agents peuvent avoir des capacités spécifiques que l'on peut exprimer formellement, et ils peuvent devoir réaliser des tâches précises qui requièrent des compétences diverses. Dans le même ordre d'idée, les agents peuvent être aussi soumis à des contraintes de ressources volatiles : exécuter certaines tâches ou utiliser certaines capacités peut être coûteux pour les agents.

Enfin, les jeux de coalitions classiques ne prennent pas en compte les restrictions que les agents peuvent avoir en termes de communication ou d'accointance : toutes les agents peuvent former des coalitions avec tous les autres. Or, dans certains domaines, comme les réseaux pair-à-pair ou les chaînes logistiques [2, 59], les agents sont soumis à de telles contraintes.

Afin que les jeux de coalitions puissent modéliser des systèmes comprenant des tâches requérant des capacités particulières, qu'elles soient décrites de façon qualitatives ou quantitatives, il a été proposé des jeux à compétences [5, 14, 15, 27, 48, 65] et des jeux à ressources [3, 10, 38, 59, 65, 67] que nous regroupons sous l'appellation *jeux à capacités*. Les tâches devenant explicites, il est ajouté au jeu un ensemble de tâches, que les coalitions doivent réaliser pour en tirer un gain. La fonction caractéristique est alors non plus définie par les coalitions mais par les tâches.

Definition 7 (Jeu à capacités) *Un jeu de coalitions à capacités est un tuple $\mathcal{G} = \langle N, \mathcal{T}, v \rangle$ où :*

- $N = \{a_1, \dots, a_n\}$ est un ensemble d'agents,
- $\mathcal{T} = \{t_1, \dots, t_m\}$ est un ensemble de tâches,
- $v : 2^{\mathcal{T}} \rightarrow \mathbb{R}$ est une fonction caractéristique qui à chaque ensemble de tâches associe une valeur réelle qui est l'utilité gagnée par une coalition lorsque qu'elle les réalise.

Les agents sont quant à eux enrichis de capacités. Une capacité qualitative doit être vue comme une compétence pouvant être requise pour la réalisation d'une tâche [5]. Une capacité quantitative peut être vue comme une fiabilité de l'agent, une ressource qu'il possède et peut mettre à profit pour la réalisation d'une tâche, ou bien comme une capacité de stockage [59].

Dans le cadre qualitatif, il a été proposé les *jeux à compétences* [5, 48]. Dans ceux-ci, un ensemble de compétences $S = \{s_1, \dots, s_k\}$ est ajouté au jeu, et les agents possèdent chacun un sous-ensemble $S(a_i) \subset S$ de compétences.

État de l'art

À chaque tâche est également associé un sous-ensemble $S(t_j) \subset S$, décrivant l'ensemble des compétences requises pour réaliser cette tâche. Afin d'obtenir un gain, les agents doivent former des coalitions réunissant les compétences nécessaires à la réalisation des tâches.

Definition 8 (Pouvoir d'une coalition) *Le pouvoir d'une coalition C est défini par $S(C) = \bigcup_{a_i \in C} S(a_i)$. Une coalition C peut réaliser une tâche t_j si, et seulement si, les compétences nécessaires à la tâche sont incluses dans son pouvoir, $S(t_j) \subseteq S(C)$.*

Des sous-classes de ce modèle ont été considérées comme les *task count skill games*, où la valeur d'une coalition est égale au nombre de tâches qu'elle peut accomplir, et les *weighted task skill games* où elle est la somme de poids associés à chaque tâche [5]. Résoudre un jeu à compétences est dans la même classe de complexité qu'un jeu de coalitions classique, sauf dans le cas favorable où les tâches requièrent un nombre constant de compétences [4].

Dans le cadre quantitatif, il a été proposé des jeux à ressources comme illustré par le modèle de Shehory et Kraus [59]. Ces jeux considèrent un ensemble des capacités quantitatives $\mathcal{B} = \{b_1, \dots, b_r\}$. Les agents ainsi que les tâches sont associés à des vecteurs qui décrivent les ressources disponibles et les ressources requises : $\forall a_i \in N, \vec{b}_i = \langle b_1^i, \dots, b_r^i \rangle$ et $\forall t_j \in \mathcal{T}, \vec{b}_{t_j} = \langle b_1^{t_j}, \dots, b_r^{t_j} \rangle$.

Definition 9 (Pouvoir d'une coalition) *Le pouvoir d'une coalition C est donné par $\vec{b}_C = \sum_{a_i \in C} \vec{b}_i$. Une coalition peut réaliser une tâche t_j si, et seulement si, les ressources requises sont inférieures aux ressources disponibles : $\forall k \in [1, r], b_k^{t_j} \leq b_k^C$.*

Il existe une sous-classe des jeux à ressources appelés *jeux flous* [3, 10, 67]. Dans ces jeux, il n'y a ni tâches, ni ressources explicites mais, de manière plus abstraite, les agents peuvent choisir leur niveau d'implication dans la coalition (équivalent aux ressources disponibles). La fonction caractéristique est alors dépendante du niveau d'implication de chacun des agents.

Pour des problèmes de chaînes logistiques où les agents sont contraints spatialement et ont des capacités de transport limitées, il a été proposé les *network flow games* (NFG) [22, 30, 35, 36]. Ces

modèles représentent la fonction caractéristique par un graphe de flot où les agents sont des arcs et leur capacité est naturellement le flot qui les traversent. La valeur d'une coalition C est alors égale au flot maximum passant uniquement par les agents de la coalition. De manière intéressante, calculer le cœur ou le nucléole d'un NFG simple où tous les agents ont une capacité de 1 est un problème polynomial, mais cela devient NP-difficile dans les autres cas [22, 36].

4 Dépendances entre coalitions

Dans les jeux de coalitions classiques, il est fait l'hypothèse que les coalitions sont indépendantes les unes des autres, au sens où il n'y a que les agents qui composent une coalition qui ont une influence sur l'utilité produite. Cette indépendance est renforcée par le fait que les structures de coalitions sont disjointes, les agents formant alors des sous-ensembles isolés les uns des autres. Or, dans certains domaines d'application, la formation d'une coalition particulière pourrait tout à fait influencer – positivement ou négativement – sur d'autres coalitions. Par exemple dans un réseau pair-à-pair, la formation de certaines coalitions de machines peut réduire l'efficacité des communications d'autres coalitions [43]. Ainsi pour relâcher l'hypothèse d'indépendance, la littérature propose deux approches : les jeux à coalitions recouvrantes et les jeux à externalités.

4.1 Jeux à coalitions recouvrantes

Dans un jeu à compétences, l'indépendance entre les coalitions peut être une limite comme, par exemple, si un agent est le seul à posséder une compétence requise pour plusieurs tâches différentes. Dans le même ordre d'idée, si le pouvoir d'une coalition d'un jeu à ressources surpasse de celui requis pour réaliser ses tâches, alors les ressources supplémentaires sont perdues. Ces deux exemples posent une question commune : est-il possible que les agents puissent participer à plusieurs coalitions ? Pour pallier cette limite, la littérature propose les *jeux à coalitions recouvrantes* [16, 58] qui permettent aux agents de participer à plusieurs coalitions simultanément.

Definition 10 (Jeu à coalitions recouvrantes)

Un jeu à coalitions recouvrantes est un tuple $\mathcal{G} = \langle N, v \rangle$ où :

- $N = \{a_1, \dots, a_n\}$ est un ensemble d'agents,
- $v : [0, 1]^n \rightarrow \mathbb{R}$ est une fonction caractéristique qui à chaque vecteur de n réels dans

État de l'art

$[0, 1]$ associe un réel, appelé utilité de la coalition et noté $v(C)$ où $C \subseteq N$.

Ces vecteurs de taille n représentent des coalitions, appelées *coalitions recouvrantes*, dont les agents peuvent faire plus ou moins partie.

Definition 11 (Coalitions recouvrantes) Une coalition recouvrante C est définie par un vecteur $r = (r_1, \dots, r_n)$ où r_j est la fraction des ressources de l'agent a_j qu'il attribut à C . Si $r_j = 0$, alors l'agent a_j n'est pas membre de C . Le support d'une coalition recouvrante est le nombre d'agents contribuant à la coalition avec un montant strictement supérieur à 0.

Le cadre classique ne permettant pas de décrire une solution dans un jeu de coalitions recouvrantes, les notions d'imputation et de solution, et par extension les concepts de solutions, doivent être redéfinis.

Definition 12 (Imputation) Soit une structure de coalitions $CS \in \mathcal{CS}_N$, $|CS| = k$, une imputation pour CS est un tuple $x = (x^1, \dots, x^k)$ où $x^i \in \mathbb{R}^n$ avec $i \in [1, k]$, et tel que pour chaque coalition partielle $r^i \in CS$ nous avons $\sum_{j=1}^n x_j^i = v(r^i)$ et $r_j^i = 0$ implique $x_j^i = 0$.

L'ensemble de toutes les imputations pour la structure de coalitions CS est notée $I(CS)$. Avec l'imputation, nous pouvons redéfinir la solution.

Definition 13 (Solution) Une solution pour un ensemble d'agents $J \subseteq N$ est un tuple (CS, x) où $CS \in \mathcal{CS}_J$ et $x = (x^1, \dots, x^k) \in I(CS)$, avec $k = |CS|$. Le gain p_j d'un agent j est $p_j(CS, x) = \sum_{i=1}^k x_j^i$.

Ainsi, le cœur devient comme suit :

Definition 14 (Cœur recouvrant) Une solution (CS, x) est dans le cœur recouvrant si pour tout ensemble d'agents $J \subseteq N$, pour toute structure de coalitions CS_J pour J , et pour toute imputation $y \in I(CS_J)$, nous avons $p_j(CS_J, y) \leq p_j(CS, x)$ pour tout agent $j \in J$.

La complexité des jeux recouvrants est dépendante du montant des ressources des agents, de la taille maximale des coalitions et de l'ordre des interactions entre les agents. Le problème de décision associé reste NP-difficile [68].

4.2 Jeux à externalités

Dans certaines situations, les coalitions peuvent influencer les autres. Par exemple, elle peuvent entrer en conflit avec d'autres coalitions lorsqu'elles doivent utiliser des outils ou réaliser des tâches en exclusion mutuelle. Ce type de situation est modélisé par les *jeux à externalités*, aussi appelés *partition function games* (PFG) [61]. Les PFGs définissent une notion de *coalitions intégrées* qui caractérise les coalitions dont l'utilité dépend de la structure de coalitions dans laquelle elles apparaissent.

Definition 15 (Coalition intégrée) Une coalition intégrée est un tuple (C, \mathcal{C}) où $C \in \mathcal{C}$ et \mathcal{C} une structure de coalitions. L'ensemble des coalitions intégrées de N est noté E_N . La fonction caractéristique est redéfinie comme suit : $v : E_N \rightarrow \mathbb{R}$. Ainsi, $v(C, \mathcal{C})$ est la valeur de la coalition intégrée C dans \mathcal{C} .

Si les concepts des solutions ensemblistes classiques peuvent être facilement redéfinis dans le cadre des PFGs [17, 61], cela n'est pas aussi aisé pour les concepts de solutions singleton car les axiomes garantissant l'unicité de ces concepts dans le cadre classique ne le garantissent plus dans le cadre des PFGs [42]. De plus, calculer des solutions stables devient plus difficile et peu de travaux ont proposé des solutions algorithmiques à ce problème [43, 50, 62] car la taille de la fonction caractéristique dépend maintenant du nombre de structures de coalitions, et non plus du nombre de coalitions.

5 Incertitude dans les jeux

Une autre hypothèse majeure des jeux de coalitions classiques est le fait que la fonction caractéristique est déterministe. En effet, dans le cadre classique, la valeur d'une coalition est connue et le gain obtenu à la coopération des agents est fixe : il n'est soumis à aucun aléa. Dans le même ordre d'idée, les agents ont une connaissance parfaite de la fonction caractéristique. Or, cela implique que les agents ont une connaissance parfaite des compétences ou capacités des autres agents, ainsi que des synergies qu'il existe entre eux. S'affranchir de cela implique alors d'introduire de l'incertitude dans les jeux de coalitions.

5.1 Jeux à informations privées

Une première source d'incertitude peut venir du fait que les agents disposent d'informations pri-

État de l'art

vées. Dans la littérature, ces informations privées décrivent les connaissances des agents sur leurs capacités ou celles des autres. Les modèles qui intègrent donc cette forme d'incertitude sont donc des extensions des jeux à capacités, qu'ils soient quantitatifs [8, 57, 58, 59] ou qualitatifs [38].

Dans un cadre quantitatif, les *jeux à informations privées* ajoutent aux jeux à capacités une *probabilité de succès* associée à chaque agent. Ceci représente une fiabilité dans la réalisation de tâches (et donc une capacité quantitative) [8, 57, 59]. Cette probabilité de succès est une information privée des agents : ils ne connaissent pas les probabilités de succès des autres agents. Les concepts de solutions sont adaptés de manière à considérer les valeurs de coalitions espérées (bien qu'il faille formuler des hypothèses sur la manière dont l'échec d'un agent influe sur la valeur d'une tâche). Il existe deux approches, fondées toutes deux sur la répétition des jeux, pour tenir compte de l'incertitude qui en résulte. D'un côté, dans les travaux de Blankenburg *et al.* [8], les agents doivent observer les coalitions formées pour obtenir leur gain. À partir de cela, il estime leurs capacités des agents qui les composent. Cette estimation forme une confiance envers les autres, confiance qui est ensuite échangée et agrégée par les différents agents pour calculer une estimation globale, sur le même principe que les systèmes de réputations [53]. D'un autre côté, Shehory et Kraus [57, 58, 59] proposent une approche plus directe : chaque agent demande explicitement aux autres leur probabilité de succès. Comme ces probabilités peuvent changer au cours du temps, les agents doivent régulièrement communiquer. Il y a donc un fort coût de communication bien que cela réduise l'incertitude.

Dans un cadre qualitatif, l'information privée ne s'applique plus directement aux capacités. En effet, dans un tel cadre, un agent sait exactement quelles compétences il possède et ces dernières ne sont pas quantifiées. C'est pourquoi dans le modèle de Kraus *et al.* [38], il est ajouté un coût à chaque tâche, coût qui est spécifique à chaque agent. L'information privée porte alors sur ce coût. Kraus *et al.* réduisent alors le problème de formation de coalitions à un problème d'enchères descendantes. Un commissaire-priseur propose des tâches à un prix décroissant et les agents peuvent tous proposer des coalitions. Les membres de ces coalitions peuvent ou non accepter en fonction de ce qu'ils estiment des coûts privés des autres agents.

Notons que les modèles présentés ici partagent tous la perte d'optimalité, les agents prenant des décisions avec des éléments inconnus, et formant alors des coalitions dans lesquelles leurs gains sont moindres que ce qu'ils avaient escomptés. Notons que ces approches ont une autre conséquence : le processus de formation de coalitions doit nécessairement passer par une phase protocolaire (répétition, communication, enchères).

5.2 Fonctions caractéristiques stochastiques

Dans un jeu classique, la fonction caractéristique qui décrit la valeur produite par une coalition est déterministe. Or, une seconde source d'incertitude peut être non plus les capacités des agents en elles-mêmes mais le résultat de l'exécution des tâches qu'entreprennent les coalitions. En effet, dans un contexte applicatif, il n'est pas garanti que les agents puissent connaître avec certitude le résultat de leurs actions, indépendamment de leurs compétences ou leurs ressources. Ceci a conduit à définir des *jeux à fonction caractéristique stochastique*. [9, 14, 15, 18, 20, 19, 26, 34]

Une première manière de penser cette stochastocité consiste à associer à chaque coalition non pas une valeur mais une variable aléatoire bayésienne [20, 19] ou floue [9, 26, 40]. Par exemple dans le cadre bayésien, les travaux de Charnes et Granot définissent la fonction caractéristique comme $v : 2^N \rightarrow \mathcal{X}_{2^N}$ où les variables aléatoires suivent des lois normales [20, 19]. Les imputations sont calculées à partir du gain espéré des coalitions et les concepts de solutions sont redéfinis sur l'espérance de la fonction caractéristique, comme le *nucléole a priori* [19].

Si cette approche est très générale, elle ne permet pas de modéliser les raisons sous-jacentes à l'incertitude. C'est pourquoi d'autres approches proposent d'augmenter les jeux de coalitions avec un modèle d'environnement stochastique, modélisant le fait que les agents ont une incertitude sur les effets de leurs actions. Par exemple, Chalkiadakis et Boutilier associent des jeux de coalitions à des processus de décision markoviens partiellement observables [15]. Ici, le processus markovien décrit des transitions non-déterministes entre des états du monde en fonction des structures de coalitions formées. Chalkiadakis et Boutilier définissent le *cœur bayésien* [14, 18], qui revient simplement à l'approche de Charnes et Granot, c'est-à-dire en s'appuyant sur l'espérance des valeurs des coalitions en fonction des actions jointes de leurs membres.

Dans le même ordre d'idée, Jeong et Shoham

État de l'art

ont, quant à eux, proposé des jeux de coalitions associées à des modèles de mondes possibles [34]. Ici, les agents représentent leur incertitude comme une distribution de probabilité sur un ensemble de mondes, chacun représentant un jeu de coalitions avec une fonction caractéristique déterministe unique. À chaque monde est associé un ensemble fini d'imputations sur lesquelles les agents expriment des préférences. Jeong et Shoham proposent alors de nouveaux concepts de solutions fondés, certes sur les préférences établies par les agents, mais surtout sur une notion de connaissance spécifique. Le concept *ex-ante* caractérise une solution avant l'observation du monde véritable, le concept *ex-interim* après réduction des mondes possibles lors que les agents reçoivent des informations mais avant l'observation du monde véritable, et le concept *ex-post* après l'observation du monde véritable. Sans incertitude, ces trois concepts équivalent au cœur.

6 Résolution décentralisée

Les approches classiques de résolution de jeux de coalitions, qu'ils soient classiques ou qu'ils en soient des variantes, se fondent généralement sur des processus centralisés, exacts ou approchés avec une diversité de méthodes : programmation linéaire, programmation dynamique, algorithmes gloutons, ou méta-heuristiques [51].

Toutefois, dans cet article, nous nous intéressons uniquement aux algorithmes décentralisés, plus adaptés à un cadre multi-agents. Notons tout d'abord que ces algorithmes décentralisés sont souvent proposés pour contexte applicatif spécifique, en particulier se restreignant aux jeux à ressources [27, 29, 44]. Toutefois, ils partagent des bases communes et deux catégories se distinguent : les algorithmes fondés sur des négociations à partir de préférences que les agents calculent localement [12, 37, 57, 58, 59, 60], et les algorithmes fondés sur des graphes dynamiques. [7, 27, 29, 44, 64].

Dans les approches fondées sur la négociation, une première étape consiste à extraire les préférences des agents sur les coalitions possibles, en calculant leur utilité espérée. Par exemple, Shehory et Kraus [57, 58, 59] établissent ces préférences via protocole de communication itéré : les agents constituent une liste restreinte de coalitions – d'une taille maximale fixée – qu'ils désirent former et contactent les agents en faisant partie afin de s'enquérir de leurs capacités, ce qui leur permet d'estimer leur valeur. D'autres approches se passent de communication et s'ap-

puient plutôt sur des heuristiques, soit en donnant une préférence aux agents les plus proches dans des modèles spatiaux [37, 60], ou en minimisant les erreurs d'attribution de ressources [12]. Enfin, une fois ces estimations réalisées, la seconde étape consiste à négocier : des coalitions sont proposées (par les agents eux-mêmes ou un commissaire-priseur) et, si tous les membres l'acceptent, elles sont formées.

Dans les approches fondées sur des graphes dynamique, des contraintes en termes de communication sont ajoutées au modèle : les agents ne peuvent pas communiquer avec n'importe quels autres. Pour cela, nous trouvons des représentations sous forme de *graphes contraints* [7, 64] ou de *réseaux d'agents organisés* [27, 29, 44]. Il s'agit essentiellement d'une distinction cosmétique car les deux représentations possèdent un fonctionnement similaire. La structure de coalitions est représentée par un graphe dont les agents en sont les sommets, et les arêtes entre ces sommets représentent une interaction. Une coalition est un sous-graphe connexe du graphe général. Durant chaque étape du protocole, les agents ne peuvent former des coalitions qu'avec un ensemble de voisins fixés a priori, et décident de cela via des heuristiques. Si deux agents décident de rejoindre une même coalition alors une arête est ajoutée entre eux. Si un agent souhaite quitter une coalition, les arêtes entre cet agent et les membres de la coalition sont détruites. Selon le modèle, la formation de coalitions s'arrête soit après un certain temps [29, 44], soit lorsqu'il n'y a plus de tâches à accomplir [7, 27, 64]. Dans ces modèles, la nature des heuristiques influe évidemment grandement sur l'optimalité de la solution. Par exemple, une politique gloutonne sur les performances des agents produit de bons résultats tandis qu'une heuristique encourageant la diversité des capacités est peu satisfaisante [27, 29]. Enfin, malgré l'ajout de contraintes de communication réduisant le nombre de structures de coalitions possibles à un instant donné, le problème reste NP-complet [64].

Comme il a parfois été souligné ci-dessus, les protocoles décentralisés peinent souvent à trouver et/ou garantir une solution globalement optimale [12, 27, 29, 59]. Cependant, trouver une solution optimale passe bien souvent par partager totalement l'information ou bien par tester plusieurs structures dans le temps pour finir par converger vers l'optimale. Un point commun à tous ces modèles est l'absence de garantie quant à l'optimalité des solutions.

État de l'art

7 Conclusion

Dans cet article, nous avons présenté une synthèse des différents modèles de formation de coalitions, en partant du cadre classique, puis en en relâchant au fur et à mesure les hypothèses. Trois axes se dégagent : la modélisation des capacités des coalitions, des dépendances entre les coalitions et de l'incertitude. Il nous semble important de souligner que le manque d'information et l'incertitude sont des points-clés dans la perte d'optimalité des protocoles décentralisés. Proposer des modèles intégrant de l'apprentissage ou des mécanismes d'estimation efficaces, et non pas des heuristiques, nous semble un piste de recherche pertinente.

Références

- [1] K. V. Aadithya, T. P. Michalak, and N. R. Jennings. Representation of coalitional games with algebraic decision diagrams. In *Proc. of the 10th AAMAS*, pages 1121–1122, 2011.
- [2] K. R. Apt and A. Witzel. A generic approach to coalition formation. *Int. Game Theory Rev.*, 11(3) :347–367, 2009.
- [3] J.-P. Aubin. Cooperative fuzzy games. *Math. Oper. Res.*, 6 :1–13, 1981.
- [4] H. Aziz and B. De Keijzer. Complexity of coalition structure generation. In *Proc. of the 11th AAMAS*, pages 191–198, 2011.
- [5] Y. Bachrach and J. S. Rosenschein. Coalitional skill games. In *Proc. of the 7th AAMAS*, pages 1023–1030, 2008.
- [6] J. F. Banzhaf III. Weighted voting doesn't work : A mathematical analysis. *Rutgers Univ. Law Rev.*, 19 :317, 1964.
- [7] F. Bistaffa, A. Farinelli, J. Cerquides, J. Rodríguez-Aguilar, and S. D. Ramchurn. Algorithms for graph-constrained coalition formation in the real world. *ACM TIST*, 8(4) :1–24, 2017.
- [8] B. Blankenburg, R. K. Dash, S. D. Ramchurn, M. Klusch, and N. R. Jennings. Trusted kernel-based coalition formation. In *Proc. of the 4th AAMAS*, pages 989–996, 2005.
- [9] B. Blankenburg, M. Klusch, and O. Shehory. Fuzzy kernel-stable coalitions between rational agents. In *Proc. of the 2nd AAMAS*, pages 9–16, 2003.
- [10] S. Borkotokey and R. Neog. Dynamic resource allocation in fuzzy coalitions : a game theoretic model. *Fuzzy Optim. Decis. Mak.*, 13 :211–230, 2014.
- [11] S. Branzei. State of the art : Solution concepts for coalitional games. *BRAIN*, 1(2) :89–101, 2010.
- [12] J. Bremer and S. Lehnhoff. Decentralized coalition formation with agent-based combinatorial heuristics. Technical report, Ediciones Universidad de Salamanca, 2017.
- [13] E. Calvo and E. Gutiérrez. Solidarity in games with a coalition structure. *Math. Soc. Sci.*, 60(3) :196–203, 2010.
- [14] G. Chalkiadakis and C. Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *Proc. of the 3rd AAMAS*, pages 1090–1097, 2004.
- [15] G. Chalkiadakis and C. Boutilier. Sequential decision making in repeated coalition formation under uncertainty. In *Proc. of the 6th AAMAS*, pages 347–354, 2008.
- [16] G. Chalkiadakis, E. Elkind, E. Markakis, and N. R. Jennings. Overlapping coalition formation. In *Proc. of the 4th WINE*, pages 307–321, 2008.
- [17] G. Chalkiadakis, E. Elkind, and M. J. Wooldridge. Computational aspects of cooperative game theory. *Synth. Lect. Artif. Intell. Mach. Learn.*, 5(6) :1–168, 2011.
- [18] G. Chalkiadakis, E. Markakis, and C. Boutilier. Coalition formation under uncertainty : Bargaining equilibria and the Bayesian core stability concept. In *Proc. of the 6th AAMAS*, pages 1–8, 2007.
- [19] A. Charnes and D. Granot. Prior solutions : Extensions of convex nucleus solutions to chance-constrained games. Technical report, Texas Univ., 1973.
- [20] 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.
- [21] M. Davis and M. Maschler. The kernel of a cooperative game. *Nav. Res. Logist.*, 12(3) :223–259, 1965.
- [22] X. Deng, Q. Fang, and X. Sun. Finding nucleolus of flow game. *J. Comb. Optim.*, 18(1) :64–86, 2009.
- [23] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2) :257–266, 1994.
- [24] E. Einy, D. Monderer, and D. Moreno. The least core, kernel and bargaining sets of large games. *Econ. Theory*, 11(3) :585–601, 1998.
- [25] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. J. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. *Math. Log. Q.*, 55(4) :362–376, 2009.
- [26] J. M. Gallardo and A. Jiménez-Losada. A characterization of the Shapley value for cooperative games with fuzzy characteristic function. *Fuzzy Sets Syst.*, 398 :98–111, 2020.
- [27] M. E. Gaston and M. desJardins. Agent-organized networks for dynamic team formation. In *Proc. of the 4th AAMAS*, pages 230–237, 2005.
- [28] D. B. Gillies. Solutions to general non-zero-sum games. *Contributions to the Theory of Games*, 4 :47–85, 1959.
- [29] R. Grinton, P. Scerri, and K. Sycara. Agent-based sensor coalition formation. In *Proc. of the 11th FUSSION*, pages 1–7, 2008.
- [30] D. Granot and F. Granot. On some network flow games. *Math. Oper. Res.*, 17(4) :792–841, 1992.
- [31] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. On the complexity of core, kernel, and bargaining set. *Artif. Intell.*, 175(12-13) :1877–1910, 2011.
- [32] R. Ichimura, Y. Sakurai, S. Ueda, A. Iwasaki, M. Yokoo, and S. Minato. Compact representation scheme

État de l'art

- of coalitional games based on multi-terminal zero-suppressed binary decision diagrams. In *Proc. of Workshop on Social Choice and Artificial Intelligence at the 22nd IJCAI*, page 40, 2011.
- [33] S. Jeong and Y. Shoham. Marginal contribution nets : A compact representation scheme for coalitional games. In *Proc. of the 6th ACM EC*, pages 193–202, 2005.
- [34] S. Jeong and Y. Shoham. Bayesian coalitional games. In *Proc. of the 23rd AAAI*, pages 95–100, 2008.
- [35] E. Kalai and E. Zemel. Totally balanced games and games of flow. *Math. Oper. Res.*, 7(3) :476–478, 1982.
- [36] W. Kern and D. Paulusma. Matching games : the least core and the nucleolus. *Math. Oper. Res.*, 28(2) :294–308, 2003.
- [37] L. Khalouzadeh, N. Nematbakhsh, and K. Zamani-far. A decentralized coalition formation algorithm among homogeneous agents. *J. Theor. Appl. Inf. Technol.*, 22(1), 2010.
- [38] S. Kraus, O. Shehory, and G. Taase. Coalition formation with uncertain heterogeneous information. In *Proc. of the 2nd AAMAS*, pages 1–8, 2003.
- [39] M. Kurz. Coalitional value. In A.E. Roth, editor, *The Shapley value : essays in honor of Lloyd S. Shapley*, pages 155–173. Cambridge Univ. Press, 1988.
- [40] M. Mareš. *Fuzzy Cooperative Games*. Springer, 2001.
- [41] M. Maschler, B. Peleg, and L. S. Shapley. Geometric properties of the kernel, nucleolus, and related solution concepts. *Math. Oper. Res.*, 4(4) :303–338, 1979.
- [42] T. P. Michalak, T. Rahwan, D. Marciniak, M. Szamotulski, and N. R. Jennings. Computational aspects of extending the Shapley value to coalitional games with externalities. In *Proc. of the 19th ECAI*, pages 197–202, 2010.
- [43] T. P. Michalak, T. Rahwan, J. Sroka, A. Dowell, M. J. Wooldridge, P. J. McBurney, and N. R. Jennings. On representing coalitional games with externalities. In *Proc. of the 10th EC*, pages 11–20, 2009.
- [44] R.-C. Mihalescu, M. Vasirani, and S. Ossowski. Dynamic coalition adaptation for efficient agent-based virtual power plants. In *Proc. of the 9th MATES*, pages 101–112, 2011.
- [45] R. Mochaourab and E. A. Jorswieck. Coalitional games in MISO interference channels : Epsilon-core and coalition structure stable set. *IEEE Trans. Signal Process.*, 62(24) :6507–6520, 2014.
- [46] O. Morgenstern and J. Von Neumann. *Theory of games and economic behavior*. Princeton Univ. Press, 1953.
- [47] A. S. Nowak and T. Radzik. A solidarity value for n-person transferable utility games. *Int. J. Game Theory*, 23(1) :43–48, 1994.
- [48] N. Ohta, A. Iwasaki, M. Yokoo, K. Maruono, V. Conitzer, and T. Sandholm. A compact representation scheme for coalitional games in open anonymous environments. In *Proc. of the 21st AAAI*, volume 6, page 21st, 2006.
- [49] M. J. Osborne and A. Rubinstein. *A course in game theory*. MIT press, 1994.
- [50] T. Rahwan, T. Michalak, M. J. Wooldridge, and N. R. Jennings. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artif. Intell.*, 186 :95–122, 2012.
- [51] T. Rahwan, T. P. Michalak, M. J. Wooldridge, and N. R. Jennings. Coalition structure generation : A survey. *Artif. Intell.*, 229 :139–174, 2015.
- [52] D. Ray. *A game-theoretic perspective on coalition formation*. Oxford Univer. Press, 2007.
- [53] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1) :33–60, 2005.
- [54] D. Schmeidler. The nucleolus of a characteristic function game. *SIAM J. Appl. Math.*, 17(6) :1163–1170, 1969.
- [55] L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28) :307–317, 1953.
- [56] L. S. Shapley and M. Shubik. Quasi-cores in a monetary economy with nonconvex preferences. *Econometrica*, pages 805–827, 1966.
- [57] O. Shehory and S. Kraus. Task allocation via coalition formation among autonomous agents. In *Proc. of the 14th IJCAI*, pages 655–661, 1995.
- [58] O. Shehory and S. Kraus. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In *Proc. of 2nd ICMAS*, pages 330–337, 1996.
- [59] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artif. Intell.*, 101(1-2) :165–200, 1998.
- [60] M. Sims, C. V. Goldman, and V. Lesser. Self-organization through bottom-up coalition formation. In *Proc. of the 2nd AAMAS*, pages 867–874, 2003.
- [61] R. M. Thrall and W. F. Lucas. N-person games in partition function form. *Nav. Res. Logist.*, 10(1) :281–298, 1963.
- [62] S. Ueda, S. Hasegawa, N. Hashimoto, N. Ohta, A. Iwasaki, and M. Yokoo. Handling negative value rules in MC-net-based coalition structure generation. In *Proc. of the 11th AAMAS*, volume 12, pages 795–804, 2012.
- [63] M. Vinyals, F. Bistaffa, A. Farinelli, and A. Rogers. Stable coalition formation among energy consumers in the smart grid. In *Proc. of the 3rd ATEES*, page 35, 2012.
- [64] T. Voice, M. Polukarov, and N. R. Jennings. Coalition structure generation over graphs. *J. Artif. Intell. Res.*, 45 :165–196, 2012.
- [65] M. J. Wooldridge and P. E. Dunne. On the computational complexity of coalitional resource games. *Artif. Intell.*, 170(10) :835–871, 2006.
- [66] G. Xu, H. Dai, D. Hou, and H. Sun. A-potential function and a non-cooperative foundation for the solidarity value. *Oper. Res. Lett.*, 44 :86–91, 2016.
- [67] G. Xu, X. Lia, H. Suna, and J. Su. The Myerson value for cooperative games on communication structure with fuzzy coalition. *J. Intell. Fuzzy Syst.*, 33 :27–39, 2017.
- [68] Y. Zick, G. Chalkiadakis, and E. Elkind. Overlapping coalition formation games : Charting the tractability frontier. In *Proc. of the 11th AAMAS*, pages 787–794, 2012.

