



HAL
open science

Contribution des systèmes multi-agents à la conduite et à la simulation de systèmes évolutifs

Arnaud Doniec

► **To cite this version:**

Arnaud Doniec. Contribution des systèmes multi-agents à la conduite et à la simulation de systèmes évolutifs. Informatique [cs]. UPHF, 2019. tel-04446820

HAL Id: tel-04446820

<https://imt-nord-europe.hal.science/tel-04446820v1>

Submitted on 8 Feb 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.

Mémoire d'habilitation à diriger des recherches

Spécialité : Informatique

ARNAUD DONIEC

Université Polytechnique des Hauts-de-France
IMT Lille Douai

Contribution des systèmes multi-agents à la conduite et à la simulation de systèmes évolutifs

soutenue le XXX devant le jury composé de :

Rapporteur	—	Amal EL FALLAH SEGHROUCHNI, Professeur, Sorbonne Université
Rapporteur	—	Marie-Pierre GLEIZES, Professeur, Université Toulouse III - Paul Sabatier
Rapporteur	—	Michel OCCELLO, Professeur, Université Grenoble Alpes
Examineur	—	Abder KOUKAM, Professeur, Université de technologie Belfort-Montbéliard
Examineur	—	Stéphane LECOUCHE, Professeur, IMT Lille Douai
Promoteur HDR	—	René MANDIAU, Professeur, Université Polytechnique des Hauts-de-France

Table des matières

1	Cadre de travail, problématiques et contextes applicatifs	5
1.1	Les systèmes évolutifs : définition et problématiques	5
1.1.1	Positionnement des systèmes évolutifs	5
1.1.2	Formalisme multi-agent d'un système évolutif	8
1.1.3	Problématiques et thèmes de recherche	9
1.2	Illustrations et domaines d'application	11
1.2.1	Modélisation des comportements humains au sein d'un espace de vente	11
1.2.2	Exploration multi-robot	12
1.2.3	Production décentralisée d'électricité dans les centrales virtuelles	14
1.2.4	Gestion de la ressource en eau pour les voies navigables	17
1.3	Conclusion	20
2	Conception de simulations multi-agents dirigée par les données	23
2.1	Simulations multi-agents et fouille de données : quelles synergies ?	23
2.1.1	Conception de simulations multi-agents	24
2.1.2	Apprentissage et fouille de données	28
2.1.3	Modélisation dirigée par les données et synergies possibles avec les systèmes multi-agents	29
2.2	Agentification dirigée par les données pour la simulation d'activité humaine	31
2.2.1	Approches pour l'identification et l'extraction automatique de comportements à partir d'observations réelles	32
2.2.2	Proposition d'une approche d'agentification dirigée par les données pour des simulations sociales et situées	34
2.3	Application à la simulation d'un espace de vente	41
2.3.1	Représentation du magasin et des produits	43
2.3.2	Attributs retenus pour le client	43
2.3.3	Modélisation de l'attraction client-produit	44
2.3.4	Description du comportement d'agents « client »	45
2.3.5	Paramétrage du modèle d'agent consommateur à partir de données réelles	47
2.3.6	Expérimentations	50
2.4	Conclusion	51
3	Coordination d'agents en environnement dynamique et incertain	53
3.1	La coordination vue comme un problème de satisfaction de contraintes	53
3.1.1	Coordination dans les systèmes en réseau	55
3.1.2	Coordination pour l'exploration multi-robot	58

TABLE DES MATIÈRES

3.1.3	Analyse des approches de coordination par contraintes	63
3.2	La coordination vue sous l'angle de la planification	65
3.2.1	Planification pour le problème d'engagement d'unités (UCP)	65
3.2.2	Planification pour la gestion de la ressource en eau dans les voies navigables	75
3.2.3	Analyse des approches de coordination par planification	85
3.3	Conclusion	85
4	Conclusion et perspectives	87
4.1	Synthèse des travaux de recherche	87
4.1.1	Contribution autour de la conception d'agents dirigée par les données	87
4.1.2	Contribution autour de la coordination d'agents en environnement dynamique et incertain	88
4.2	Perspectives	89
4.2.1	La simulation multi-agent de systèmes évolutifs	89
4.2.2	Coordination multi-agent pour le pilotage de systèmes évolutifs	91
4.2.3	Vers une approche intégrée mêlant coordination et simulation multi-agent	93
5	Annexe : curriculum vitæ	111
5.1	Etat Civil	111
5.2	Titres Universitaires	111
5.3	Parcours professionnel	112
5.4	Activités d'Enseignement	112
5.4.1	Enseignements réalisés à l'UVHC	112
5.4.2	Enseignements réalisés au sein de l'IMT Lille Douai	113
5.4.3	Supports de cours	115
5.4.4	Responsabilités administratives et pédagogiques	116
5.4.5	Projet d'enseignement	118
5.5	Activités de Recherche	119
5.5.1	Projets de recherche financés	119
5.5.2	Collaborations scientifiques	120
5.5.3	Encadrements	121
5.5.4	Autres activités	123
5.6	Publications	124
5.6.1	Indicateurs bibliométriques	124
5.6.2	Bilan Quantitatif	125
5.6.3	Liste des Publications	125

Table des figures

1.1	Principales caractéristiques des systèmes évolutifs étudiés	7
1.2	Exemples de flottes de robots. A gauche, l'équipe de l'université du Michigan [Olson <i>et al.</i> , 2013]. A droite, flotte de Wifibots utilisée au sein de l'URIA	13
1.3	Illustration du concept de centrale virtuelle	15
1.4	Schéma hydrologique du réseau des voies navigables en région Hauts-de-France	18
1.5	Rectangle de navigation associé à un bief	19
1.6	Ecluse	19
1.7	Inondation de berges suite au débordement du bief Douai-Don-Cuinchy (juin 2016)	20
2.1	Processus de construction d'une simulation (adapté de [Shannon, 1998] et [Galán <i>et al.</i> , 2009])	25
2.2	Structure générale d'analyse de scène vidéo [Li <i>et al.</i> , 2015]	32
2.3	Processus d'analyse et de compréhension automatique d'observations réelles	34
2.4	Modèle de comportement considéré	35
2.5	Intégration des techniques d'apprentissage dans un processus de conception de simulations multi-agents	36
2.6	Exemples de captures de trajectoires en environnement intérieur [Saffar, 2013, Sylvain, 2016]	38
2.7	Exemples de découpage de l'environnement observé. A gauche, découpage manuel et annotation de celui-ci par le concepteur de la simulation. A droite, découpage automatique de l'environnement à l'aide de diagramme de Voronoi [Saffar <i>et al.</i> , 2011]	39
2.8	Représentation symbolique d'une trajectoire [Saffar <i>et al.</i> , 2011, Höferlin <i>et al.</i> , 2011]	40
2.9	Transformation de trajectoire en matrice de points de passage	40
2.10	Choix de vecteurs caractéristiques possibles sur les données de déplacements individuels	41
2.11	Vue globale de la simulation	42
2.12	Agencement du supermarché considéré	47
2.13	Exemples de <i>clusters</i> obtenus	49
2.14	Comparaison réel / simulé sur la base de cartes de chaleur	51
3.1	Utilisation de CSP pour la coordination d'agents en carrefour [Doniec <i>et al.</i> , 2008]	54
3.2	Modélisation sous forme d'un graphe d'une partie du réseau des Hauts-de-France (cf. figure 1.4)	56
3.3	Déploiement d'un réseau MANET sur une flotte de robots	58
3.4	Positions futures de deux robots devant se coordonner	59
3.5	Ordonnancement des déplacements possibles pour un robot	61
3.6	Expérimentations sur différents types d'environnements	62

TABLE DES FIGURES

3.7	Expérimentations : durée de l'exploration en fonction du nombre de robots	63
3.8	Expérimentations : variation du nombre de robots coordonnés	64
3.9	Plateforme « Energie répartie »	72
3.10	Réseau IEEE 14 nœuds	72
3.11	Utilisation de l'algorithme P-DPGO avec 6 sources de production et l'apparition de pannes	73
3.12	Coût cumulé de la production	74
3.13	Différence de coûts pour des horizons de planification $h = 1$ et $h = 24$	74
3.14	Non-satisfaction de la demande avec un horizon de planification $h = 1$	75
3.15	Discretisation d'un bief	76
3.16	Localisation du réseau considéré	80
3.17	Décomposition en agents du réseau considéré	80
3.18	Scénario 1 : navigation autorisée sur 12h et pas de pluie	81
3.19	Scénario 2 : navigation autorisée sur 24h et pas de pluie	81
3.20	Scénario 3 : navigation autorisée sur 24h et occurrence d'un évènement pluvieux intense non anticipé	82
3.21	Scénario 3 : navigation autorisée sur 24h et occurrence d'un évènement pluvieux intense anticipé	83
3.22	Scénario 3 : navigation autorisée sur 24h avec un évènement pluvieux intense anticipé qui finalement ne se produit pas	83
3.23	Evolution du coût d'une politique jointe en fonction des itérations de l'algorithme	84

Introduction

Les travaux exposés dans ce document ont été menés au sein de l'URIA (Unité de Recherche en Informatique et Automatique)¹ dont le thème de recherche fédérateur concerne l'ingénierie des systèmes évolutifs.

Il est difficile de donner une définition consensuelle d'un système évolutif puisque le terme est utilisé dans différents contextes et domaines de recherche. En automatique par exemple, il fait souvent référence à un système dynamique dont les caractéristiques (et donc le comportement) sont amenées à évoluer au cours du temps : soit de manière progressive (dérive), soit de manière soudaine (saut). Dans le domaine informatique, les systèmes évolutifs peuvent faire référence à des logiciels en exploitation pour lesquels certaines parties de code sont amenées à évoluer pour amener des nouvelles fonctionnalités, corriger des bugs, etc. Par abus de langage, le terme évolutif peut parfois être employé pour faire référence à des systèmes artificiels utilisant des algorithmes bioinspirés destinés à résoudre des problèmes d'optimisation² ou reproduire des phénomènes du vivant. Les sciences de la matière font également référence aux systèmes évolutifs pour l'étude des propriétés rhéologiques de matériaux en fonction du temps.

Les travaux de recherche de l'URIA sur les systèmes évolutifs s'intéressent à deux problématiques que sont leur conception et leur exploitation efficace. Dans le cadre des systèmes autogérés, dotés de capacités d'adaptation, les travaux de l'unité portent notamment sur :

- la conception d'architectures logicielles d'un système embarqué, distribué et mobile ;
- l'adaptation logicielle du système durant son fonctionnement ;
- la planification de la conduite du système tout au long de son cycle de vie ;
- la création de systèmes autonomes en termes de communication et de raisonnement.

Concernant l'analyse des systèmes évolutifs existants, les travaux de l'unité visent à :

- observer le système et modéliser par l'analyse de la donnée ses différents comportements ;
- simuler le système pour mieux comprendre sa dynamique interne et externe et en déduire des stratégies de conduite adaptée ;
- prédire les évolutions temporelles du système ;
- diagnostiquer les éventuelles défaillances qui peuvent apparaître.

L'ensemble des travaux de recherche de l'URIA se sont organisés autour de trois thèmes³ de recherche et de leurs interactions. Le premier thème : ICARE (Ingénierie des Composants pour Applications Réparties et Embarquées) est composé de deux permanents (à la date d'écriture de ce docu-

1. Depuis le 1er janvier 2019, les départements et unités de recherche de l'IMT Lille Douai sont regroupés au sein de CERI (Centre d'Enseignement, de Recherche et d'Innovation). L'URIA fait partie depuis cette date du CERI Numérique.

2. Dans ce contexte, le terme consacré est plutôt : algorithmes évolutionnaires.

3. A la date de rédaction de ce document, la structuration interne du nouveau CERI Numérique en thématique de recherche n'est pas encore actée. Je fais donc référence aux anciens thèmes de l'URIA pour le positionnement interne de mes travaux.

ment) et s'intéresse à la conception des modèles et des outils facilitant le développement de logiciels modulaires et adaptables dynamiquement [Tesone *et al.*, 2018] pour les systèmes embarqués et/ou distribués, et plus spécifiquement les robots mobiles autonomes [Dichtl *et al.*, 2018]. Le second thème : SUSE (SUivi des Systèmes Evolutifs) est composé de 8 permanents et se focalise sur des approches de suivi guidées par les données (modélisation expérimentale, identification de systèmes, modèles temporels) [Ajib *et al.*, 2018, Hadid *et al.*, 2018] visant à disposer d'une meilleure compréhension des phénomènes dynamiques et évolutifs et à permettre la mise en place de stratégies de conduite adaptative ou de maintenance prédictive. Le dernier thème : A³ (Agents, Apprentissage et Adaptation) est composé de 8 permanents travaillant que la modélisation et la prise de décision au sein de processus évolutifs avec des outils de l'apprentissage automatique et de l'intelligence artificielle (systèmes multi-agents, programmation par contraintes, approches markoviennes) [Karami *et al.*, 2016, Lujak *et al.*, 2017].

Mes activités de recherche s'inscrivent dans le thème A³ et ses interactions avec les deux autres thèmes de l'unité. Plus précisément, je me suis intéressé à des systèmes évolutifs présentant la caractéristique d'être distribués avec potentiellement une implication forte de l'être humain.

Ces systèmes seront modélisés dans la suite du document sous la forme de systèmes multi-agents (SMA). Ces derniers constituent aujourd'hui un formalisme incontournable pour le développement de systèmes (logiciels ou physiques) intelligents distribués et la modélisation de phénomènes sociaux [Mercure *et al.*, 2016, Clausen *et al.*, 2017]. Les éléments constitutifs d'un SMA, appelés agents, y interagissent entre eux (directement ou indirectement) au sein d'un environnement pour atteindre des objectifs (parfois collectifs, parfois individuels) de manière autonome.

Dans les travaux présentés par la suite, les évolutions des systèmes considérés pourront avoir des origines diverses. Soit elles seront subies et causées par des événements extérieurs au système ; soit elles seront le résultat du changement de comportement d'un ou plusieurs agents composant le système. Dans les deux cas, la dynamique temporelle du système en sera affectée soit de manière progressive (dérive lente et continue par exemple), soit de manière soudaine (par exemple, une panne). Notons également que ces évolutions pourront impacter le système de manière locale (seuls quelques agents seront concernés par l'évolution) ou dans sa globalité.

Axe de recherche

Deux problématiques autour des systèmes évolutifs seront présentées dans ce document. Une première concerne l'étude et la compréhension de ces systèmes via leur simulation. Il s'agit ici de modéliser ces systèmes et leurs évolutions dans le temps de manière à en reproduire leur dynamique. L'outil de simulation résultant de cette modélisation fournit alors un outil d'exploration virtuel permettant de soumettre le système étudié à différents scénarios et d'en visualiser les évolutions résultantes. Une seconde problématique va s'intéresser aux approches permettant la conduite de ces systèmes évolutifs. Il s'agit cette fois de proposer des mécanismes de décision permettant aux éléments constitutifs du système de s'adapter face à des changements de contexte. Ces deux problématiques s'inscrivent dans des axes de recherche majeurs en intelligence artificielle (et plus particulièrement les systèmes multi-agents) : la simulation comportementale d'une part et la coordination multi-agent d'autre part.

Les simulations comportementales, bien qu'elles soient devenues une alternative crédible aux simulations classiques, restent difficiles à mettre en œuvre (notamment lorsqu'il s'agit de rendre compte d'activités humaines et de phénomènes sociaux). De nombreux travaux visent donc à fournir des mé-

thodologies et outils facilitant le développement de telles simulations. L'expertise humaine nécessaire pour l'élaboration de ce type de simulation reste cependant difficile à synthétiser et à retranscrire de manière informatique. Mes travaux menés sur cet axe proposent de combiner modélisation multi-agent et modélisation dirigée par les données pour faciliter le développement de simulations comportementales. Ils s'appuient sur différentes techniques d'apprentissage automatique telles que la classification non supervisée (clustering).

La coordination est un axe de recherche particulièrement actif dans la communauté multi-agent. Elle désigne l'ensemble des processus qui permettent d'atteindre un état global souhaité tout en prenant en compte : l'organisation du SMA, les états locaux de chaque agent et leur capacité à communiquer entre eux. Mes travaux sur cet axe ont consisté en la proposition d'approches génériques de coordination reposant sur des modèles de raisonnement éprouvés en IA (programmation par contraintes, processus markoviens). Ces approches ont été notamment évaluées pour des problèmes nécessitant de la part des agents une adaptation en environnement incertain.

Contexte applicatif

D'un point de vue applicatif, plusieurs types de systèmes évolutifs seront étudiés dans ce document. La conception d'agents dirigée par les données et la simulation comportementale seront illustrées dans le contexte des industries du commerce. Le système ainsi considéré sera le couple espace de vente / client. Sur différentes durées d'observation, ce système est évolutif puisque le comportement des clients peut être changeant à l'échelle de l'heure, de la semaine, du mois, de l'année et influencé par différents facteurs : affluence du magasin, actions de marketing, interactions avec d'autres consommateurs, saisonnalité, etc. Le développement d'agents dans ce contexte requiert donc de s'appuyer sur ces différentes échelles d'observation.

La coordination multi-agent est quant à elle illustrée au travers trois applications : l'exploration multi-robot en environnement inconnu, la production décentralisée d'électricité et la gestion adaptative de l'eau dans les voies navigables. Dans les trois cas, il s'agit de mettre en place des stratégies de conduite distribuée capables de s'adapter lors de l'évolution des objectifs ou face à l'apparition d'évènements imprévus. Il peut s'agir de blocages dans le cadre multi-robot, de pannes dans le cas de la production électrique ou d'évènements climatiques inattendus tels que les crues pour les voies navigables.

Structure du document

Ce manuscrit est structuré en quatre chapitres. Le premier présente les enjeux et problématiques traités dans ce document. En particulier, une vision multi-agent des systèmes évolutifs est introduite et illustrée au travers différents contextes applicatifs.

Le second chapitre correspond au premier axe de recherche introduit ci-avant, à savoir : la conception d'agents dirigée par les données. Il présente les synergies possibles entre les systèmes multi-agents et la fouille de données et introduit une approche d'agentification de comportements humains dans la simulation.

Le chapitre 3 traite le second axe de recherche : la coordination d'agents en environnement dynamique et incertain. Deux contributions y sont détaillées : une première consistant à exprimer la

Introduction

coordination comme la solution d'un problème de satisfaction de contraintes; une seconde considérant la coordination sous l'angle de la planification.

Le chapitre 4 détaille les perspectives de chacun des deux axes de recherche et conclut ce manuscrit.

Chapitre 1

Cadre de travail, problématiques et contextes applicatifs

Notre société est actuellement confrontée à un certain nombre de problèmes auxquels elle aura à répondre dans un futur assez proche : tarissement des ressources naturelles, pollution, évolution de la démographie, changements climatiques. Ces défis sociétaux vont induire des évolutions profondes au sein de notre société. La transition énergétique par exemple va générer des bouleversements dans notre vie quotidienne : adaptation de nos modes de déplacement, modification de nos habitudes de consommation, évolution de nos modes de vie, etc.

Relever ces défis sociétaux requiert bien souvent la conception, la compréhension ou encore l'exploitation de systèmes de plus en plus complexes, aux multiples interactions et dont la dynamique dans le temps est évolutive.

Dans une première section, nous présenterons ces systèmes sous l'angle des systèmes multi-agents. Nous détaillerons ensuite les problématiques liées à la conception et l'exploitation de ces systèmes et nous les illustrerons au travers différents exemples d'application.

1.1 Les systèmes évolutifs : définition et problématiques

1.1.1 Positionnement des systèmes évolutifs

Afin de positionner notre cadre de travail et plus précisément les systèmes évolutifs, nous rappelons brièvement ce que sont les systèmes complexes, les systèmes dynamiques et les systèmes distribués.

1.1.1.1 Systèmes complexes

Il existe dans la littérature un grand nombre de définitions pour un système complexe. Elles sont souvent contextualisées par rapport à des disciplines diverses et variées comme par exemple : la biologie, la chimie, la physique, l'informatique, la sociologie, l'économie, etc.

Comme le propose Yaneer Bar-Yam, il est néanmoins possible de recenser un certain nombre

de propriétés communes à ces différentes définitions [Bar-Yam, 1997]. Ainsi, un système complexe dispose généralement d'une ou plusieurs des caractéristiques suivantes :

- un nombre d'éléments constitutifs relativement grand ;
- des interactions fortes et multiples entre ces éléments ;
- une diversité et variabilité de ces éléments et de leurs interactions ;
- une difficulté à prédire les effets des interactions des éléments constitutifs sur le comportement global du système.

Cette dernière caractéristique fait référence à la notion d'émergence qui est souvent abordée dans le cadre des systèmes complexes [Gilbert, 1995]. En effet, le comportement au niveau macroscopique du système est difficilement déductible à partir des comportements individuels des entités (observables à un niveau plus microscopique) qui composent le système.

D'autres propriétés sont parfois attribuées aux systèmes complexes dans la littérature [Cumenal, 2017] :

- plusieurs objectifs et sous-objectifs peuvent cohabiter au sein d'un même système, avec parfois des incompatibilités et des antagonismes entre eux ;
- des causalités entre événements (au sein du système) qu'il n'est pas toujours possible de déduire du fait de leur éloignement temporel ou de conséquences non intuitives de ces événements.

L'ensemble des propriétés précédemment énoncées rend les systèmes complexes difficiles :

- à comprendre du fait de la multiplicité des éléments qui les composent et de leurs interactions ;
- à analyser : l'observabilité du système n'est pas toujours garantie et lorsqu'elle est possible, l'ensemble des observations peut représenter une masse de données importante à traiter et à interpréter ;
- à modéliser : la création d'un modèle nécessite une construction abstraite et simplifiée du système complexe qui, là encore, est non triviale du fait de la multiplicité des éléments qui le composent ;
- à simuler : lorsqu'un modèle du système existe, sa simulation n'est pas toujours possible du fait des problèmes de passage à l'échelle pour des systèmes de grande taille ;
- à contrôler : un nombre d'éléments constitutifs relativement grand va induire pour le système un espace d'états possibles extrêmement large dans lequel la prise de décision n'est pas toujours aisée.

1.1.1.2 Systèmes complexes dynamiques et systèmes complexes distribués

Au sein des systèmes complexes, de nombreux travaux s'intéressent à leur évolution spatiale et temporelle. Cette évolution se caractérise par des changements d'états du système dus aux interactions entre les éléments composants le système. Celui-ci est alors qualifié de système dynamique.

Les systèmes dynamiques peuvent généralement s'appuyer sur les mathématiques pour décrire leur évolution au moyen d'un ensemble d'équations différentielles. Chaque élément composant le système peut par exemple être exprimé sous la forme d'une variable dont les valeurs évoluent au cours de la vie du système sous l'influence d'autres variables. Les systèmes les plus simples ont une dynamique linéaire : les évolutions subies par les variables du système sont exprimables sous une forme mathématique linéaire. A l'inverse, dans les systèmes dynamiques non linéaires (qui forment un sous-ensemble des systèmes complexes), il n'est pas possible d'exprimer une proportionnalité entre l'évolution perçue des variables et les causes de cette évolution.

Les entités composant un système complexe peuvent parfois être physiquement distribuées au sein de l'environnement dans lequel elles sont plongées. Ces systèmes complexes distribués sont parfois qualifiés de « large échelle » : la complexité¹ ici provient à la fois de la taille de l'environnement, du nombre et de la distribution des entités dans l'environnement et des relations entre ces entités.

Il s'agit ici de systèmes *naturellement* distribués et non d'une distribution *artificiellement* construite pour appliquer des approches de parallélisation. Une vision centralisée de ces systèmes n'est pas toujours possible et/ou souhaitable, obligeant ainsi à modéliser et/ou contrôler ces systèmes de manière distribuée.

1.1.1.3 Systèmes évolutifs

Les systèmes évolutifs considérés dans ce manuscrit sont des systèmes complexes. Ils sont donc à ce titre composés d'un grand nombre d'éléments en interaction pouvant être physiquement distribués au sein du système.

Le terme évolutif fait référence à la dynamique temporelle du système qui est elle-même susceptible d'évoluer dans le temps. Cette évolution peut être progressive et lente (i.e. sur une échelle de temps plus grande que celle servant à exprimer la dynamique du système); ou à l'inverse rapide et soudaine. Elle peut avoir des origines endogènes au système comme par exemple une ou plusieurs entités dont le comportement dérive au cours du temps et vient de ce fait remettre en cause toute la dynamique du système. Mais l'évolution peut être aussi causée par des facteurs exogènes avec des perturbations provenant de l'environnement extérieur au système.

L'évolution n'est pas forcément toujours subie par le système, elle peut être parfois voulue et initiée par le système lui-même pour justement répondre aux perturbations provenant de l'extérieur. On parle alors plutôt d'adaptation : le système va chercher à ajuster sa dynamique pour pouvoir continuer à atteindre des objectifs qui lui sont propres. Ces capacités d'adaptation reposent généralement sur l'anticipation. Malheureusement, la prédiction des états futurs d'un système évolutif est difficile notamment lorsque l'évolution provient de perturbations extérieures, souvent difficilement modélisables.

La figure 1.1 résume les principales caractéristiques des systèmes évolutifs considérés dans ce manuscrit.

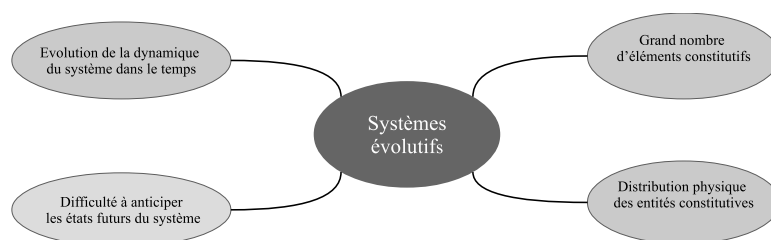


FIGURE 1.1 – Principales caractéristiques des systèmes évolutifs étudiés

Au vu des différentes caractéristiques que nous venons d'énoncer autour des systèmes complexes, dynamiques, distribués, évolutifs, le formalisme le plus adapté pour les modéliser et les exploiter nous semble être celui des systèmes multi-agents [Calo *et al.*, 2017].

1. nous parlons ici du sens commun et non de la complexité algorithmique

1.1.2 Formalisme multi-agent d'un système évolutif

1.1.2.1 Éléments d'un système multi-agent

Apparus dans les années 90, les systèmes multi-agents (SMA) sont aujourd'hui largement utilisés pour :

- le développement d'applications informatiques « intelligentes », ouvertes, distribuées et adaptatives ;
- la modélisation et la simulation de phénomènes complexes et sociaux ;
- la résolution décentralisée et collective de problèmes difficiles.

Un SMA peut se définir comme un ensemble d'entités autonomes plongées dans un environnement et interagissant entre elles dans le cadre d'une organisation structurant les fonctions de chacune des entités [Demazeau, 1995]. Au-delà de cette définition très générale, un certain nombre de caractéristiques sont généralement associées aux SMA :

- la tendance des agents à disposer d'objectifs, de compétences et de ressources qui leur sont propres ;
- la propension du système à se passer de contrôle global au profit d'une prise de décision répartie sur les agents ;
- la notion d'émergence qui peut être présente dans certains SMA lorsque des phénomènes macroscopiques, non initialement définis, apparaissent du fait d'interactions locales entre agents.

L'agent, composant principal d'un SMA, dispose d'un cycle de vie en trois phases : perception, décision et action. La phase de perception permet à l'agent d'acquérir un certain nombre d'informations provenant de l'environnement dans lequel il est plongé. La phase de décision correspond au raisonnement de l'agent lui permettant de choisir les actions qu'il devra réaliser. Enfin, la phase d'action est la réalisation effective des décisions préalablement établies. Les agents dits réactifs adopteront des comportements simples associant des actions spécifiques à une perception donnée. Dans ce manuscrit, nous considérerons plutôt des agents cognitifs et sociaux dont la phase de décision reposera sur des représentations explicites et symboliques de l'environnement permettant d'obtenir des comportements complexes.

L'environnement, seconde facette d'un SMA, joue un rôle essentiel puisqu'il fournit un espace commun à tous les agents leur permettant d'interagir. Plusieurs propriétés peuvent caractériser l'environnement [Russell et Norvig, 2010]. Lorsque la prise d'informations et l'exécution d'actions ne sont pas limitées en nombre dans le temps, l'environnement est qualifié de *continu* (et *discret* dans le cas contraire). Un environnement peut être qualifié de *dynamique* lorsque son état dépend des actions des agents, mais aussi de processus extérieurs au SMA comme par exemple une intervention humaine. Le qualificatif *statique* s'applique dès lors que le changement d'état n'est imputable qu'aux seules actions du système. D'une manière générale, lorsque ce changement d'état peut être complètement calculable à partir de son état courant et des effets induits par les actions des agents, l'environnement sera *déterministe*. L'impossibilité de prédire l'état futur à partir de l'état courant dénote un environnement *non déterministe*. Enfin, un environnement qui pourra être parfaitement observé par les agents sera dit *accessible*. Les systèmes évolutifs décrits précédemment et étudiés dans ce manuscrit seront plutôt plongés dans des environnements dynamiques, non déterministes et non accessibles.

La troisième facette d'un SMA concerne l'interaction. Elle se définit comme la relation dynamique instaurée entre plusieurs agents du fait de leurs actions combinées et réciproques. En fonction de l'application, elle peut prendre différentes formes : communication indirecte via l'environnement, dialogues entre agents, négociation, coordination, etc.

Enfin, l'organisation, dernière facette d'un SMA, définit l'agencement des relations entre les agents. Elle fournit un schéma d'interaction global garantissant un comportement cohérent du système. L'organisation peut être statique et définie lors de la création du SMA ; mais dans certains cas, les agents peuvent faire preuve d'adaptation et changer d'organisation durant la vie du système.

1.1.2.2 Formalisation d'un système évolutif

Dans un cadre général et sous l'angle des systèmes multi-agents, nous pouvons définir de manière un peu plus formelle un système évolutif comme la donnée d'un ensemble d'agents ag_1, \dots, ag_n évoluant et interagissant dans un environnement E :

- Chaque agent ag de l'ensemble A dispose d'un ensemble d'états S_{ag} et d'un ensemble d'actions réalisables A_{ag} . A chaque agent de A est associé un historique de ses états et actions dans le temps : $H_{ag} = \{(s, a)_0, \dots, (s', a')_t, (s'', a'')_{t+1}, \dots\}$ avec $s, s', s'' \in S_{ag}$ et $a, a', a'' \in A_{ag}$.
- Les états possibles de l'environnement E sont obtenus par le produit cartésien $S_{ag_1} \times \dots \times S_{ag_n} \times Ev$ où Ev représente un ensemble d'évènements disposant de probabilités d'apparition P_{Ev} totalement indépendantes des états et actions des agents et proches de zéro la plupart du temps.
- La dynamique temporelle de chaque agent ag peut s'exprimer comme la probabilité de passer d'un état $s \in S_{ag}$ à un état $s' \in S_{ag}$ en fonction des états et actions de l'ensemble des agents de A et des évènements $e \in Ev$: $P_{ag}(H_{ag_1}, \dots, H_{ag_n}, e)$. La dynamique temporelle de E est composée de l'ensemble des dynamiques individuelles de A .

L'ensemble du système peut subir une évolution dans le temps qui peut se traduire par :

- la suppression ou l'ajout d'états pour un ou plusieurs agents ;
- la suppression ou l'ajout d'actions pour un ou plusieurs états ;
- un changement des probabilités P_{ag} ;
- l'apparition ou la disparition d'évènements dans Ev ;
- un changement des probabilités P_{Ev} .

L'apparition et la disparition d'états, d'actions et d'évènements seront plutôt assimilées à des évolutions brusques et soudaines (par exemple, un robot perd la capacité à se déplacer suite à une défaillance technique). Les évolutions des probabilités P_{ag} et P_{Ev} pourront être soit soudaines, soit lentes et étalées dans le temps (par exemple, lorsque le système de localisation d'un robot n'est pas prévu pour se recalculer et perd progressivement en précision).

Dans cette définition, nous supposons que les évolutions peuvent être associées à des dates, nous permettant ainsi de considérer un temps discret et non continu. La prise en compte de l'humain et de son influence sur le système peut se faire à deux niveaux. Soit nous pouvons considérer que l'humain est un agent à part entière du système, soit le considérer comme externe au système ; auquel cas les effets de ses actions impactant le système peuvent être vus sous la forme d'évènements de Ev .

Dans la suite du document et fonction des applications, nous serons amenés à poser des hypothèses permettant de simplifier cette définition très générale.

1.1.3 Problématiques et thèmes de recherche

Dans ce manuscrit, nous nous intéresserons aux deux problématiques que sont :

- l'étude et la compréhension des systèmes évolutifs au travers leurs données et leur simulation,
- la conception de mécanismes de coordination d'agents dans un contexte évolutif.

1.1.3.1 Simulation dirigée par les données des systèmes évolutifs

Eu égard aux propriétés précédemment décrites, il est naturel d'envisager la simulation de systèmes évolutifs avec des approches comportementales à bases d'agents. Ces dernières se sont largement démocratisées ces 20 dernières années notamment grâce aux nombreux travaux de recherche autour des méthodologies et outils de simulation. On compte par exemple aujourd'hui plusieurs dizaines de plateformes permettant le développement de simulations multi-agents. La conception de simulation comportementale reste tout de même relativement difficile lorsqu'il s'agit de simuler des phénomènes sociaux évoluant dans le temps et impliquant des humains. Elle nécessite souvent le recours à un ou plusieurs experts disposant de connaissances sur le phénomène à simuler et capables d'élaborer les modèles de comportements à implémenter et à paramétrer. Cette tâche de modélisation est centrale dans le processus de construction d'une simulation : un modèle comportemental non adapté ou tout simplement mal paramétré conduira nécessairement à une simulation non réaliste.

Avec l'apparition ces dernières années des objets et capteurs connectés, il est de plus en plus facile de collecter de la donnée sur des activités humaines durant de longues périodes. Les capacités actuelles de stockage et de traitement permettent également d'analyser de grosses masses de données afin d'en extraire de la connaissance. Partant de ce constat, il est légitime de se demander si les techniques de fouille de données peuvent faciliter la construction de simulations comportementales.

Cette question est abordée au chapitre 2 au travers de travaux menés à l'interface entre les modélisations « agent » et les modélisations « dirigées par les données ». Ils explorent en particulier les apports des approches d'apprentissage non supervisé pour l'élaboration de modèles comportementaux d'agents.

1.1.3.2 Coordination multi-agent : approches génériques et prise en compte de l'incertain

La coordination désigne l'ensemble des processus qui permettent d'atteindre un état global souhaité tout en prenant en compte l'organisation du SMA, les états locaux de chaque agent et leur capacité à communiquer entre eux. Elle constitue un axe majeur de recherche dans le domaine des systèmes multi-agents. Durant la dernière décennie, de nombreux travaux se sont focalisés sur le développement de mécanismes de coordination basés soit sur des CSP (Constraint satisfaction problem) / COP (Constraint optimization problem) distribués (disCSP/DCOP), soit sur des processus de décision markoviens distribués. Comme le souligne Lesser et Corkill [Lesser et Corkill, 2014], ces approches ont permis d'apporter une meilleure analyse et compréhension de la complexité inhérente aux problèmes de coordination. Elles ont également permis d'étudier dans un cadre plus formel la notion d'optimalité des mécanismes de coordination. Ces deux approches présentent également l'avantage d'être générique. En effet, leurs modèles mathématiques sous-jacents offrent une certaine expressivité en termes de modélisation, ce qui facilite leur utilisation dans une large variété d'applications.

La problématique de coordination d'agents en contexte évolutif se focalise sur l'anticipation et/ou l'adaptation aux évolutions possibles du système listées en section 1.1.2.2. Dans ce contexte, les processus de décision markoviens distribués (MMDP, DEC-MDP, DEC-POMDP) sont particulièrement adaptés à la coordination en environnement incertain. Malheureusement, ces approches sont connues pour passer difficilement à l'échelle du fait de leur complexité calculatoire. A l'inverse, les approches basées sur les DisCSP / DCOP permettent d'envisager des coordinations d'un nombre significatif d'agents, mais n'intègrent pas nativement la notion d'incertitude dans leur modèle.

Le chapitre 3 étudie l'utilisation de ces deux approches pour des coordinations d'agents au sein de systèmes évolutifs. Les travaux présentés explorent individuellement chacune des approches, mais également l'hybridation entre les deux.

1.2 Illustrations et domaines d'application

Les deux thèmes de recherches précédemment introduits sont illustrés dans cette section au travers diverses applications. La section 1.2.1 présente une application orientée « commerce » en lien avec la conception d'agents dirigée par les données. Les trois sections suivantes illustrent la coordination d'agents en environnement dynamique et incertain au travers trois applications : l'exploration multi-robot, la production décentralisée d'électricité et la gestion de la ressource en eau dans les réseaux de voies navigables.

1.2.1 Modélisation des comportements humains au sein d'un espace de vente

Le secteur du commerce de détail et de la distribution est essentiel à l'économie d'un pays. En 2016, le chiffre d'affaires du secteur au niveau mondial atteignait 22 049 milliards de dollars. Depuis les années 2000, le secteur se divise en deux avec d'un côté les enseignes physiques traditionnelles et de l'autre le commerce électronique. Même si pour le moment, le chiffre d'affaires généré par le commerce électronique reste encore inférieur au commerce traditionnel (1 915 milliards de dollars au niveau mondial en 2016), de nombreux analystes prévoient un équilibrage des volumes de vente d'ici quelques années.

Dans ce contexte, les enseignes de magasins physiques sont amenées à mettre en place toute sorte d'actions de marketing et de merchandising permettant de dynamiser la surface de vente. Un exemple d'action est le ré-agencement de magasin. Une telle opération représente des coûts et des pertes non négligeables : achats de nouveaux linéaires, travaux de décorations, changement d'implantation des caisses, des allées, réduction de la surface de vente le temps des travaux et donc du chiffre d'affaires. L'impact du ré-agencement de l'enseigne physique doit donc être évalué au plus juste de manière à quantifier la hausse du chiffre d'affaires qui en découlera.

D'une manière générale, toutes actions marketing obligent les gestionnaires de magasin à mener des études sur le comportement de leurs clients. La simulation en tant qu'outil d'aide à la décision peut faciliter ces études. En particulier, la simulation peut être utilisée pour :

- visualiser de manière interactive la dynamique globale du magasin ;
- étudier les différents flux de clients à l'intérieur du magasin : flux aux entrées, formations de queues aux caisses, orientation des flux dans les différentes allées ;
- identifier les points chauds et les points froids (i.e. les zones les plus fréquentées et les zones les moins visitées du magasin) ;
- évaluer les impacts de différentes actions (ré-agencements de l'espace de vente, opérations publicitaires, promotions).

La simulation d'un magasin est un exemple typique de simulation sociale : la dynamique globale du magasin dépend des activités humaines qui s'y produisent. En particulier, les activités des clients sont centrales : leur déambulation dans les rayons, leurs interactions avec les vendeurs et les produits, leur passage en caisse. Les nombreuses études sociologiques et comportementales sur les clients et consommateurs montrent que le comportement de ces derniers est fortement évolutif dans le temps.

A l'échelle d'une année, les phénomènes de saisonnalité par exemple sont largement étudiés en marketing. Les segmentations de clients classiques comme la RFM² par exemple tiennent compte de l'évolution de la fréquence des achats dans le temps. Outre les aspects de saisonnalité, le comportement d'un client tend également à évoluer sur une échelle de temps plus réduite. Lors d'une visite en magasin par exemple, la probabilité d'achat diminue au fur et à mesure du temps (on parle de pression temporelle) [Lallement, 2008].

Dans la littérature, les simulations sociales sont classiquement abordées sous l'angle des simulations comportementales³ à base d'agents [Gilbert, 2008, Amblard et Phan, 2006]. La conception des comportements dans un cadre de simulation multi-agent est une tâche difficile qui requiert :

- de pouvoir observer individuellement les entités qui composent le système à simuler,
- de pouvoir analyser et interpréter ces observations,
- de trouver un modèle permettant de retranscrire la compréhension des phénomènes observés,
- de pouvoir implémenter informatiquement ce modèle,
- d'être en mesure de le paramétrer de manière à retranscrire les variations possibles du système,
- de valider le modèle utilisé en simulation.

Ces différents verrous font l'objet de travaux de recherche depuis plusieurs années. Différentes approches et méthodologies ont ainsi été proposées pour faciliter la conception de simulations comportementales à base d'agents.

Les travaux présentés en section 2.2 sont en lien avec le PICOM (Pôle de compétitivité des Industries du Commerce) et en partenariat avec le LAMIH de l'Université de Valenciennes. Ils ont été menés dans le cadre des thèses d'Imen Saffar (2013) et d'Antoine Sylvain (2016) et s'inscrivent dans une démarche récente qui vise à hybrider : modélisation dirigée par les données et simulation comportementale [Bera *et al.*, 2015, Zhang *et al.*, 2016, Singh *et al.*, 2018].

Les sections suivantes présentent les applications traitées dans le cadre du second axe de recherche sur la coordination d'agents en environnement dynamique et incertain.

1.2.2 Exploration multi-robot

L'un des intérêts de la robotique mobile est de pouvoir substituer des robots à des humains pour des missions qui pourraient potentiellement se révéler dangereuses. L'exploration d'une zone inconnue en est un parfait exemple. Les applications sont nombreuses : surveillance et reconnaissance en terrain hostile (zone de guerre, zone irradiée, etc), cartographie de régions difficiles d'accès (exploration sous-marine, planétaire).

Dans ce contexte, une exploration multi-robot (figure 1.2) présente de nombreux avantages par rapport à une approche mono-robot :

- utilisation de robots aux capacités complémentaires plutôt qu'un seul robot embarquant tous les senseurs et effecteurs utiles à la mission ;
- résilience aux pannes : une flotte de robots pourra toujours continuer la mission (éventuellement en mode dégradé) si un ou plusieurs robots deviennent inopérants ;
- temps d'exploration réduit : à capacité de déplacement identique, une flotte de robots permettra de couvrir une plus grande zone qu'un unique robot.

2. Méthode de classement des clients en fonction de 3 critères : la Récence (date du dernier achat), la Fréquence (des achats sur une période donnée), le Montant (le cumul des achats réalisés sur cette même période.)

3. également appelées individus-centrées

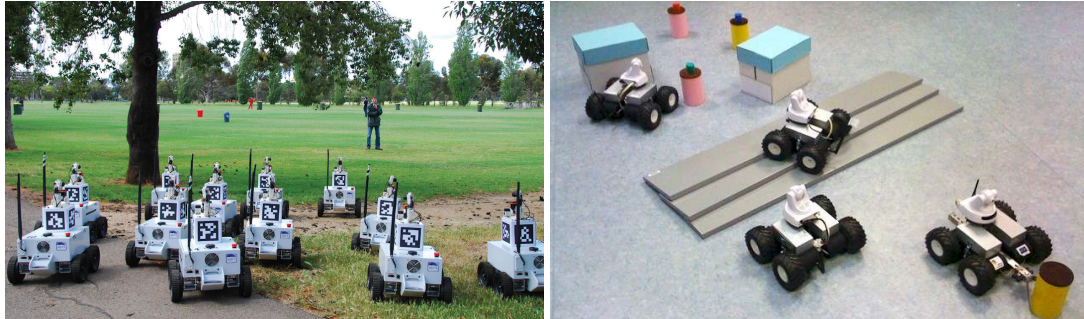


FIGURE 1.2 – Exemples de flottes de robots. A gauche, l'équipe de l'université du Michigan [Olson *et al.*, 2013]. A droite, flotte de Wifibots utilisée au sein de l'URIA

1.2.2.1 Exploration avec contraintes de communication

Une stratégie d'exploration efficace utilisant un groupe de robots consiste à faire déplacer les robots vers la frontière entre la zone inconnue et la zone déjà explorée. Ce principe, introduit par Yamauchi, [Yamauchi, 1998], a ensuite été décliné en plusieurs approches de coordination. Une première famille d'approches propose la définition de points de rendez-vous permettant aux robots de s'échanger les informations acquises à intervalles de temps réguliers [Fox *et al.*, 2000, de Hoog *et al.*, 2010].

Une seconde consiste à explorer la zone tout en maintenant un lien de communication entre tous les robots de la flotte. Classiquement, les communications entre les robots se basent sur un MANET (Mobile Ad hoc Network) [Perkins, 2001] : chaque robot est un nœud du réseau et est capable d'envoyer, de recevoir ou de relayer l'information. Le routage au sein des MANETs a fait l'objet de nombreux travaux [Abolhasan *et al.*, 2004, Royer et Toh, 1999], mais en faisant l'hypothèse que les déplacements des nœuds du MANET étaient totalement aléatoires. Or ce n'est pas forcément le cas dans un système multi-robot dans lequel chaque robot est autonome et planifie ses déplacements en fonction de ces objectifs et de ces contraintes physiques (capacité de franchissement, autonomie de sa batterie, etc). Un robot participant en tant que nœud à un MANET doit donc intégrer, en plus de ses contraintes de déplacement, des fonctions de routage de l'information.

1.2.2.2 Application à la robotique de sauvetage

Les travaux sur cette problématique présentés en section 3.1.2 ont été menés dans le cadre de la thèse de Tuan Van Le qui s'inscrit dans un contexte de gestion de catastrophes naturelles dans les zones urbaines (projet AROUND⁴). L'objectif est de fournir à des sauveteurs humains une aide sous la forme de robots mobiles capables :

- d'explorer une zone et de trouver des victimes,
- de prendre en temps réels des images de ces victimes,
- de cartographier l'emplacement des victimes,
- d'acheminer des vivres à ces victimes.

Dans ce contexte, une exploration multi-robot avec contrainte de connectivité évite la perte de

4. Autonomous Robots for Observation of Urban Networks after Disasters [Boucher *et al.*, 2009].

temps liée aux rendez-vous (même si elle limite la surface couverte à chaque instant de l'exploration par les robots puisque chaque robot doit rester à portée Wifi d'au moins un autre robot de la flotte). Elle permet également de maintenir un lien avec un opérateur humain qui peut alors accéder en temps réel aux découvertes faites par la flotte ou orienter l'exploration vers un point d'intérêt.

La coordination à mettre en place entre les robots pour réaliser cette tâche d'exploration doit concilier deux buts antagonistes : faire en sorte que la flotte couvre la plus grande zone géographique possible (de sorte à accélérer la découverte d'espaces non explorés) tout en garantissant le maintien de la connectivité au sein du MANET.

1.2.3 Production décentralisée d'électricité dans les centrales virtuelles

Historiquement, la gestion de l'électricité en France (de sa production à sa commercialisation) était assurée de manière centralisée : un seul opérateur (EDF) ayant une vision globale de la demande sur le territoire, contrôlant la production assurée par un nombre relativement restreint de centrales électriques (moins d'une trentaine) et gérant l'acheminement de l'électricité produite jusqu'au consommateur final.

Depuis une dizaine d'années, les réseaux électriques vivent une profonde mutation résultant d'une part de la libéralisation du marché de l'électricité et d'autre part du développement des énergies vertes. D'un point de vue organisationnel, l'ouverture à la concurrence a ainsi obligé l'état français à séparer les activités de production, de transport et de distribution de l'électricité (création de RTE⁵ en août 2005 afin de permettre l'arrivée sur le marché de nouveaux fournisseurs⁶ au côté de l'opérateur historique).

Conjointement à cette libération du marché, les pouvoirs publics ont encouragé l'introduction des énergies renouvelables dans le cadre de la lutte contre les émissions de gaz à effet de serre. Ces nouvelles formes de production d'énergie (essentiellement l'éolien et le photovoltaïque pour la France) ont été introduites progressivement représentant au début qu'une faible part de la production nationale. A ce titre, ces énergies vertes ont bénéficié (et bénéficie encore) de conditions dérogatoires d'accès au réseau : le gestionnaire du réseau (RTE) est tenu d'accepter la production électrique de ces sources dans les meilleures conditions tant qu'elle ne met pas en péril la sûreté du réseau. Plus récemment sont également apparues des velléités de pilotage de la demande au travers des nouveaux compteurs connectés⁷ venant ainsi complexifier un peu plus le paysage du marché électrique. Tout ceci laisse donc entrevoir une gestion de l'électricité (production, acheminement, consommation) qui va devoir être de plus en plus distribuée, s'éloignant ainsi du modèle de gestion historique.

1.2.3.1 Le concept de centrales virtuelles

La décentralisation de la production électrique verte pose un certain nombre de problèmes du fait de la nature des sources de production. D'une part, elles ont des capacités de production bien inférieures aux énergies traditionnelles (fossiles, nucléaires) : une éolienne seule fournit une puissance comprise entre 1 et 10 mégas watt alors qu'un seul cœur de centrale nucléaire fournit autour des 800

5. Réseau de transport d'électricité

6. En date du 1 mars 2018, on compte 26 offres alternatives à l'opérateur historique pour la fourniture électrique aux particuliers.

7. par exemple, les compteurs Linky en France

mégas watt. Ce rapport disproportionné entre les deux augmente la difficulté à coordonner l'effort de production entre énergies traditionnelles et énergies renouvelables.

D'autre part, les énergies renouvelables sont généralement intermittentes, car tributaires des conditions climatiques : par exemple, un parc éolien ne pourra pas produire s'il n'y a pas de vent, si l'orientation du vent n'est pas bonne ou s'il y a trop de vent (risque de casse mécanique pour l'éolienne). Les prévisions de production reposent donc sur des prédictions météorologiques qui peuvent être erronées obligeant à des ajustements en temps réels qui sollicitent fortement les sources de production classiques (nucléaire, charbon, gaz, etc).

Dans ce contexte, de nombreux chercheurs [Pudjianto *et al.*, 2007, Ruiz *et al.*, 2009, Saboori *et al.*, 2011] se sont penchés sur le concept de centrales virtuelles (*Virtual Power Plant* en anglais) qui consiste à agréger ensemble :

- des sources de production décentralisées et intermittentes (si possible diversifiées, par exemple des éoliennes et des panneaux photovoltaïques) ;
- des sources de production d'appoint contrôlables ;
- et des dispositifs de stockage de l'énergie.

L'objectif est de pouvoir considérer sur le réseau une centrale virtuelle multi-sources comme une centrale électrique classique :

- en atteignant une capacité de production critique, équivalente aux centrales classiques ;
- en s'engageant de manière certaine (i.e. quelque soit la précision des prévisions météorologiques) sur la production de puissance pour satisfaire une demande de consommateurs ;
- en s'acquittant, si besoin, des services système⁸.

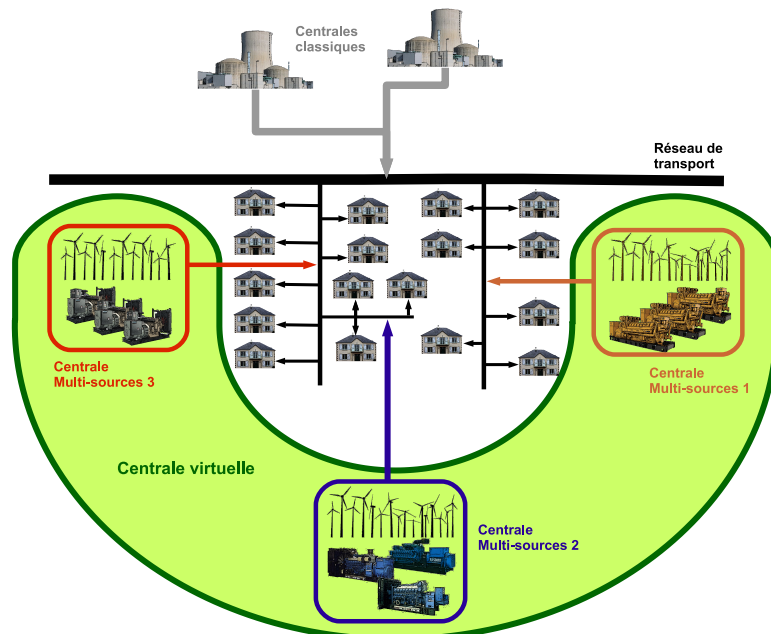


FIGURE 1.3 – Illustration du concept de centrale virtuelle

Pour ce faire, les dispositifs de stockage d'une centrale virtuelle permettent de compenser les er-

8. une partie de la puissance électrique produite est utilisée pour assurer la stabilité du réseau (en fréquence et en tension)

reurs de prédictions météorologiques qui engendrent des déficits de production. Le principe consiste à profiter des périodes où la capacité de production des sources intermittentes est excédentaire (au regard de la demande) pour emmagasiner de l'électricité. A l'heure actuelle, les dispositifs utilisables dans les centrales virtuelles sont essentiellement basés sur du stockage électrochimique sous forme de batteries. Celles-ci ont l'avantage d'être très réactives et de pouvoir libérer rapidement l'énergie emmagasinée. Malheureusement, en contrepartie, la quantité d'énergie pouvant être stockée reste faible. Des travaux cherchent d'ailleurs à utiliser des capacités de stockage non-dédiées comme par exemple les batteries des véhicules électriques [Mwasilu *et al.*, 2014, Blanc-Rouchossé *et al.*, 2018].

Le stockage ne peut donc compenser les écarts entre la demande et la production que sur de courtes périodes. Il est donc nécessaire, en complément du stockage, de venir adjoindre des sources de production contrôlables. Ces sources se présentent souvent sous la forme de groupes diesel ou de petites turbines à gaz. Fonctionnant avec des énergies fossiles, elles permettent de compenser les écarts de production sur de plus longues périodes et de recharger les batteries à un niveau minimal lorsque la production des sources intermittentes n'est pas suffisante.

1.2.3.2 Apport des systèmes multi-agents pour les centrales virtuelles

Il est possible d'adopter deux points de vue par rapport à une centrale virtuelle. Si l'on se place du point de vue du gestionnaire du réseau français (RTE), une centrale virtuelle n'est finalement qu'un nœud de plus dans le réseau de production. A ce niveau, c'est l'arrivée massive de centrales virtuelles qui posera problème dans le futur : le gestionnaire sera alors confronté à une augmentation importante des points de production sur le réseau. Une gestion centralisée telle qu'elle est faite actuellement posera des problèmes :

- de centralisation des données nécessaires à la coordination et la supervision de la production ;
- de passage à l'échelle lors de la planification des consignes assignées à chaque centrale ;
- de résilience en cas de problèmes de communication avec les différents points de production sur le réseau ;
- de gestion de crise et de mise en place d'opérations de délestage⁹ pour l'anticipation et le traitement des black-out.

Si l'on se place maintenant du point de vue du gestionnaire d'une centrale virtuelle, on retrouve des problématiques assez similaires. En effet, une centrale virtuelle peut potentiellement être constituée d'un grand nombre de sources de productions (intermittentes ou contrôlables) engendrant les mêmes problèmes de centralisation des données, de planification, de résilience, de gestion des incertitudes et des cas extrêmes. La centrale virtuelle apparaît donc ici clairement comme un système distribué (production décentralisée) évolutif (soumis aux aléas de production : erreurs de prédiction, panne et maintenance préventive des sources).

Dans ce manuscrit, nous nous intéressons à l'apport des systèmes multi-agents pour la gestion décentralisée des centrales virtuelles multi-sources. Les premiers travaux sur l'utilisation des SMA dans le domaine des réseaux électriques intelligents datent des années 2000. Ils se sont intéressés à différentes problématiques telles que la surveillance et le diagnostic [McArthur *et al.*, 2004], le contrôle distribué [Dimeas et Hatziargyriou, 2005], [Huang *et al.*, 2009], la gestion des incidents [Nagata et Sasaki, 2002], la modélisation [Hopkinson *et al.*, 2006] ou encore la simulation du réseau [Davidson *et al.*, 2008]. [Roche *et al.*, 2010] et [Rehtanz, 2003] donnent un aperçu global des applications possibles des systèmes multi-agents dans les réseaux électriques. Une des solu-

9. effacement momentané de certaines charges sur le réseau

tions les plus abouties de gestion de centrales virtuelles à base d'agents est Powermatcher (<http://flexiblepower.github.io>) développé dans le cadre des projets européens CRISP et INTEGRAL. Cette solution adopte une structure de type marché et se focalise uniquement sur les aspects économiques liés à l'exploitation d'une centrale virtuelle. Les sources locales de production sont représentées par des agents qui doivent se coordonner par un système d'enchères pour proposer une production globale en fonction d'un prix de marché de l'électricité. Quinze ans après ces premiers travaux, l'usage des systèmes multi-agents dans les centrales multi-sources semble aujourd'hui bien implanté [Clausen *et al.*, 2017, Adu-Kankam et Camarinha-Matos, 2018, Skarvelis-Kazakos *et al.*, 2016, Feron et Monti, 2017].

Pour notre part, nous nous sommes intéressés aux contraintes techniques des centrales virtuelles multi-sources. En particulier, nous nous sommes focalisés sur la coordination des sources de production contrôlables prenant en compte les aléas de production provenant des sources intermittentes ainsi que les potentielles défaillances des unités de production. La contribution présentée en section 3.2.1 porte sur une résolution multi-agent du problème d'engagement de sources de production (UCP¹⁰). Ce problème consiste à planifier la production de chaque source de manière à satisfaire une demande en électricité pré-établie sur un horizon temporel (en règle générale 24h). Les contraintes techniques propres aux sources de production ne permettent pas de répondre instantanément et dans n'importe quelle condition à cette demande (par exemple, pour des groupes diesel, il y a des temps de préchauffage et de refroidissement à respecter). L'UCP consiste donc à définir à moindre coût : (i) l'ordonnement d'allumage / d'extinction de chaque source sur la période considérée et (ii) la production de chaque source pour chaque période de temps où elle est en marche de manière à satisfaire la demande.

Ces travaux sur l'UCP ont été menés dans le cadre du projet AgentVPP. Ce projet financé par l'ADEME s'est déroulé de 2010 à 2012 et a impliqué trois partenaires académiques : le Laboratoire d'électrotechnique et d'électronique de puissance de Lille (L2EP), le Laboratoire en Génie Electrique de Grenoble (G2ELAB) et l'URIA, ainsi qu'un partenaire industriel : Maïa Eolis.

1.2.4 Gestion de la ressource en eau pour les voies navigables

Le transport fluvial de marchandises connaît un déclin depuis les années 70. Il représentait à l'époque 14 milliards de tonnes-kilomètres¹¹ pour ne plus représenter que 7,5 milliards en 2015¹². Cependant, les politiques récentes de transport de marchandises souhaitent remettre le transport fluvial sur le devant de la scène. Celui-ci apparait en effet comme une des alternatives possibles au transport routier. Pour certaines cargaisons (pondéreux, conteneurs, matières dangereuses), le transport fluvial se révèle même être une bien meilleure option que le transport routier ou ferroviaire. D'une manière générale, le transport fluvial peut mettre en avant des qualités particulièrement appréciables dans un contexte de développement durable :

- réseau fluvial peu sujet à la saturation garantissant une fiabilité du temps de transport ;
- un taux d'accidentologie proche de zéro garantissant une certaine sécurité ;
- des rejets et une consommation en carburant relativement faibles au regard du tonnage transporté ;
- existence de ports fluviaux connectés aux réseaux ferroviaires et routiers pouvant ainsi servir

10. Unit Commitment Problem

11. unité correspondant au déplacement d'une tonne de marchandises sur un kilomètre

12. https://www.statistiques.developpement-durable.gouv.fr/sites/default/files/2018-11/tome2_cctn2007_publie_2008_modif_cle7bf42e-1.pdf

de nœuds d'échange multimodal.

Les plans de développement pour le transport fluvial pourraient cependant être entravés dans les années à venir par les phénomènes liés aux changements climatiques. De nombreuses études prévoient des augmentations significatives des températures, mais également des impacts importants sur les précipitations. Plusieurs travaux de recherche envisagent un accroissement du nombre de phénomènes extrêmes liés aux pluies : hausse en fréquence et en intensité des crues et des épisodes de sécheresse venant impacter les réseaux hydrographiques [Beuthe *et al.*, 2012, Boé *et al.*, 2009].

1.2.4.1 Réseau de voies navigables : principe et fonctionnement

D'un point de vue topologique, un réseau de voies navigables est composé de cours d'eau naturels et de canaux artificiels. Pour rendre possible la navigation sur ces voies d'eau qui peuvent présenter des déclivités et donc des niveaux d'eau non homogènes, un certain nombre d'ouvrages hydrauliques tels que les écluses et les ascenseurs à bateaux équipent le réseau. Ces ouvrages permettant de faire passer des bateaux d'un niveau en amont vers un niveau plus bas en aval viennent segmenter le réseau en tronçons de niveau d'eau uniforme appelés bief.

Le réseau hydraulique ainsi décomposé en biefs est un système potentiellement de grande taille. Si l'on se place uniquement en région des Hauts-de-France, l'opérateur VNF¹³ (en charge de la gestion des voies navigables sur le territoire) doit gérer pas moins de 88 biefs (cf. figure 1.4).

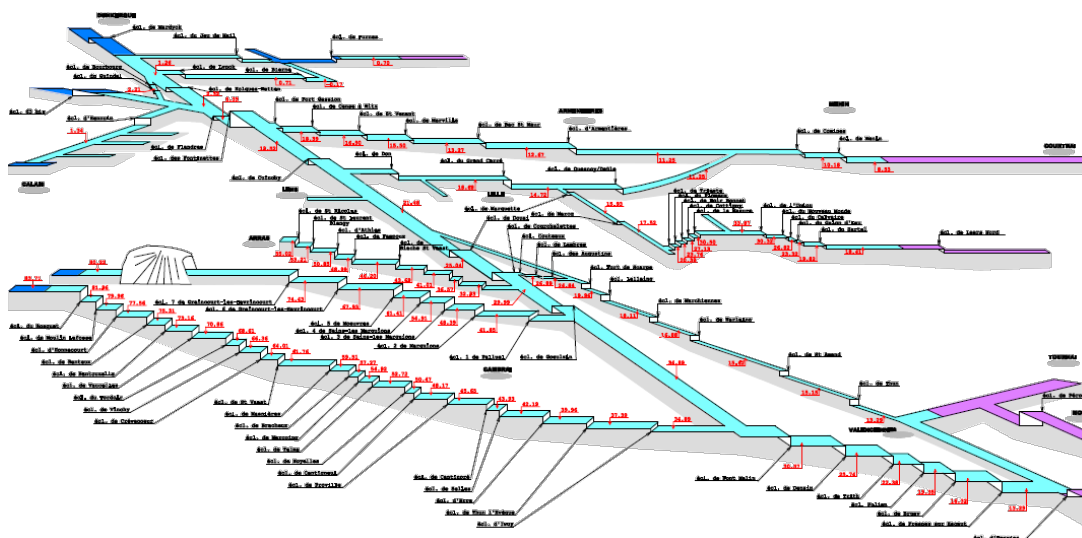


FIGURE 1.4 – Schéma hydrologique du réseau des voies navigables en région Hauts-de-France

Sur chaque bief, VNF se doit d'assurer un service de navigation répondant à des contraintes techniques de déplacements des bateaux. Ces derniers évoluent dans un rectangle de navigation (voir figure 1.5) dont la base est formée par le chenal de navigation. La largeur du chenal est déterminée non pas en fonction de la largeur du canal, mais en fonction de la hauteur d'eau restante sous coque du bateau lorsque celui-ci est au maximum de sa charge. Le rectangle de navigation est également dimensionné

13. Voies Navigables de France

par rapport à la hauteur libre sous les ponts et souterrains de manière à garantir une garde suffisante pour le passage du bateau.

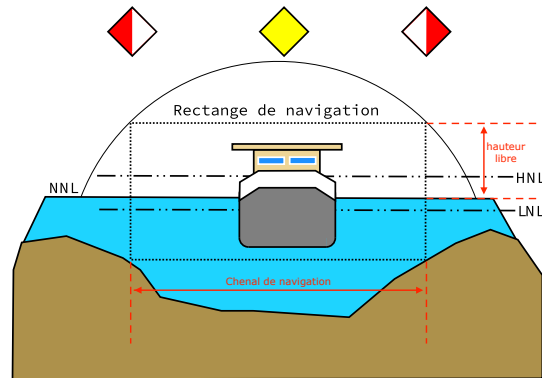


FIGURE 1.5 – Rectangle de navigation associé à un bief

En fonction du tonnage autorisé pour les bateaux naviguant sur le bief, deux limites sont associées au rectangle de navigation : le niveau haut de navigation (Highest Navigation Level - HNL) et le niveau bas de navigation (Lowest Navigation Level - LNL) qui correspondent respectivement à des hauteurs d'eau maximales et minimales à ne pas dépasser sur le bief de manière à ce que le bateau reste inscrit dans le rectangle de navigation. Idéalement, le gestionnaire vise à conserver sur l'ensemble du réseau un niveau d'eau égale au NNL (Normal Navigation Level).

1.2.4.2 Gestion de la ressource en eau : pratiques actuelles et évolutions futures

Le franchissement d'une écluse par un bateau s'accompagne d'un transfert d'eau au bief amont vers le bief aval. Le volume relâché, appelé bassinée, correspond au volume d'eau emprisonné entre les deux portes lors du remplissage du sas (figure 1.6).

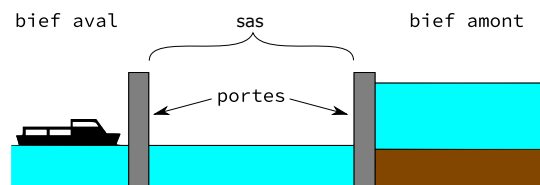


FIGURE 1.6 – Ecluse

Suivant la densité de trafic sur le réseau, la navigation fluviale peut donc être très consommatrice en eau et elle constitue la première source de perturbation dans le réseau. D'autres perturbations peuvent venir s'ajouter :

- les échanges avec les nappes phréatiques (plus ou moins négligeables suivant l'étanchéité du canal),
- le déversement des rivières naturelles,
- les rejets sauvages (des particuliers ou des industriels),
- les pompes.

Enfin, viennent s'ajouter les évènements météorologiques (périodes de pluies, sécheresse) dont certains peuvent être extrêmes et impacter très fortement les niveaux d'eau dans les biefs (parfois jusqu'au débordement de celui-ci : figure 1.7).



FIGURE 1.7 – Inondation de berges suite au débordement du bief Douai-Don-Cuinchy (juin 2016)

Pour s'approcher au plus près du NNL (ou à défaut rester dans l'intervalle [LNL, HNL]) sur l'ensemble du réseau, le gestionnaire VNF dispose d'un ensemble d'ouvrages d'art (portes, barrages, pompes de relevage, bassins de rétention) lui permettant de déplacer et de stocker des volumes d'eau. Actuellement, la régulation du niveau d'eau est basée sur une expertise humaine construite au fil des années en fonction d'observations locales et manuelles.

Récemment, VNF a entrepris l'instrumentation d'un certain nombre de biefs : des capteurs permettant des mesures des niveaux d'eau et des débits à différents points ainsi que des automates permettant de piloter informatiquement des ouvrages. Dans ce contexte, il devient envisageable de construire des stratégies de gestion optimale de la ressource en eau basée sur des approches de l'IA. Ainsi, dans [Thang Pham *et al.*, 2014], les auteurs présentent de premiers travaux combinant une approche MPC (Model Predictive Control [Garcia *et al.*, 1989]) et les systèmes multi-agents pour la mise en place d'un contrôle distribué de canaux d'irrigation.

Les travaux qui seront présentés en section 3.1.1 ont été initiés dans le cadre du projet Gepet-Eau. Ce projet, financé par le GICC (Gestion et Impacts du Changement Climatique), vise à développer de nouvelles stratégies de gestion de l'eau dans le contexte de trafic fluvial croissant et d'augmentation des phénomènes climatiques extrêmes. Il a impliqué deux partenaires académiques : IRSTEA et l'URIA ainsi que VNF et l'agence de l'eau Artois - Picardie. Ces travaux ont ensuite été approfondis dans le cadre de la thèse de G. Desquesnes (section 3.2.2).

1.3 Conclusion

Nous avons présenté dans ce chapitre notre vision d'un système évolutif : un système composé d'un grand nombre d'entités en interaction (système complexe) et dont la dynamique temporelle évolue elle-même dans le temps.

De par leur nature et leurs caractéristiques, nous avons pris le parti d'aborder ces systèmes évolu-

tifs comme des systèmes multi-agents. En particulier, nous avons dégagé deux problématiques :

- l'étude et la compréhension des systèmes évolutifs aux travers leurs données et simulation,
- la conception de mécanismes de coordination dans un contexte évolutif.

Les travaux menés sur ces deux problématiques sont respectivement présentés dans les chapitres 3 et 4 de ce document.

La dernière partie de ce chapitre a présenté les domaines d'applications de ces travaux. Les systèmes évolutifs considérés y sont de diverses natures : systèmes artificiels (flotte de robots et centrale virtuelle de production électrique), systèmes humains (magasin), systèmes naturels (réseaux hydrographiques).

Chapitre 2

Conception de simulations multi-agents dirigée par les données

Nous avons évoqué au chapitre 1 les systèmes évolutifs impliquant des comportements humains qui, par nature, peuvent être extrêmement changeants dans le temps. La simulation de ces systèmes implique souvent la reproduction de phénomènes sociaux résultant d'interactions humaines. De ce cas, les approches de simulation comportementale à base d'agents sont souvent préférées aux approches de simulation classique [Amblard, 2010].

Introduites il y a une trentaine d'années, les simulations comportementales à base d'agents ne sont plus cantonnées aux prototypes de recherche développés en laboratoire et sont aujourd'hui largement répandues. Des plateformes facilitant leur développement, telles que Netlogo¹ ou Anylogic², ont largement contribué à leur démocratisation dans des domaines variés : transport [Merkuryeva et Bolshakovs, 2010], logistique [Barbosa et Leitão, 2011, Yang et Chen, 2018], santé [Djanatliev et German, 2013], etc.

Dans ce chapitre, nous abordons la problématique de modélisation d'agents dans les simulations comportementales. La tâche de modélisation est fondamentale pour l'obtention d'une simulation réaliste et reste délicate pour de nombreuses applications, notamment celles impliquant des comportements humains [Proctor et Zandt, 2018] et des phénomènes sociaux complexes.

Dans une première section, nous ferons un état de l'art sur la modélisation agent, la modélisation dirigée par les données et les synergies entre les deux. Nous introduirons ensuite une approche dirigée par les données pour « l'agentification » au sein de simulations comportementales. Enfin, nous présenterons une application à la simulation de magasins.

2.1 Simulations multi-agents et fouille de données : quelles synergies ?

Les modèles utilisés en simulation sont une représentation abstraite et simplifiée du système réel à simuler qui permet de décrire et reproduire sa dynamique. Pour des systèmes réels, gouvernés par des lois physiques, les modèles reposent souvent sur des outils mathématiques comme par exemple

1. <https://ccl.northwestern.edu/netlogo/>

2. <https://www.anylogic.com>

les équations différentielles. L'usage d'équations est beaucoup moins évident lorsque la dynamique du système réel repose sur des comportements humains, des interactions sociales et nécessite la prise en compte d'aspects psychologiques. Les simulations comportementales à base d'agents se révèlent alors beaucoup plus appropriées.

2.1.1 Conception de simulations multi-agents

2.1.1.1 Généralités

Les simulations multi-agents consistent en un ensemble d'entités individuelles en interaction ; chacune d'entre elles modélisant une partie du système à simuler. Ces entités sont des agents logiciels évoluant dans un environnement, exécutant de manière autonome un comportement approprié et motivé par des croyances et motivations propres. A chaque pas de simulation, l'agent perçoit l'environnement simulé, décide quelle action il exécutera au prochain pas de temps et agira en conséquence.

L'élaboration d'une simulation comportementale peut comprendre deux phases : une phase de *construction* initiale et une phase de *révision* et d'*adaptation*. Ces deux phases peuvent être opérées suivant deux approches. Une première approche dite *ascendante* qui consiste à spécifier les états et le comportement global (attendus) de la simulation (niveau macroscopique) pour ensuite les décliner au niveau de chaque agent (niveau microscopique). L'instanciation des comportements individuels doit souvent être revue et adaptée empiriquement pour obtenir le comportement global voulu. La seconde approche dite *descendante* part du niveau microscopique en spécifiant les comportements locaux de chaque agent : le comportement global de la simulation au niveau macroscopique est obtenu par émergence des interactions entre les agents. Là encore la phase de révision est nécessaire pour ajuster les comportements locaux des agents de manière à converger vers le comportement global attendu.

La figure 2.1 illustre le positionnement de ces deux phases dans le processus global d'élaboration d'une simulation. Celui-ci comprend cinq étapes.

- Etape 1 : l'observation du monde réel durant laquelle le concepteur va pouvoir collecter un certain nombre de données décrivant le ou les phénomènes à simuler.
- Etape 2 : l'abstraction et la modélisation du monde réel qui s'appuient sur l'étape 1 pour produire des modèles des phénomènes à reproduire. Les premiers modèles sont généralement formulés en langage naturel et sont successivement raffinés pour aboutir à un modèle implémentable.
- Etape 3 : l'implémentation du modèle obtenu à l'étape précédente (les trois premières étapes correspondent à la *construction* initiale du modèle).
- Etape 4 : l'exécution de la simulation résultante. Cette étape s'accompagne d'une calibration des différents paramètres de la simulation.
- Etape 5 : la validation de la simulation qui s'opère généralement en comparant des données issues de la simulation et des données réelles.

Lorsque l'étape 5 n'est pas concluante, le concepteur de la simulation doit *réviser* et *adapter* la simulation en remettant en cause soit l'implémentation du modèle, soit la modélisation elle-même.³

L'étape de modélisation se révèle être la plus délicate du processus. En effet, elle requiert de la part du concepteur de répondre à un certain nombre de questions ; dont celles présentées ci-dessous :

3. Notons qu'en pratique des retours en arrière entre les étapes peuvent également s'opérer. Par exemple, si l'exécution de la simulation entraîne des plantages dus à des erreurs d'implémentation, le concepteur repasse par l'étape 4 avant de tenter toute validation.

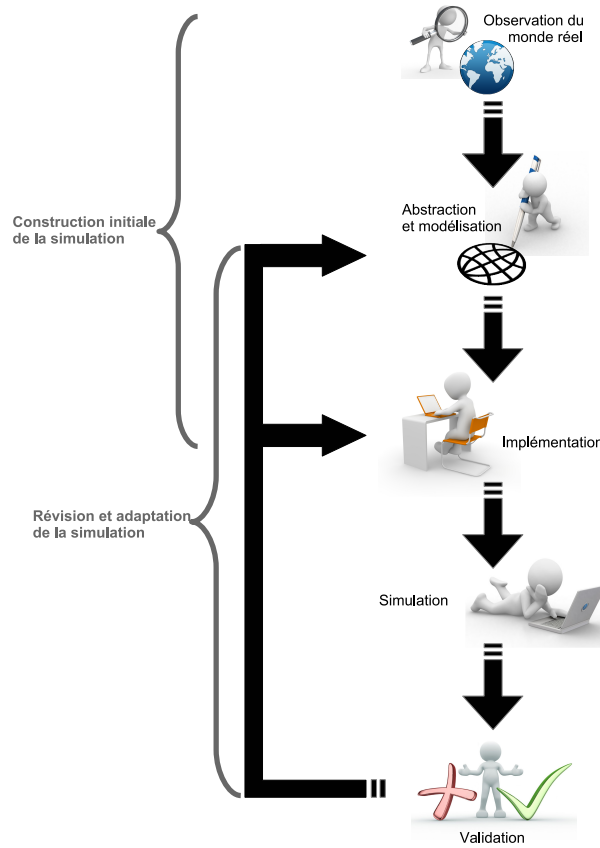


FIGURE 2.1 – Processus de construction d’une simulation (adapté de [Shannon, 1998] et [Galán *et al.*, 2009])

— *Quelles sont les entités à simuler ?*

Répondre à cette question nécessite dans un premier temps d’identifier les entités qui participent au phénomène à simuler et d’en dégager éventuellement des grandes catégories. Par exemple, pour une simulation de trafic routier, parmi les entités qui participent aux phénomènes de trafic, il y a les usagers de la route. Dans ces usagers, on distingue : les piétons, les conducteurs de deux-roues, les automobilistes qui eux-mêmes peuvent se décliner en sous-catégories. Dans un second temps, il convient de sélectionner sur l’ensemble des entités celles qui devront être modélisées : en effet, par souci d’abstraction et de simplification, certaines entités peuvent être négligées du fait de leur contribution minime aux phénomènes réels observés (principe de parcimonie). Par exemple, est-ce utile de modéliser des convois exceptionnels dans une simulation de trafic d’une ville qui n’est traversée qu’une à deux fois par an par ce type de camions ?

— *Quelles sont les caractéristiques de ces entités ?*

L’identification des grandes catégories d’entités à simuler peut se baser entre autres sur leurs caractéristiques apparentes. Par exemple, dans une simulation de trafic routier, l’âge peut être une caractéristique permettant de créer des sous-groupes de piétons.

— *Quelles sont les interactions entre ces entités ?*

Pour chaque entité retenue et devant être modélisée, il est nécessaire de comprendre comment elle interagit avec son environnement et avec d'autres entités. Cette étape peut se révéler difficile : souvent l'observation du monde réel permet uniquement de constater le résultat de ces interactions.

— *Comment les entités se comportent-elles ?*

Les interactions que peuvent avoir les entités font partie de leur comportement. Celui-ci désigne l'ensemble des processus cognitifs (perception, prise de décision, apprentissage, etc) qui permettent aux entités de choisir leurs actions.

— *Est-ce que la dynamique du système observé est stable dans le temps ?*

En observant des phénomènes à différents moments et sur différentes durées, il est possible de détecter des variations de dynamique (c'est le cas des systèmes évolutifs). Suivant les objectifs de la simulation, il est nécessaire ou non d'en tenir compte.

— *Comment distribuer les agents dans la simulation ?*

Lorsque chaque entité a été modélisée sous forme d'un agent avec un comportement et des caractéristiques qui lui sont propres, il reste à déterminer : la composition de la population d'agents qui sera créée pour la simulation (c'est-à-dire la représentativité de chaque typologie d'agents), la séquence de création de cette population (c'est-à-dire à quelle fréquence seront créés les agents ? dans quel ordre ?), le paramétrage de chaque typologie d'agents.

Ces dernières années, de nombreux travaux de recherche se sont intéressés à faciliter la conception d'une simulation multi-agent. Nous pouvons distinguer dans la littérature trois types de travaux :

- des travaux sur les plateformes et langages dédiés au développement de systèmes multi-agents ;
- des travaux sur les langages de modélisation ;
- des travaux sur les méthodologies de conception et l'ingénierie dirigée par les modèles.

Notons que certains travaux proposent à la fois des modèles formels d'agents associés à une méthodologie de conception, et le tout outillé par une plateforme de développement. Dans la suite, nous illustrons chaque type par un ou plusieurs exemples.

2.1.1.2 Langage et plateforme de développement d'agents

Une simulation multi-agent peut bien sûr être implémentée à partir de zéro en utilisant un langage de programmation classique tel que C, C++, Java, etc. Cependant, la tendance actuelle est plutôt à l'utilisation de plateforme de développement spécifique. On en distingue deux types : des outils généraux de conception de systèmes multi-agents comme par exemple la plateforme JADE [Bellifemine *et al.*, 2007] ou le langage SARL [Rodriguez *et al.*, 2015] et des outils de conception de simulation comportements à base d'agents avec par exemple les plateformes Repast [North *et al.*, 2013], Netlogo [Tisue et Wilensky, 2004], JaSim [Guiot *et al.*, 2014], Gama [Grignard *et al.*, 2013].

Bien que pouvant reposer sur des spécifications et des choix techniques différents, ces plateformes et langages fournissent la plupart du temps : des services de nommage et de communication entre agents, une plateforme d'exécution, un ordonnanceur, une définition abstraite des agents (sous forme de classes, de composants, etc). Les plateformes « orientées simulation » apportent généralement en plus une gestion graphique d'un environnement pouvant aller de la simple grille 2D à l'environnement 3D complexe en passant par le GIS (Geographic Information System). Pour une présentation plus détaillée et exhaustive de ces plateformes, le lecteur pourra se référer à [Abar *et al.*, 2017].

L'usage de ces plateformes et notamment celles « orientées simulation » facilite grandement la

tâche d'implémentation présentée en figure 2.1. L'introduction de langages dédiés pour la spécification des agents (comme par exemple dans les plateformes Netlogo ou Gama) simplifie l'écriture de codes complexes comparativement à des paradigmes de programmation plus classiques. En revanche, comme ces plateformes n'imposent aucun cadre formel ni d'architecture interne à l'agent, le concepteur n'est pas guidé dans sa tâche de modélisation en amont de l'implémentation.

2.1.1.3 Langage de modélisation

Pour éviter de passer directement d'un modèle « fait sur papier » à son implémentation dans une plateforme, il existe des langages de modélisation permettant de définir des modèles d'agents dans un formalisme de haut niveau avec une sémantique adaptée à la philosophie multi-agent. Nous pouvons citer par exemple : AOP (Agent-oriented programming) [Shoham, 1993] Agent UML (extension d'UML) [Bauer *et al.*, 2001], Agent Modeling Language (AML) [Cervenka et Trencansky, 2007], Agent-Object-Relationship Modeling [Wagner, 2003]. Tous ces langages offrent un niveau d'abstraction supérieur à un langage de programmation classique et permettent de manipuler plus facilement les différents concepts inhérents aux SMA tels que les notions de groupe, de rôle, d'organisations par exemple.

Les langages de modélisation peuvent être intégrés à des méthodologies utilisées dans une démarche de conception dirigée par les modèles.

2.1.1.4 Méthodologie de conception et ingénierie des modèles

Inspirées des processus de développement en génie logiciel, de nombreuses méthodologies de conception de SMA ont été proposées : PASSI [Cossentino, 2005], GAIA [Wooldridge *et al.*, 2000], ADELFE [Bernon *et al.*, 2003], INGENIAS [Pavón et Gómez-Sanz, 2003] ou encore O-Mase [Garcia-Ojeda *et al.*, 2008]. Parmi celles-ci certaines peuvent être facilement spécialisées pour la conception de simulations multi-agents : citons par exemple INGENIAS [Pavon *et al.*, 2008], Gaia [Streltchenko *et al.*, 2005], PASSIM [Cossentino *et al.*, 2008]), ASPECS [Cossentino *et al.*, 2010]. D'autres, comme IODA [Kubera *et al.*, 2011] ont directement été pensées pour la simulation comportementale.

Ces méthodologies sont parfois associées à des approches d'ingénierie dirigée par les modèles. Ceux-ci peuvent fixer une architecture pour l'agent et orienter la conception suivant différentes vues (l'organisation, les interactions, les agents, l'environnement) (comme par exemple dans INGENIAS). Ces vues peuvent se matérialiser sous la forme de représentations graphiques (UML par exemple) et certaines approches vont même jusqu'à proposer une génération automatique de code pour ces vues.

Toutes ces approches sont des avancées majeures dans le domaine de la simulation comportementale. Elles permettent de guider le concepteur en lui imposant par exemple un point de vue (la simulation sous l'angle de l'interaction avec IODA ou de la coopération avec ADELFE) au travers duquel il peut commencer à élaborer des premières abstractions du ou des phénomènes à simuler. L'expertise humaine mobilisée par le concepteur varie suivant les approches. Dans la méthode ADELFE par exemple, l'expertise requise porte sur l'identification des cas non coopératifs.

Dans la suite de ce chapitre, nous étudierons l'apport possible des méthodes d'apprentissage et de fouille de données pour aider le concepteur en amont des méthodologies précédemment citées. L'objectif de cette étude est d'évaluer si la fouille de données permet de répondre aux questions listées en section 2.1.1.1. Dans la sous-section qui suit, nous ne présenterons pas les approches d'apprentissage

dites par renforcement. Même si elles ont déjà été utilisées dans le cadre de simulations [Klein, 2009], elles restent cependant plus appropriées à des problématiques de contrôle. Par ailleurs, lorsqu'il s'agit de reproduire des comportements humains, le cadre markovien qui interdit la prise en compte de décisions passées est un peu trop restrictif. Nous nous focalisons donc dans la suite sur des travaux qui ont tenté de construire des synergies entre les systèmes multi-agents et la fouille de données. Ces travaux sont présentés après un survol rapide des méthodes utilisées en fouille de données.

2.1.2 Apprentissage et fouille de données

La fouille de données peut se définir comme l'extraction automatique de connaissances cachées au sein d'un ensemble de données potentiellement volumineux. Ses connaissances cachées peuvent prendre différentes formes : des tendances, des relations entre variables, des répétitions de motifs. Les techniques d'apprentissage utilisées pour la fouille de données peuvent se scinder en deux catégories : les méthodes d'apprentissage supervisé et non supervisé. Les présentations de ces deux familles seront succinctes et se contenteront d'introduire les briques utiles à notre proposition de conception d'agents dirigée par les données. Le lecteur intéressé pourra se référer aux ouvrages suivants pour un état de l'art plus complet et exhaustif : [Tufféry, 2005, Tan *et al.*, 2006].

2.1.2.1 Apprentissage supervisé

Ces méthodes consistent à apprendre, à partir d'une base d'exemples, une fonction de prédiction pouvant servir à estimer la valeur d'une variable dans un ensemble continu et infini (problème de régression) ou à attribuer à une variable une valeur (étiquette) prise dans un ensemble discret et fini (problème de classification). L'apprentissage supervisé repose donc sur une base de données d'apprentissage qui contient des exemples annotés par un expert et suffisamment représentatifs du problème à traiter.

Les problèmes de classification peuvent se rencontrer dans le cadre d'une modélisation à base d'agents. Supposons que le concepteur d'une simulation ait préalablement identifié plusieurs types d'agents et souhaite les paramétrer à partir de données réelles. Il peut alors être intéressant pour lui de classer ces données par typologie d'agent de manière à se faciliter l'extraction des valeurs de paramètres à injecter dans ses modèles d'agents. En ce sens, l'apprentissage supervisé peut être vu comme une manière de généraliser des données d'entrées inconnues.

Les algorithmes d'apprentissage supervisé sont nombreux et se répartissent en quatre grandes familles. La première regroupe les algorithmes basés sur la logique classique dont l'objectif est de construire des règles logiques liant les attributs des différents exemples de l'ensemble d'apprentissage. Par exemple, les arbres de décision [Quinlan, 1986] sont obtenus en regroupant sur une même feuille de l'arbre les exemples présentant des caractéristiques communes. La lecture de l'arbre fait apparaître des règles du type : *Si attribut1 = ... Et attribut2 = ... Et ... Alors Etiquette de la classe.*

La seconde famille regroupe les approches utilisant le concept de neurone artificiel [Haykin *et al.*, 2009]. Dans un cadre supervisé, l'apprentissage d'un réseau de neurones utilise comme données des couples (entrée, sortie) pour adapter le poids des connexions entre les neurones de manière à faire correspondre au mieux chaque entrée avec la sortie attendue.

Une troisième famille rassemble les approches utilisant les statistiques et les probabilités. On y trouve par exemple la classification bayésienne [Domingos et Pazzani, 1997] qui permet, via un mo-

dèle probabiliste, de traiter des données présentant des caractéristiques statistiquement indépendantes.

La dernière famille regroupe les méthodes à bases de noyaux. Elles sont particulièrement indiquées lorsque les dépendances et les régularités des données d'apprentissage ne sont pas linéaires. Une fonction noyau est un produit scalaire permettant de réexprimer les données d'entrée dans un espace de plus grande dimension autorisant l'utilisation de méthodes linéaires. L'exemple le plus connu est les machines à vecteurs de support (SVM) [Boser *et al.*, 1992].

L'apprentissage supervisé suppose que l'on dispose de connaissances *a priori* sur les données à analyser ; or, l'objectif est d'explorer de nouvelles données inconnues sans avoir d'exemples antérieurs auxquels se référer, il est nécessaire de s'orienter vers l'apprentissage non supervisé.

2.1.2.2 Apprentissage non supervisé

L'apprentissage non supervisé permet de se passer de l'expert et d'un étiquetage d'une base d'apprentissage. L'objectif est de trouver au sein des données des structures sous-jacentes, des motifs non connus *a priori*. Ces motifs et structures peuvent prendre la forme de classes (groupe de données similaires), de règles logiques (règles d'association par exemple), de relations entre variables (régresseur) ou encore de modèles connexionnistes (réseaux de neurones par exemple).

La tâche de partitionnement de données (clustering en anglais) est l'une des plus traitées dans la littérature. Complémentaire à la classification évoquée plus haut, elle consiste à segmenter un ensemble de données en regroupant au sein de classe des groupes d'individus proches en termes de caractéristiques. Une bonne segmentation des données implique donc une similarité très forte entre individus d'une même classe et une similarité faible entre individus de classes différentes. Cette similarité s'exprime dans l'espace des caractéristiques par la notion de proximité. Cette proximité induit un calcul de distance entre individus qui, suivant les données manipulées, pourra se baser sur différentes formules (distance euclidienne, distance de Manhattan, distance de Mahalanobis, etc). La sélection des caractéristiques et le type de distance utilisée peut ainsi fortement influencer le résultat final.

Il existe de nombreuses méthodes de classification non supervisée :

- des méthodes plutôt géométriques utilisant par exemple les centroïdes comme dans l'algorithme des k-moyennes [Hartigan et Wong, 1979] ;
- des approches hiérarchiques qui incrémentalement vont réduire le nombre de classes en fusionnant à chaque fois deux classes relativement similaires [Johnson, 1967] ;
- des méthodes basées sur les réseaux de neurones (exemple : les cartes auto-organisatrices [Kohonen, 1988]).

De par sa capacité à « découvrir » de nouvelles informations, l'apprentissage non supervisé est couramment utilisé dans les approches dites de modélisation dirigée par les données. Nous présentons dans la suite une tendance récente qui vise à utiliser conjointement les systèmes multi-agents et la modélisation dirigée par les données.

2.1.3 Modélisation dirigée par les données et synergies possibles avec les systèmes multi-agents

La modélisation dirigée par les données vise à produire des modèles empiriques (par opposition à des modèles analytiques basés sur des modèles mathématiques ou physiques) d'un phénomène réel à partir de l'analyse de données résultant de ce même phénomène [Sharma *et al.*, 2018]. Historiquement,

les premières approches de modélisation dirigée par les données se sont intéressées à l'utilisation de régresseurs linéaires et de modèles ARIMA sur des séries temporelles. Les travaux en apprentissage et fouille de données ont par la suite ouvert de nouvelles perspectives à ce type de modélisation.

Par ailleurs, l'arrivée massive dans notre quotidien des caméras, capteurs et autres objets connectés, facilite de nos jours la collecte de données décrivant activités humaines et phénomènes sociaux. De plus, les capacités actuelles de stockage et de traitement permettent également d'envisager facilement l'analyse de grosses masses de données afin d'en extraire de la connaissance. On trouve donc dans la littérature de nombreux travaux utilisant les techniques d'apprentissage et de fouille de données pour l'observation, la reconnaissance et la modélisation de comportements humains [Cook et Krishnan, 2015, Pantic *et al.*, 2007].

Partant de ce constat, il est légitime de se demander si les techniques d'apprentissage et de fouille de données peuvent être utilisées dans le cadre des systèmes multi-agents et des simulations comportementales. Depuis une dizaine d'années, des chercheurs s'intéressent aux interactions possibles entre les domaines de la fouille de données et des systèmes multi-agents [Cao, 2009]. Le terme « Agent Mining » qui fait référence aux synergies possibles entre la fouille de données et les agents [Cao *et al.*, 2012] a d'ailleurs fait son apparition dans les thématiques de nombreux journaux et conférences. L'ouvrage [Cao, 2009] identifie deux types de contributions sur ce thème : l'apport des SMA pour le « data mining » et l'apport du « data mining » pour les SMA.

2.1.3.1 Apport des SMA pour la fouille de données et l'apprentissage

Dans ces contributions, les systèmes multi-agents peuvent être vus comme une solution aux problématiques de passage à l'échelle et de privacité lors du traitement de grandes masses de données décentralisées. Les SMA peuvent également aider à améliorer la qualité de l'apprentissage. Les algorithmes de clustering ont par exemple été adaptés à des frameworks multi-agents [Ravindra Babu *et al.*, 2010] permettant ainsi de générer de manière concurrente des classifications dont les résultats sont ensuite combinés par un système de vote de manière à obtenir une plus grande justesse dans la segmentation des données.

La simulation multi-agent peut également être utilisée pour consolider des données. Considérant une simulation suffisamment réaliste, les résultats de celle-ci peuvent venir compléter les données d'entrée d'un processus d'apprentissage [Baqueiro *et al.*, 2009]. Les avantages d'une telle approche sont de pouvoir contrôler via la simulation la « bonne » diversité des données en entrée, d'éviter la génération de données erronées ou bruitées (contrairement à des données collectées sur le terrain), de pouvoir garantir une masse de données statistiquement suffisante.

2.1.3.2 Apport de la fouille de données et de l'apprentissage pour les SMA

Dans [Remondino et Correndo, 2005], les auteurs posent pour la première fois les contours d'un usage de la fouille de données pour la conception de simulations comportementales à base d'agents. Ils envisagent uniquement l'utilisation de la fouille de données sur le résultat des simulations et distinguent deux types de contributions possibles :

- une utilisation endogène de la fouille de données consistant à intégrer au sein même de l'agent des capacités d'apprentissage lui permettant d'analyser les données issues de ses précédentes exécutions dans l'optique d'améliorer le réalisme de la simulation ;

- une application exogène où la fouille de données est plutôt utilisée du point de vue du concepteur de la simulation pour analyser les résultats de celle-ci et aider à la phase de révision et d'adaptation des comportements des agents.

L'intérêt d'utiliser des techniques de fouille sur les données de simulation est d'obtenir un haut niveau de description des résultats afin de faciliter l'analyse et la comparaison : entre plusieurs séries d'expérimentations ou entre données réelles et simulées [Remondino et Correndo, 2006, Baqueiro *et al.*, 2009].

L'utilisation de la fouille de données pour abstraire des observations réelles et construire des modèles d'agent a été relativement peu abordée dans la littérature. Parmi les quelques travaux existants, la thèse de S. Hassan [Hassan, 2010] pose les bases d'une méthodologie générale permettant d'injecter des données réelles dans une simulation comportementale. Il considère ainsi trois acteurs impliqués dans le développement de la simulation : l'expert du domaine, l'expert en modélisation et le développeur. Le rôle de l'expert du domaine est de choisir quelles données sont intéressantes à collecter et sur la base de ces données de proposer une abstraction du phénomène à simuler. Grâce à cette abstraction, l'expert en modélisation propose un modèle d'agents en y intégrant des caractéristiques identifiées dans les données disponibles. Ce modèle est ensuite implémenté et paramétré par le développeur. La dernière étape de la méthodologie, la validation, implique les trois acteurs qui évaluent le réalisme de la simulation et décident si l'une ou plusieurs des étapes doivent être répétées pour améliorer la simulation.

Dans le détail, Hassan identifie à différentes étapes de la méthodologie l'usage de techniques de fouille de données et d'apprentissage. Ainsi, en amont de l'abstraction, il identifie l'usage d'ontologies pour organiser les données collectées qui pourraient provenir de différentes sources et l'usage d'algorithmes de clustering pour pouvoir regrouper en plus petits ensembles d'éventuelles grandes masses de données. Durant la phase de modélisation, il suggère également l'utilisation de réseaux de neurones pour faciliter la modélisation de comportements adaptatifs. Tout comme [Baqueiro *et al.*, 2009], il prévoit également l'usage de la fouille de données pour faciliter l'interprétation et la validation des résultats de simulation.

Poursuivant cette démarche d'intégration des techniques de fouille de données dans le processus de conception d'une simulation comportementale, nous avons investigué au travers des thèses d'Imen Saffar et d'Antoine Sylvain les possibilités offertes par l'apprentissage automatique pour l'identification et l'extraction de comportements d'agents à partir d'observations d'activités humaines.

2.2 Agentification dirigée par les données pour la simulation d'activité humaine

Les objectifs des deux thèses précitées étaient de pousser un peu plus loin l'usage de l'apprentissage automatique notamment sur les tâches d'abstraction et de modélisation. Sans aller jusqu'à se passer complètement de l'expertise du domaine, l'idée était d'évaluer la faisabilité à remplacer une partie de cette expertise par des approches automatisées utilisant les données d'observation. Dans cette section, nous dressons un état de l'art sur les méthodes d'apprentissage automatique dédiées à l'identification et l'extraction de comportements à partir d'observations réelles. Après une discussion sur les usages et limites de toutes ces méthodes, nous introduisons une approche d'agentification basée sur la classification non supervisée.

2.2.1 Approches pour l'identification et l'extraction automatique de comportements à partir d'observations réelles

Nous présentons ici un certain nombre de techniques de la littérature permettant d'identifier et d'extraire des comportements sur la base d'observations réelles. Les données de ces observations peuvent prendre différentes formes : séquences audio [Rouas *et al.*, 2006], vidéo [Ghorbel *et al.*, 2016], suivi de trajectoires par RFID⁴ [Kim *et al.*, 2013], trace de navigation web [Dziczkowski *et al.*, 2011], etc. Bien qu'une grande majorité des travaux traitent d'observations vidéos, les approches proposées sont souvent transposables à d'autres types de données.

2.2.1.1 Apprentissage d'actions simples

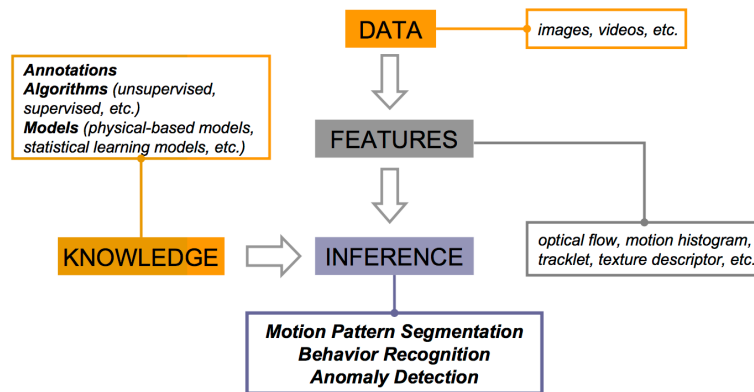


FIGURE 2.2 – Structure générale d'analyse de scène vidéo [Li *et al.*, 2015]

L'analyse et la compréhension automatique de séquences vidéos reposent sur une succession d'étapes (figure 2.2) à commencer par l'extraction de caractéristiques au sein des images composant la séquence. Parmi les caractéristiques les plus souvent utilisées, on compte :

- le flot optique qui représente le mouvement apparent de chaque pixel sur plusieurs images consécutives [Beauchemin et Barron, 1995];
 - les trajectoires des personnes dans l'image exprimées soit en coordonnées cartésiennes, soit en vitesse, soit en relations spatio-temporelles entre les objets de l'image, etc [Cedras et Shah, 1995];
 - les descripteurs géométriques de blobs⁵ : frontière du blob, moment, squelette, etc [Hu, 1962].
- Ces caractéristiques sont ensuite exploitées pour reconnaître des actions et primitives simples. Trois types d'approches existent à ce niveau :
- les approches non paramétriques qui s'appuient sur des appariements 2D (ou 3D) entre des blobs et des formes prédéfinies [Bobick et Davis, 2001, Gorelick *et al.*, 2007];
 - les approches volumétriques qui ne se basent pas sur des extractions de caractéristiques image par image, mais considèrent plutôt la séquence vidéo comme un volume des intensités des pixels qui est ensuite traité par des algorithmes de filtrages [Young *et al.*, 2001, Jhuang *et al.*, 2007], d'appariements [Shechtman et Irani, 2005];

4. Radio Frequency IDentification

5. Un blob est une région d'une image contenant un ensemble de pixels connectés spatialement.

- les approches paramétriques qui se basent sur des modèles prédéfinis de dynamique temporelle d'actions comme par exemple les HMM (Hidden Markov Model) qui imposent que le temps soit modélisé comme une séquence de transitions probabilistes entre des états discrets [Moore *et al.*, 1999, Hongeng et Nevatia, 2003].

Une fois reconnues, des actions simples peuvent ensuite être combinées ensemble pour apprendre des activités⁶ plus complexes.

2.2.1.2 Apprentissage d'activités

Les travaux de la littérature sur l'apprentissage d'activités se répartissent en trois familles.

La première regroupe les approches graphiques qui vont considérer des modèles d'activités mettant en scène des actions primitives. Des réseaux de Petri, par exemple, peuvent être utilisés pour modéliser des séquencements particuliers d'actions et définis *a priori* [Castel *et al.*, 1996, Albanese *et al.*, 2008].

Inspirée de la fouille de texte, la seconde famille rassemble les approches grammaticales qui se basent sur la définition d'un langage pour décrire des activités complexes. Les phrases de ce langage sont des activités construites à partir de mots (les actions reconnues au niveau bas) et de règles de grammaire qui définissent un modèle d'activité [Ryoo et Aggarwal, 2006]. La définition d'un tel langage étant difficile, des travaux se sont penchés sur l'apprentissage de règles de grammaire à partir de corpus, mais les résultats sont limités et difficilement généralisables [De La Higuera, 2000].

Enfin, on trouve des approches utilisant des systèmes de règles logiques permettant de décrire des connaissances liées au domaine des observations. Ces règles peuvent ainsi décrire des structures de scènes, d'événements, d'enchaînement d'actions, etc [Rota et Thonnat, 2000].

2.2.1.3 Usages et limites de l'apprentissage à partir d'observations réelles

Les méthodes d'apprentissage précédemment décrites sont motivées par la problématique très ambitieuse de compréhension automatique d'observations réelles qui consiste à répondre à quatre questions :

- Qui ? , c'est-à-dire quels sont les protagonistes, les acteurs impliqués dans la scène observée ;
- Quoi ? , question qui fait référence aux actions et activités de ces protagonistes ;
- Pourquoi ? , qui désigne les motivations et intentions qui ont poussé les protagonistes à agir ;
- et enfin Comment ? , qui regroupe les processus de décision qui ont produit les actions observées

La figure 2.3 met en perspective ces questions au regard des travaux existants dans la littérature. Les approches décrites précédemment permettent de répondre aux deux premières questions (Qui et Quoi ?) moyennant l'introduction de connaissances *a priori* sur les observations du monde réel. Ces connaissances consistent la plupart du temps en des modèles des activités potentiellement observables.

Comme le soulignent les auteurs dans [Turaga *et al.*, 2008], la question du « Pourquoi ? » reste un challenge et peu de travaux se penchent sur l'inférence des motivations et intentions. Enfin, la question du « Comment ? » semble à l'heure actuelle hors de portée (la problématique étant même très peu évoquée comme perspective de recherche). Or la réponse à ces deux questions permet justement

6. Le terme *activité* désigne dans la plupart des articles un enchaînement d'actions élémentaires pouvant éventuellement impliquer plusieurs personnes.

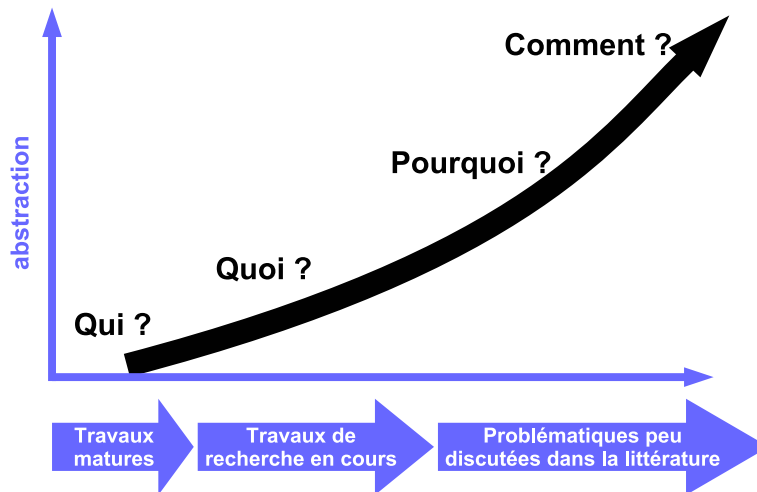


FIGURE 2.3 – Processus d’analyse et de compréhension automatique d’observations réelles

d’identifier les buts qui conditionnent un comportement d’agents dans une simulation comportementale.

Une dernière limite, de plus en plus discutée dans les conférences en IA, concerne l’explicabilité des méthodes d’apprentissage. D’une manière générale, plus une méthode d’apprentissage va être efficace en termes de prédiction, plus l’obtention des résultats va être difficile à comprendre pour un humain [Holzinger, 2018]. L’apprentissage profond par exemple est souvent considéré comme une boîte noire : il y est difficile de savoir quelles sont les données d’entrée qui ont permis d’aboutir à la décision [Samek *et al.*, 2017].

En première approche et dans un cadre général, il ne semble donc pas possible d’utiliser des techniques d’apprentissage pour construire, sans l’intervention d’experts humains, des comportements d’agents. Cependant, pour certains types de simulation, ces approches d’apprentissage peuvent tout de même grandement faciliter (voire automatiser) certaines tâches de conception.

2.2.2 Proposition d’une approche d’agentification dirigée par les données pour des simulations sociales et situées

Compte tenu des limites évoquées précédemment, le choix a été fait dans les thèses d’Imen Saffar et d’Antoine Sylvain de se focaliser sur des simulations d’activités humaines pouvant être modélisées par des agents situés. Dit autrement, nous nous restreignons aux activités humaines impliquant des déplacements spatiaux. Deux raisons motivent ce choix. D’une part, la littérature sur les modèles de déplacements d’agents situés est extrêmement riche et nombre d’entre eux sont suffisamment génériques pour être utilisés, quelle que soit l’application visée. D’autre part, pour ce type de simulation, il est assez facile pour le concepteur d’inférer les motivations et intentions des agents en fonction de la localité des actions observées.

Dans la suite, les comportements de nos agents situés reposeront donc sur un modèle de déplacement décomposé en trois niveaux de décision [Hoogendoorn et Bovy, 2004] :

- niveau stratégique : au niveau duquel l’agent décidera d’un plan (une trajectoire) en fonction

de ses buts, motivations et points d'intérêt (POI) à visiter;

- niveau tactique : qui consistera à suivre la trajectoire planifiée au niveau stratégique tout en évitant les éventuels obstacles et autres agents circulant dans l'environnement;
- niveau opérationnel : où l'agent sera amené à réaliser de manière effective son déplacement en changeant sa position dans l'environnement.

Ce modèle de déplacement pourra être éventuellement alimenté au niveau stratégique par un comportement supplétif capable de générer dynamiquement des points d'intérêt (POI) dans l'environnement en fonction de buts et motivations de haut niveau.

La figure 2.4 présente le modèle de comportement considéré par la suite. Les briques de « planification d'itinéraire » et de « suivi de trajectoire et évitement d'obstacles » font respectivement référence à des algorithmes classiques de la littérature [Ahuja *et al.*, 1990, Helbing et Molnár, 1995, Teknomo, 2006, Reynolds, 1999] qui pourront être paramétrés via l'apprentissage sur des données réelles d'observation.

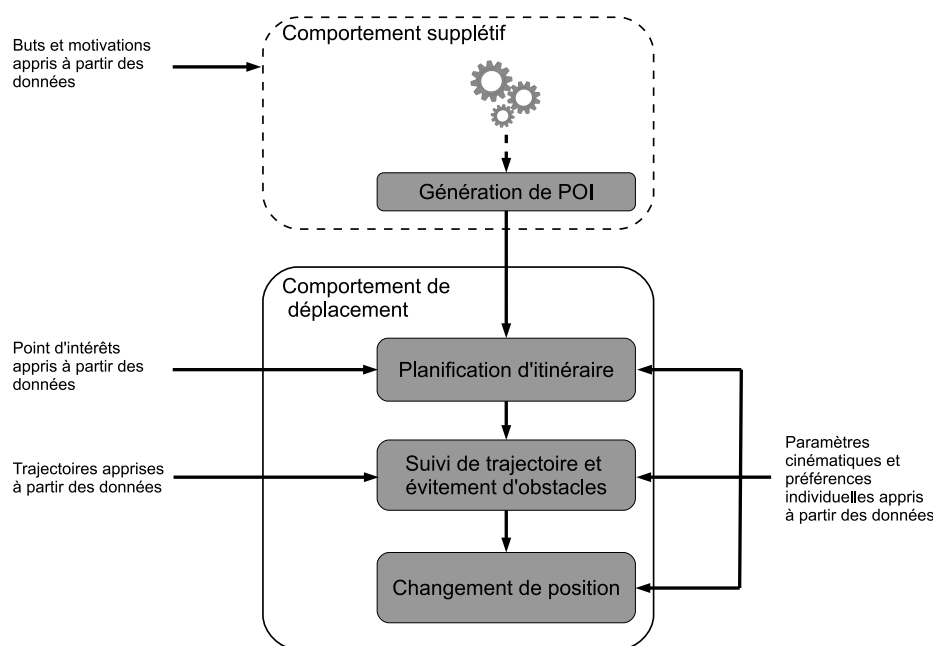


FIGURE 2.4 – Modèle de comportement considéré

Tous les agents partageront ce modèle et les différentes typologies d'agents s'exprimeront donc au travers des buts, motivations et points d'intérêt ainsi qu'au travers des paramètres cinématiques utilisés pour configurer les algorithmes précédemment évoqués. L'apport des techniques de fouille de données sera donc sur l'identification de cet ensemble de caractéristiques propres à chaque catégorie d'agents.

Sur la base de ces hypothèses, nous proposons dans la suite de décomposer plus finement les tâches d'observation et d'abstraction du monde réel présentées en section 2.1.1 (voir figure 2.1) en y intégrant les aspects de fouille de données (figure 2.5). Dans ce processus, l'étape d'observation du monde réel consiste à collecter des données (individuelles et collectives) via l'usage de différents capteurs (cf. section 2.2.2.1). Ces données sont ensuite nettoyées (suppression des données erronées, traitement des données bruitées, etc) et séparées en deux ensembles : une partie pour l'apprentissage

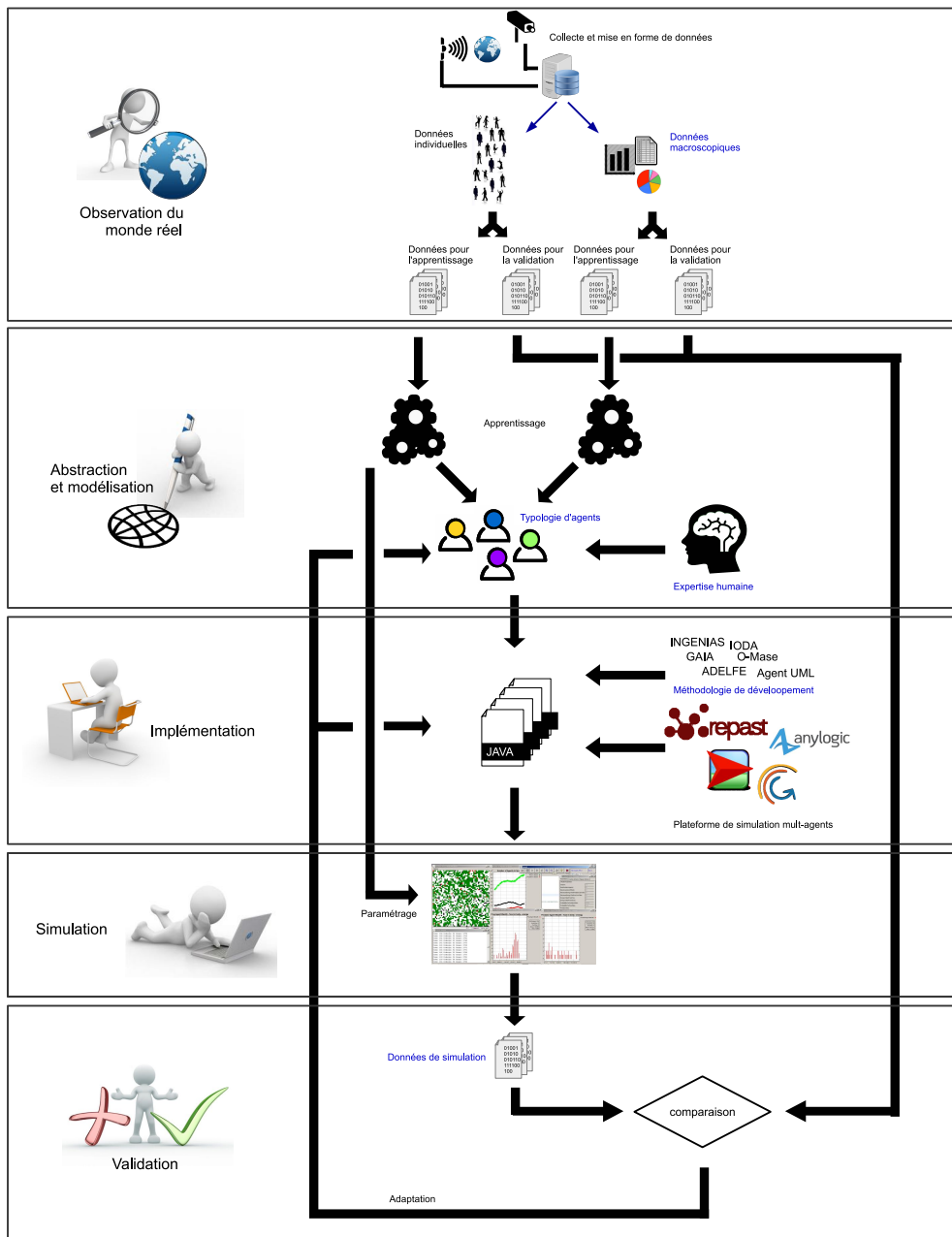


FIGURE 2.5 – Intégration des techniques d’apprentissage dans un processus de conception de simulations multi-agents

et une partie pour la validation.

L’étape d’abstraction et de modélisation (cf. section 2.2.2.2) se base alors sur l’ensemble d’apprentissage ainsi constitué pour permettre :

- l’identification des grandes catégories d’agents qui viendront peupler la simulation,

- l'extraction des paramètres de ces agents qui viendront alimenter le modèle de comportement précédemment fixé.

A ce niveau, l'expertise humaine est requise pour interpréter et traduire les lieux visités en termes de buts et de motivations de haut niveau.

Le modèle de déplacement ainsi paramétré via les données peut ensuite être implémenté via les outils et méthodologies présentés en sections 2.1.1.2 et 2.1.1.4. La simulation résultant de cette phase d'implémentation est ensuite évaluée en comparant ses résultats avec la partie des données réelles laissée pour la validation. Si l'écart entre le réel et le simulé est trop important, la modélisation et/ou son implémentation sont remises en cause et adaptées de manière à accroître le réalisme de la simulation.

Dans la suite, nous détaillons successivement les phases d'observation du monde réel et d'abstraction / modélisation.

2.2.2.1 Observation du monde réel : collecte et usage des données

Pour pouvoir s'appuyer sur des techniques d'apprentissage lors de la phase d'abstraction et de modélisation, il est nécessaire d'avoir à disposition une masse de données suffisante et représentative du phénomène à simuler. La phase d'observation doit donc s'appuyer sur des approches de collectes plutôt automatisées. Les phénomènes que nous cherchons à modéliser étant basés sur des déplacements, plusieurs technologies peuvent être utilisées. Le tableau 2.1 regroupe les principales.

TABLEAU 2.1 – Technologies utilisables pour la capture de données de déplacement

	données individuelles	données collectives
boucles de comptage / capteurs de présence		+
caméra	+	+
RFID	+	+
Smartphone	+	+

Les technologies les plus basiques comme les boucles de comptage et autres capteurs de présence sont relativement limitées et ne sont capables que de fournir des données collectives de type flux de passage. Elle nécessite une instrumentation de l'environnement et de ce fait elles ne pourront collecter des données qu'à des endroits bien précis.

Les caméras requièrent également une instrumentation de l'environnement. Elles peuvent capturer des données collectives telles que des flux ou des densités d'occupation de l'espace, mais peuvent aussi, sous certaines conditions (couverture suffisante de l'environnement en nombre de caméras, absence de phénomènes d'occultation), capturer des déplacements individuels moyennant l'usage d'algorithmes de tracking vidéo dans un réseau de caméra [Hou *et al.*, 2017].

Le RFID est une alternative possible aux caméras pour le suivi de personnes. Une antenne placée dans l'environnement peut émettre dans son périmètre proche (de l'ordre du mètre) des ondes radio auxquelles des marqueurs (tag) peuvent répondre. Chaque tag est associé à un identifiant unique, ce qui permet de garantir un suivi individuel de celui-ci moyennant le déploiement d'un nombre suffisant d'antennes dans un environnement. Tout comme les caméras, il est également possible de réaliser un comptage par zone du nombre de tags présents et ainsi obtenir des cartes de densités et des flux de passage.

Le suivi de personnes à l'aide de smartphone suit la même logique à la différence qu'il n'y a pas

forcément d'instrumentation de l'environnement à opérer puisque la géolocalisation d'un smartphone s'appuie sur les infrastructures des opérateurs téléphoniques. Les données collectées sont nativement individuelles, mais peuvent être agrégées pour fournir des données plus macroscopiques. L'usage du smartphone est clairement plus adapté à la collecte de données en extérieur et sur de grandes distances tandis que les caméras et le RFID sont plus propices à un usage à l'intérieur de bâtiments.

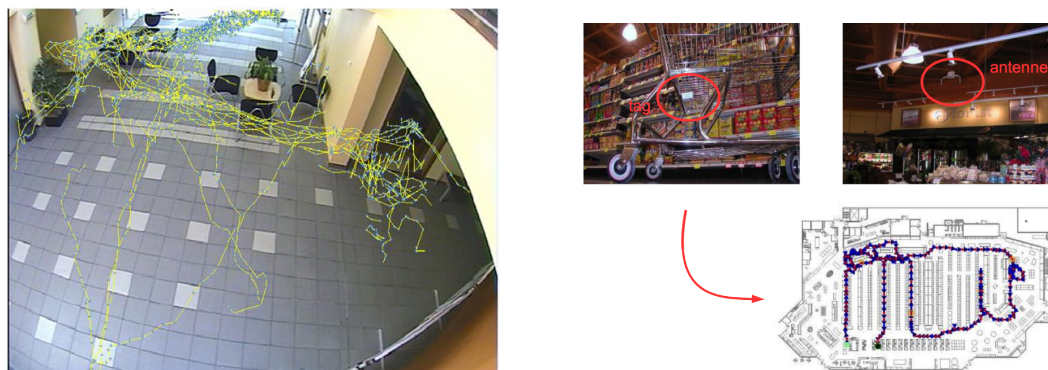


FIGURE 2.6 – Exemples de captures de trajectoires en environnement intérieur [Saffar, 2013, Sylvain, 2016]

Dans la cadre de la thèse d'Imen Saffar, les données ont été capturées via une caméra de surveillance. Pour la thèse d'Antoine Sylvain, les trajectoires proviennent de relevés de position obtenus grâce à l'usage d'antennes et de tag RFID. (figure 2.6). Dans les deux cas, nous avons pu obtenir à la fois des données individuelles de type trajectoires et des données collectives de flux et de densité (sous forme de carte de chaleur de l'environnement). Ces données ont ensuite été utilisées suivant le processus décrit à la figure 2.5 pour construire respectivement une simulation de hall de bâtiment universitaire (thèse d'Imen Saffar) et une simulation de magasin (thèse d'Antoine Sylvain).

2.2.2.2 Abstraction et modélisation : identification des typologies et comportement d'agents

En préambule à toute tentative de modélisation, il est nécessaire d'abstraire et de simplifier les observations du monde réel. Nous proposons ici d'abstraire le monde réel en cherchant à identifier les entités qui sont prépondérantes au sein du phénomène observé. Par « prépondérant », nous entendons une représentativité numérique suffisante sur l'ensemble des données d'observation. Pour pouvoir faire apparaître cette représentativité numérique, il est nécessaire de regrouper sous forme de classes l'ensemble des entités présentes dans les observations. Pour ce faire, nous proposons l'utilisation de techniques de clustering sur les données individuelles collectées.

Différentes caractéristiques extraites de ces données individuelles peuvent être utilisées pour faire apparaître les entités prépondérantes du phénomène observé. Si le concepteur choisit de considérer comme vecteur de caractéristiques une liste de coordonnées cartésiennes, il pourra regrouper et modéliser ensemble des usagers empruntant des chemins similaires dans l'environnement et partageant vraisemblablement des motivations communes. Par exemple dans la thèse de Saffar, les individus ayant des trajectoires allant d'une salle de la classe à la machine à café définissent la classe des individus (élèves ou enseignants) désireux de prendre un café durant leur pause [Saffar, 2013]. Si le concepteur décide d'extraire des caractéristiques plus cinématiques telles que la vitesse, les change-

ments de direction, il obtiendra des typologies relatives à la manière de se déplacer et d'éviter les obstacles. A la fin de cette étape, des clusters peu représentés (i.e. ceux contenant un nombre d'individus relativement faible au regard de la base utilisée pour l'apprentissage) pourront être éliminés.

Dans le cadre des thèses d'Imen Saffar et d'Antoine Sylvain, deux principaux algorithmes de clustering ont été utilisés : K-Means [Hartigan et Wong, 1979] et EM [Dempster *et al.*, 1977]. Ces deux algorithmes itératifs diffèrent dans leur approche de partitionnement. K-Means adopte une vision plutôt géométrique du partitionnement en utilisant la notion de centroïdes (barycentre des clusters) et de distance entre individus. EM, pour sa part, utilise une vision plutôt statistique en utilisant un modèle de mélange gaussien et la notion de maximum de vraisemblance. De par son principe de fonctionnement, l'algorithme du K-Means aura du mal à gérer des clusters qui se chevauchent ou des clusters de forme non convexe ; dans ces cas, l'algorithme EM est plus adapté. En revanche, pour les deux algorithmes, il est nécessaire de prédéfinir le nombre de clusters que l'on souhaite atteindre. Plusieurs méthodes existent pour aider au choix de ce nombre en se basant uniquement sur des considérations statistiques sans connaissance des phénomènes réels ayant engendré les données : visualisation graphique de la variance en fonction du nombre de clusters (méthode Elbow), visualisation graphique de la mesure de similarité (outil Silhouette [Rousseeuw, 1987]), cross-validation [Kohavi, 1995].

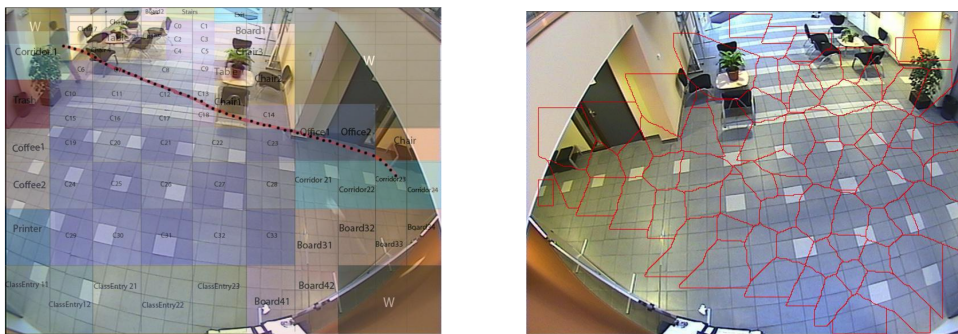


FIGURE 2.7 – Exemples de découpage de l'environnement observé. A gauche, découpage manuel et annotation de celui-ci par le concepteur de la simulation. A droite, découpage automatique de l'environnement à l'aide de diagramme de Voronoi [Saffar *et al.*, 2011]

Afin de pouvoir segmenter des données, ces algorithmes existants requièrent des données homogènes en termes de taille. Par exemple, dans le cas de trajectoires, celles-ci doivent être prétraitées de manière à en extraire des vecteurs de caractéristiques de même taille. Dans la thèse d'Imen Saffar, nous avons eu recours à un ré-échantillonnage classique permettant d'homogénéiser le nombre de points constitutifs et de conserver l'ordre des points. Cette transformation nécessite un découpage de l'environnement qui peut être réalisé manuellement, permettant ainsi au concepteur de la simulation d'associer à certains lieux de l'environnement des actions réalisables. Par exemple, dans la figure 2.7, les cellules étiquetées sur le côté gauche de l'image font référence à un « espace détente » regroupant les distributeurs de boissons.

Cet étiquetage peut ensuite être utilisé pour construire une représentation symbolique des trajectoires facilitant ainsi l'identification des buts et motivations sous-jacents à chaque trajectoire [Saffar *et al.*, 2011, Höferlin *et al.*, 2011]. Chaque point de la trajectoire est ainsi étiqueté par l'identifiant de la région dans lequel le point se trouve. Dans l'exemple de la figure 2.8, la trajectoire verte peut être symbolisée par la chaîne de caractères A,A,C,C,F,D,D,D,D. Cette représentation symbolique peut ensuite être simplifiée en prenant en compte différents critères comme par exemple la durée de présence

dans la région. Dans l'exemple de la trajectoire verte, le temps de présence dans la région A étant relativement court, les deux premières occurrences de A peuvent être réduites à une. Par contre, le temps de présence dans la région D ne permet de supprimer que 2 occurrences sur les 4 initiales.

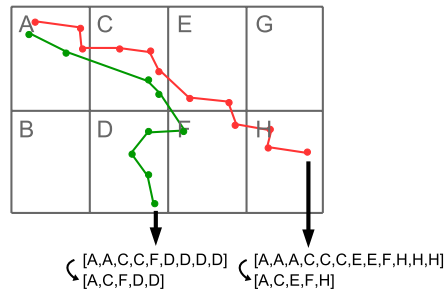


FIGURE 2.8 – Représentation symbolique d'une trajectoire [Saffar *et al.*, 2011, Höferlin *et al.*, 2011]

Dans la thèse d'Antoine Sylvain, le comportement de déplacement étant guidé par un comportement d'achat de plus niveau, il n'y avait pas d'intérêt à conserver l'ordre des points de chaque trajectoire et il était plutôt souhaitable de faire apparaître des points d'intérêt de l'environnement. Chaque trajectoire a donc été transformée en une matrice de points de passage en utilisant une grille qui discrétise l'environnement observé (figure 2.9).

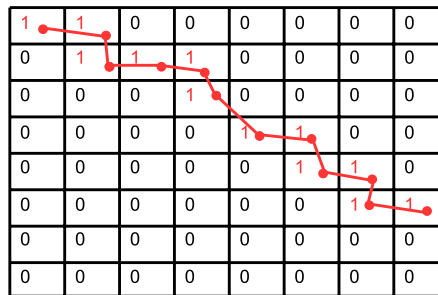


FIGURE 2.9 – Transformation de trajectoire en matrice de points de passage

La figure 2.10 résume les différents clustering qui peuvent être appliqués sur les données de déplacements individuels. Les clusters obtenus contiennent ainsi des informations de nature diverse :

- information de centralité : moyenne et médiane des individus regroupés dans le cluster, individu le plus central dans la classe (médoïde) ;
- variations statistiques au sein du cluster : cardinalité, écart type, écart moyen, variance ;
- information décrivant l'emprise du cluster dans l'espace : forme du cluster, convexité de l'enveloppe.

Dans sa thèse, Imen Saffar a proposé l'utilisation suivante de ces informations.

- Chaque cluster est interprété comme une typologie d'agents. Dans le cas où le concepteur réalise plusieurs clusterings en considérant différents vecteurs caractéristiques (topologiques, cinématiques), celui-ci pourra regarder la répartition des individus dans les différents clusters obtenus pour affiner le paramétrage des différentes typologies d'agents.
- Les informations de centralité, issues du clustering de trajectoires, permettent de fournir : des trajectoires moyennes utilisables directement par le niveau tactique de comportement de

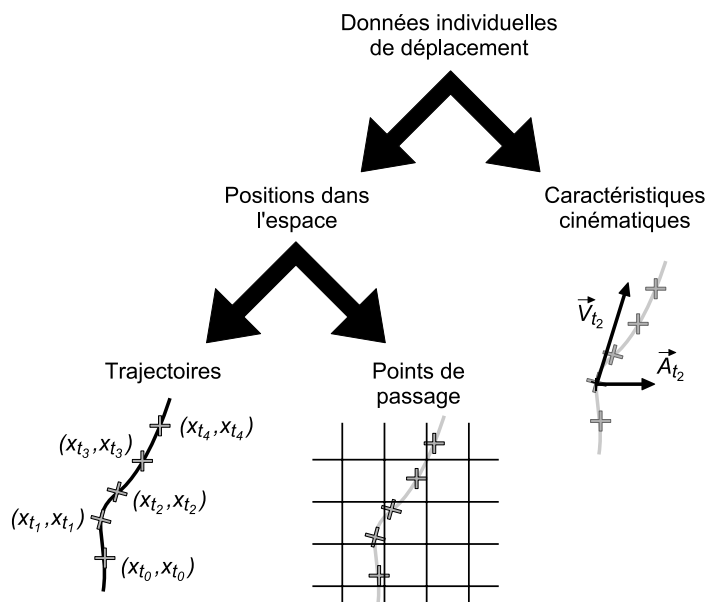


FIGURE 2.10 – Choix de vecteurs caractéristiques possibles sur les données de déplacements individuels

déplacement, des points d'intérêt (points de passage moyen) utilisables en entrée du niveau stratégique, des paramètres cinématiques permettant de régler les différents algorithmes du comportement de déplacement.

- La cardinalité de chaque cluster ramenée à la taille de la base d'apprentissage (en nombre d'individus) fournit la représentativité (en pourcentage) de chaque typologie d'agent au sein de la simulation.

Cette méthodologie d'agentification introduite dans [Saffar, 2013] a été appliquée, en preuve de concept, à la simulation d'un hall de bâtiment universitaire. Les agents simulés dans cette première application disposaient uniquement d'un comportement de déplacement ; les motivations (points d'intérêts de l'environnement) leur étant fournies à partir de clusters. La même méthodologie a ensuite été reprise dans la thèse d'Antoine Sylvain [Sylvain, 2016] pour un cas plus complexe de simulation d'un magasin. Ce cas nous a amenés à considérer un comportement de déplacement motivé par un comportement d'achat (comportement supplétif). Cette application est présentée en détail en section suivante.

2.3 Application à la simulation d'un espace de vente

Nous présentons dans cette section un cas concret d'utilisation de la fouille de données dans la conception d'une simulation comportementale. Elle consiste à reproduire les activités d'achat et de déplacement de clients au sein d'un supermarché (Figure 2.11). Il s'agit donc de mettre en œuvre un ensemble d'agents situés (représentant des clients) et évoluant dans un même environnement, représentation virtuelle d'un supermarché. Les agents sont libres de leur navigation à travers les différentes parties du magasin : allées, rayons, caisses. Les produits du magasin sont considérés comme des res-

sources de l'environnement.

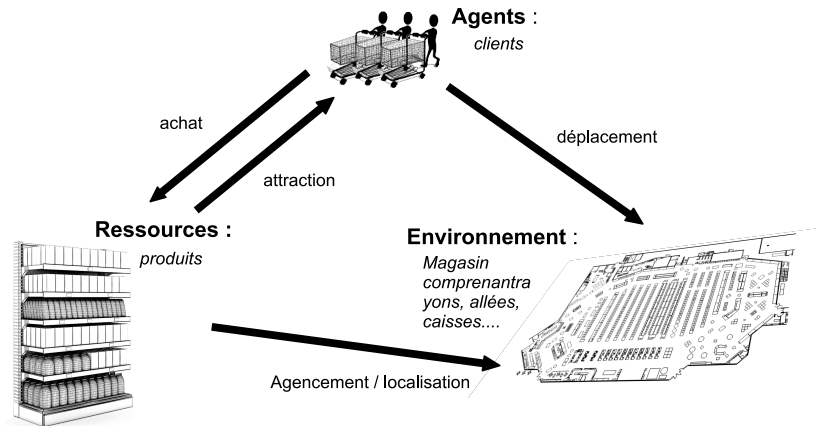


FIGURE 2.11 – Vue globale de la simulation

Dans la littérature, deux sous-comportements décrivant les agissements d'un client en magasin sont souvent considérés : un comportement de déambulation et un comportement d'achat. Les études en sociologie, psychologie et marketing montrent que l'intention d'achat est conditionnée par plusieurs facteurs [Grewal *et al.*, 1998] : le besoin du produit (liste de course), la réputation de la marque du produit, le prix, la position du produit dans les rayonnages. Elles montrent également que l'environnement perçu par le client influence aussi son intention d'achat. Ainsi, chaque client s'accorde un temps imparti pour réaliser ses courses. On parle de pression temporelle qui agit sur la réceptivité du client aux produits et le pousse à sortir du magasin une fois le temps imparti écoulé. De la même manière, le niveau de foule dans les différents rayons peut inciter le client à raccourcir leur parcours en magasin [Aylott et Mitchell, 1998]. Le comportement du client est donc évolutif en fonction de ces pressions temporelles et spatiales.

Dans la thèse d'Antoine Sylvain, une modélisation de l'influence de ces différents facteurs a été proposée sous la forme d'une relation d'attraction client-produit. Cette relation est utilisée pour définir un comportement d'achat venant influencer le comportement de déplacement comme suggéré dans la figure 2.4. L'ensemble de ces deux comportements est qui plus est évolutif durant la simulation en fonction de la pression temporelle propre à chaque agent et de la densité de clients en rayon. Dans ce contexte, nous considérons une version simplifiée du cadre formel introduit en section 1.1.2.2. Chaque agent dispose de deux actions élémentaires : se déplacer à un point spécifique du magasin et acheter un produit en rayon (i.e mettre dans le caddy). L'état de l'agent correspond à sa position en magasin. Les états de l'environnement, quant à eux, correspondent aux produits cartésiens des localisations individuelles des clients. Ainsi, l'état de congestion résultera d'un grand nombre d'agents localisés dans un même rayon. Dans ce modèle, nous ne considérerons pas d'évènement extérieur pouvant venir modifier l'état global de l'environnement : $E_v = \emptyset$ et $P_{E_v} = 0$.

Nous présentons dans la suite le modèle de comportement client et l'usage de données réelles pour

paramétrer ce comportement.

2.3.1 Représentation du magasin et des produits

La surface commerciale d'un supermarché est relativement structurée de par la présence des linéaires et présentoirs qui forment des allées et des rayons. Cette structuration est facilement modélisable sous la forme d'un graphe : les nœuds du graphe correspondant soit aux intersections entre les allées et rayons, soit à des localisations de catégories de produits, et les arcs permettant de définir les connexités et les obstacles entre ces différents points du magasin. A chaque nœud sont associées des informations additionnelles comme le nom du rayon auquel il appartient, la catégorie de produit correspondante, la liste des produits disponibles, etc.

Chaque produit p du magasin est défini par $\langle Cat_p, Rpr_p, A_p \rangle$. Cat_p désigne la catégorisation du produit (par exemple : Epicerie Salée / Snack / Chips) et Rpr_p le prix réel du produit. En plus de ces deux attributs, nous associons à chaque produit un vecteur A_p regroupant quatre paramètres d'attractivité du produit.

- *Attractivité relative à la hauteur du produit en rayon* ($A_p^1 \in [0, 1]$). De nombreuses études en marketing et psychologie du consommateur montre qu'un produit situé à hauteur des yeux à plus de chance d'être acheté.
- *Attractivité relative au positionnement du prix* dans une même catégorie ($A_p^2 \in [0, 1]$). Ce paramètre est directement lié à la capacité d'achat des clients.
- *Attractivité de la marque* ($A_p^3 \in \{0, 0.5, 1\}$) : ce paramètre fait référence au positionnement du produit dans une gamme, à la réputation de la marque. Pour des produits alimentaires, il est par exemple possible de distinguer : le low cost ($A_p^3 = 0$), les marques distributeurs ($A_p^3 = 0.5$) et les grandes marques ($A_p^3 = 1$).
- *Attractivité de la promotion* ($A_p^4 \in \{0, 0.5, 1\}$). On distingue ici les offres spéciales pouvant être appliquées aux produits : pas de réduction, remise, déstockage, etc.

Pour mettre en place la relation d'attraction client-produit, ces quatre paramètres sont mis en correspondance avec quatre paramètres de réceptivité côté client.

2.3.2 Attributs retenus pour le client

Chaque client simulé a est localisé dans son environnement via des coordonnées cartésiennes et un nœud de rattachement. Il est également défini par un tuple $\langle Ln_a, Lp_a, Vd_a, \eta_a(t), Ra \rangle$.

Pour le comportement de déplacement de l'agent, un itinéraire au sein du magasin est prédéfini pour chaque agent a et représenté sous la forme d'une liste Ln_a de points d'intérêt à visiter.

Pour le comportement d'achat, les intentions d'achat initiales de chaque agent sont regroupées dans une table de hachage Lp_a contenant tous les produits du magasin. A chaque clé (produit) $p \in Lp_a$ est associée une intention d'achat. Une fois l'agent créé, celle-ci pourra évoluer dynamiquement à chaque pas t de la simulation notée $I_a^p(t)$. Cette intention évoluera dans le temps et permettra à l'agent de décider s'il achète ou non un produit.

D'autres paramètres sont également définis pour un client : sa vitesse de déplacement désirée Vd_a et une pression temporelle ressentie à l'instant t : $\eta_a(t)$ normalisée entre 0 et 1. A sa création, chaque agent dispose d'une pression temporelle initiale $:\eta_a(0)$; au cours de la simulation, la pression

temporelle va augmenter pour tendre vers 1 et inciter l'agent à abréger son parcours en magasin.

Enfin, chaque agent va disposer de quatre paramètres de réceptivité regroupés dans le vecteur R_a :

- Réceptivité au positionnement du produit en rayon ($R_a^1 \in [0, 1]$) qui influence le regard et prise en main du produit.
- Réceptivité au budget ($R_a^2 \in [0, 1]$) qui exprime un niveau de prix que le client est prêt à supporter.
- Réceptivité aux marques ($R_a^3 \in \{0, 0.5, 1\}$) qui représente l'inclination pour les produits low cost, les marques distributeurs ou les grandes marques.
- Réceptivité aux promotions et offres spéciales ($R_a^4 \in \{0, 0.5, 1\}$) qui caractérise la sensibilité d'un client aux diverses opérations de marketing.

2.3.3 Modélisation de l'attraction client-produit

La relation d'attraction entre un client a et un produit p [Sylvain *et al.*, 2014] va dépendre de la mise en correspondance notée $\Phi(a, p)$ des deux vecteurs R_a et A_p dont les composantes sont normalisées dans l'intervalle $[0, 1]$. L'équation 2.1 exprime la relation $\Phi(a, p)$ comme une somme pondérée de fonctions appliquées aux composantes des vecteurs A_p et R_a .

$$\Phi(a, p) = \frac{1}{4} \sum_{i=1}^4 f_i(A_p^i, R_a^i) \quad (2.1)$$

Les fonctions f_i utilisées dépendent du type d'attributs considérés : quantitatif ou qualitatif. Lorsque A_p^i et R_a^i concernent des informations quantitatives (le prix, la hauteur en rayon par exemple), la fonction utilisée est définie par l'équation 2.2. L'application de la fonction tend vers 1 lorsque l'attractivité et la réceptivité sont proches. A l'inverse, lorsque la différence devient significative, le résultat tend vers 0. Les fonctions f_1 et f_2 sont définies suivant cette équation et permettent de comparer respectivement la hauteur du regard d'un client avec la position du produit et l'attente du client en termes de prix avec le positionnement tarifaire d'un produit.

$$f_i(A_p^i, R_a^i) = 1 - |A_p^i - R_a^i| \quad \text{où} \quad A_p^i, R_a^i \in [0, 1] \quad (2.2)$$

Lorsque A_p^i et R_a^i se rapportent à des informations plutôt qualitatives (les promotions, la marque par exemple), la fonction utilisée est définie par l'équation 2.3. Les fonctions f_3 et f_4 sont définies sur ce modèle. Par exemple, si un agent a est absolument insensible aux marques (i.e. $R_a^3 = 0$), alors la fonction f_3 tendra vers 0. A l'inverse, si un consommateur est attentif à la réputation des marques qu'il achète (i.e. $R_a^3 = 1$), alors il sera particulièrement attiré vers les produits p dont la valeur A_p^3 tend vers 1.

$$f_i(A_p^i, R_a^i) = A_p^i \cdot R_a^i + \frac{(1 - R_a^i)}{2} \quad (2.3)$$

où $A_p^i, R_a^i \in \{0, 0.5, 1\}$

Ce modèle d'attraction est à la base du comportement d'agent décrit ci-après.

2.3.4 Description du comportement d'agents « client »

Le comportement comprend cinq étapes correspondant à des prises de décision ou des actions précises.

Etape 1 : choix du prochain rayon Lorsqu'un agent a entre dans le magasin ou quitte sa position courante, il doit déterminer le prochain lieu du magasin à visiter. Il utilise pour cela sa position courante dans le graphe pour calculer le prochain nœud sur lequel se rendre en utilisant la liste Ln_a . A la création de l'agent, la liste Ln_a est ordonnée et correspond au plus court chemin partant de l'entrée, visitant tous les points d'intérêts de l'agent et finissant par un passage en caisse. Au fur et à mesure de la simulation, ce chemin pourra évoluer du fait de la visite de certains nœuds non prévus initialement, mais particulièrement attractifs au regard du modèle précédemment décrit.

Cette attractivité des nœuds est représentée sous la forme d'un score $S_a(n)$ calculé par l'agent a suivant l'équation 2.4. Plus le nœud possèdera des produits attractifs du point de vue de l'agent, plus le score sera élevé. Le prochain nœud initialement prévu dans la liste Ln_a bénéficie d'un bonus de 1 de manière à limiter le phénomène de dispersion par rapport à la liste Ln_a de départ.

$$S_a(n) = \sum_{p_i \in Lp_a(n)} \Phi(a, p_i) + \begin{cases} 1 & \text{si } n \in Ln_a \\ 0 & \text{sinon} \end{cases} \quad (2.4)$$

avec :

$S_a(n)$: le score du nœud n considéré par l'agent a

$Lp_a(n)$: l'ensemble des produits rattachés à ce nœud n

Ln_a : la suite des points d'intérêts initiaux pour l'agent a

Le nœud n' retenu par l'agent sera celui présentant le meilleur score. Si le nœud n' n'est pas le prochain élément de Ln_a , n' est ajouté en début de liste Ln_a de manière à être visité aux étapes 2 et 3 qui suivent.

Etape 2 : déplacement vers le prochain nœud Le déplacement de la position courante de l'agent vers le prochain nœud de Ln_a se base sur des algorithmes classiques de la littérature : planification de chemin et évitement d'obstacles. Pour l'implémentation de ces algorithmes, l'agent doit disposer d'un cône de vision lui permettant notamment de détecter la présence d'autres agents sur son chemin. Cette information est également utilisée pour mettre à jour dynamiquement la pression temporelle de l'agent. Normalisée entre 0 et 1, la densité d'agent $\delta_a(t)$ perçue dans le cône de vision vient s'ajouter à la pression temporelle initiale $\eta_a(0)$ (Equation 2.5).

$$\eta_a(t) = \eta_a(0) + \delta_a(t) \quad (2.5)$$

avec :

$\eta_a(0) \in [0, 1]$: la pression temporelle initiale,

$\delta_a(t) \in [0, 1]$: la densité perçue à l'instant t .

Une densité d'agents élevée à l'instant t influence temporairement la pression temporelle à l'instant t . Cela signifie qu'un agent arrivant dans un rayon très chargé aura une pression temporelle élevée

durant sa visite, ce qui aura un impact sur l'intention d'achat de tous les produits p de la liste Lp_a suivant l'équation 2.6. Dès que l'agent reviendra dans un rayon moins chargé, sa pression temporelle redescendra, proche de son état initial $\eta_a(0)$.

$$I_a^p(t) = (1 - \eta_a(t)), \dots, I_a^p(t - 1) \quad (2.6)$$

avec :

$\eta_a(t) \in [0, 1]$: la pression temporelle à l'instant t ,

$I_a^p(0) \in [0, 1]$: l'intention initiale d'achat d'un produit p définie par Lp_a .

Etape 3 : évaluation des produits du nœud courant Lorsqu'un agent a arrive sur un nœud n , il récupère la liste des produits attachés à ce nœud et estime leur intérêt. Pour chaque produit $p \in Lp_a(n)$, l'agent évalue son intention d'achat contextualisée notée $C_a^p(t)$ suivant l'équation 2.7.

$$C_a^p(t) = \frac{I_a^p(t) + \Phi(a, p)}{2} \quad (2.7)$$

avec :

$I_a^p(t)$: l'intention d'achat de l'agent a pour le produit p à l'instant t ,

$C_a^p(t)$: l'intention d'achat contextualisée pour le produit p à l'instant t ,

$\Phi(a, p)$: la mise en correspondance des attributs agents et produits.

L'agent a construit l'ensemble $Lp_a^{Sel}(n)$ composé pour chaque catégorie Cat_j des produits p_k présentant la valeur maximale $C_a^{p_k}(t)$ (équation 2.8). A la fin de cette étape, il obtient donc pour chaque catégorie représentée sur le nœud les produits avec les plus hautes intentions d'achat.

$$Lp_a^{Sel}(n) = [max_{p_k \in Cat_1} C_a^{p_k}(t), \dots, max_{p_k \in Cat_j} C_a^{p_k}(t), \dots, max_{p_k \in Cat_l} C_a^{p_k}(t)] \quad (2.8)$$

avec l le nombre de catégories de produits présentes sur le nœud n .

Etape 4 : Délibération L'agent a évalue individuellement chaque produit de Lp_a^{Sel} et décide d'acheter ou non le produit en comparant l'intention d'achat contextualisée (équation 2.7) du produit avec un seuil d'achat noté SdA (équation 2.9).

$$\{p \in Lp_a^{Sel}(n) / C_a^p(t) \geq SdA\} \quad (2.9)$$

Une fois le produit p acheté, son prix réel Rpr_p sera utilisé pour calculer le montant total du panier. De plus, tous les produits appartenant à la même catégorie que p verront leur intention d'achat mise à zéro pour le reste du parcours.

Etape 5 : passage en caisse Lorsque l'agent a a fini son parcours en magasin, il se dirige vers les caisses. Le choix de la caisse se base sur deux critères : la distance entre la caisse et la position courante ainsi que le nombre de clients attendant à la caisse. L'agent a essaie donc de trouver la valeur minimale S_{a,c_j} pour toutes les caisses c_j ouvertes (équation 2.10).

$$S_{a,c_j} = (1 + Q_{c_j}), \dots, \Delta(a, c_j) \quad (2.10)$$

avec :

Q_{c_j} : la taille de file d'attente à la caisse c_j ,

$\Delta(a, c_j)$: la distance entre l'agent a et la caisse c_j .

Après le passage en caisse, le client disparaît de la simulation.

2.3.5 Paramétrage du modèle d'agent consommateur à partir de données réelles

Nous présentons maintenant l'application de la méthodologie détaillée en section 2.2.2.2 à ce modèle d'agent consommateur. Les données utilisées nous ont été mises à disposition par la société américaine *Sorensen Associate*.

2.3.5.1 Données de supermarché

Ces données, recueillies dans le cadre d'un audit, concernent un supermarché californien d'une surface commerciale de 4150 m^2 . L'agencement intérieur y est classique : 17 rayons situés au centre et les espaces dédiés aux produits frais (boucherie, poissonnerie, fruits et légumes) situés en périphérie. Les entrées / sorties sont quant à elle situées de part et d'autre d'une ligne de caisses qui en totalise 11. La figure 2.12 présente le plan de masse annoté du magasin.

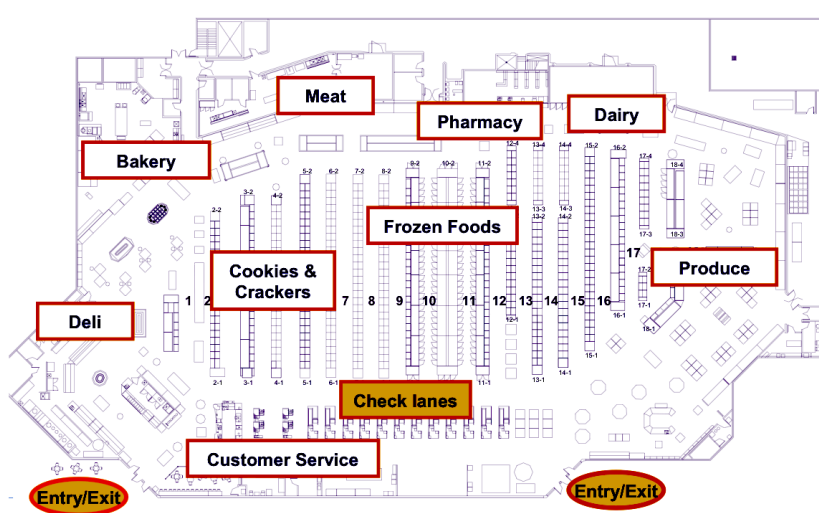


FIGURE 2.12 – Agencement du supermarché considéré

Le plafond du magasin a été équipé d'antennes (environ 1 500) capables de détecter chaque seconde les signaux de tag RFID placés sur certains chariots et paniers du magasin (250 tags déployés en tout). Ce système permet ainsi de suivre le cheminement individuel d'un client en magasin de son entrée jusqu'à son passage en caisse. Lorsque le tag RFID du chariot d'un client passe sous une antenne, les coordonnées de cette antenne viennent s'ajouter à une liste représentant les points de passages du

client. Dès que le chariot passe la ligne de caisse, la liste associée est retranscrite dans une base de données en tant que trajectoire d'un client. Puis, elle est remise à zéro pour la prochaine utilisation du chariot. Ce recueil de données s'est fait sur une période de 28 jours permettant d'obtenir 2 076 trajectoires de clients.

Sur cette même période, les tickets de caisse ont été historisés (environ 220 000). Parmi ces tickets, ceux correspondant aux 2 076 trajectoires n'ont pu être isolés : plusieurs chariots équipés de tags RFID à une même caisse sont difficilement discernables individuellement.

Ces données ont été partitionnées, la moitié servant à l'apprentissage et au paramétrage de la simulation, l'autre pour la validation. Dans la suite, nous présentons le traitement et l'analyse des trajectoires de manière à venir alimenter notre modèle d'agent client.

2.3.5.2 Clustering de trajectoires

Comme présenté en section 2.2.2.2, la segmentation des trajectoires permet de faire apparaître des typologies de déplacements motivés par le visite de point d'intérêt de l'environnement. Pour le paramétrage de notre modèle d'agent client, il s'agit de pouvoir renseigner plus particulièrement la liste Lp_a de chaque agent a de la simulation.

Suivant la méthodologie exposée en figure 2.9, nous avons transformé l'ensemble des trajectoires $T_i (1 \leq i \leq 2076)$ en matrice de points de passage. La matrice M obtenue est de taille 2076×1500 : chaque ligne i de la matrice correspond à une trajectoire client, chaque colonne j correspond à un point de passage (antenne) en magasin :

$$M = (a_{ij})_{1 \leq i \leq 2076, 1 \leq j \leq 1500} = \begin{cases} 1 & \text{si } j \in T_i \\ 0 & \text{sinon} \end{cases}$$

L'algorithme K-Means [Hartigan et Wong, 1979] a été utilisé sur M pour segmenter les points de passage. Le paramètre k a été déterminé empiriquement en testant différentes valeurs comprises en 5 et 30. Chaque résultat de clustering a été visualisé sous l'outil *Silhouette* de Matlab. Entre 15 et 30 clusters, le découpage est intéressant puisque nous obtenons une belle variété de comportements tout en garantissant une représentativité des clusters supérieure à 1% de la population. En dessous de 15, les comportements sont trop agrégés, ce qui conduirait à une simulation beaucoup trop simplifiée de la réalité. Au-dessus, de 30, il devient difficile de faire la distinction entre certains clusters relativement proches en termes d'individus. La figure 2.13 présente quelques exemples de clusters parmi les 30 retenus pour nos simulations. Les points noirs correspondent à la fréquentation moyenne des points de passage d'un cluster : plus les points apparaissent en gros, plus ils sont représentés dans le cluster. Dans [Larson *et al.*, 2005], les auteurs ont également exploité le même jeu de données via des approches de clustering mais en conservant les trajectoires. Les clustering obtenus sont bien entendu différents.

La population d'agents a été construite en suivant la cardinalité de chaque cluster. Le tableau 2.2 synthétise les pourcentages d'agents issus de chaque cluster.

2.3.5.3 Autres paramètres

En complément des trajectoires dont le traitement vient d'être présenté, nous avons également pu avoir accès aux données de vente du magasin avec par exemple pour chaque produit : son prix, son

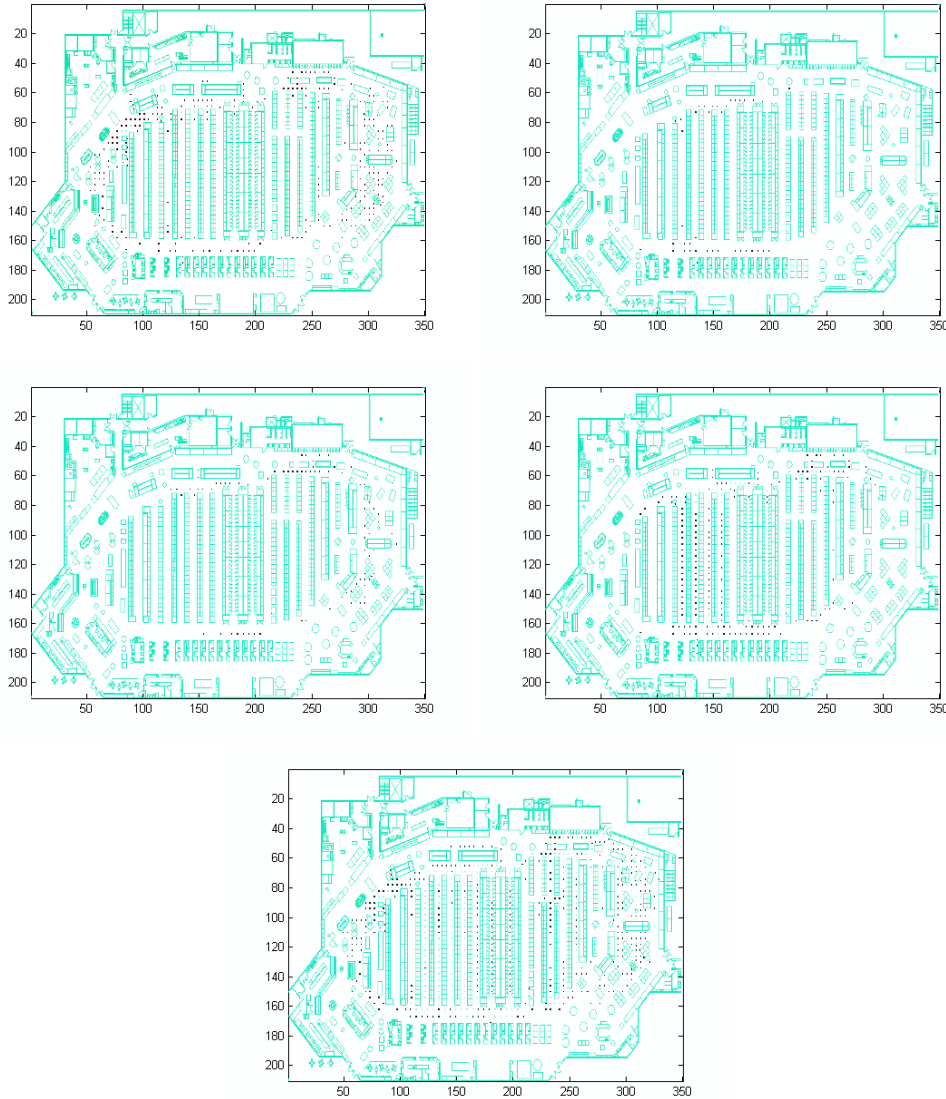


FIGURE 2.13 – Exemples de *clusters* obtenus

positionnement en rayon, sa catégorisation, sa marque et son positionnement, sa part de marché, les offres promotionnelles associées, etc. Ses éléments nous ont permis de paramétrer les caractéristiques des produits : $\langle Cat_p, Rpr_p, A_p \rangle$

Pour l'initialisation de la liste Lp_a de chaque agent, nous nous sommes basés sur les tickets de caisse en triant les produits du plus vendu au moins vendu. Le classement obtenu a été utilisé pour définir l'intention initiale d'achat I_a^p . Tous les agents sont initialisés avec les mêmes intentions d'achat, ce sont leurs parcours différents en magasin qui produiront des paniers différents par agent. Le seuil d'achat SdA a été calibré empiriquement à partir de la première moitié des données de manière à

TABLEAU 2.2 – Répartition des différentes typologies de clients dans la simulation

Cluster 1	7 %
Clusters 3, 6, 14, 15, 17, 29	6 %
Clusters 4, 10, 22, 27	4 %
Clusters 8, 11, 13, 16, 19, 25, 26	3 %
Clusters 5, 12, 18, 20, 21, 23, 24, 30	2 %
Clusters 2, 7, 9, 28	1 %

obtenir un panier moyen égal à la moyenne relevée par la société *Sorensen Associate* lors de l’audit du magasin.

2.3.6 Expérimentations

Pour valider le modèle proposé, nous avons réalisé une série d’expérimentations comparant les résultats de simulation et la seconde moitié des données à notre disposition. Notre modèle étant composé de deux sous-comportements liés : un comportement de déambulation et un comportement d’achat, nous avons retenu comme critères de comparaison : le temps passé en magasin, la distance parcourue, le montant moyen du panier et la carte de chaleur du magasin.

Les résultats présentés dans le tableau 2.3 sont une moyenne sur 25 simulations d’une journée. Pour chaque journée simulée, 5 000 agents sont générés suivant la répartition du tableau 2.2.

TABLEAU 2.3 – Comparaison réel / simulé

	Données réelles	Résultats de simulation	Ecart relatif
Temps passé dans le magasin (minutes)	23	19	21 %
Distance parcourue dans le magasin (meters)	2 067	1 533	34 %
Montant moyen du panier (dollars)	27,1	25,85	4,8 %

Les comportements obtenus en simulation sont assez diversifiés. Par exemple, les distances parcourues varient entre 108 mètres (pour les agents issus d’un cluster correspondant à une visite courte en magasin) jusqu’à 3000 mètres (pour les agents ayant un parcours quasiment exhaustif du magasin). Naturellement, le temps de présence en magasin suit une tendance similaire avec des temps variant de 2 minutes à plus de 1 heure. En moyenne, la distance réelle parcourue en magasin par les clients est d’environ 2000 mètres. En simulation, la moyenne obtenue est inférieure de 30 %. Cette différence peut s’expliquer par l’utilisation d’un algorithme de planification de plus court chemin permettant aux agents de se déplacer entre deux points de la liste L_n . Cet algorithme tend à produire en simulation des trajectoires optimales alors que dans la réalité nous pouvons observer des zigzags, des retours en arrière et autres détours qui viennent allonger le parcours des clients en magasin. Concernant le montant moyen du panier, l’écart entre le réel et le simulé est inférieur à 5%.

Un dernier critère de comparaison porte sur la répartition des flux dans le magasin et les densités qui en découlent. Nous avons ainsi comparé des cartes de chaleurs (représentant les points chauds et les points froids du magasin). La figure 2.14 présente une carte de chaleur réelle obtenue lors de

l'audit et une carte de chaleur obtenue en simulation. Les zones très fréquentées (en rouge) sont dans les deux cas localisées sur les mêmes quatre rayons situés entre les abscisses 120 et 170, ainsi que sur les têtes de gondole. Similairement, sur les deux cartes, les points froids (en bleu / vert) se retrouvent sur les extérieurs de la surface de vente correspondant à des zones de moindre passage.

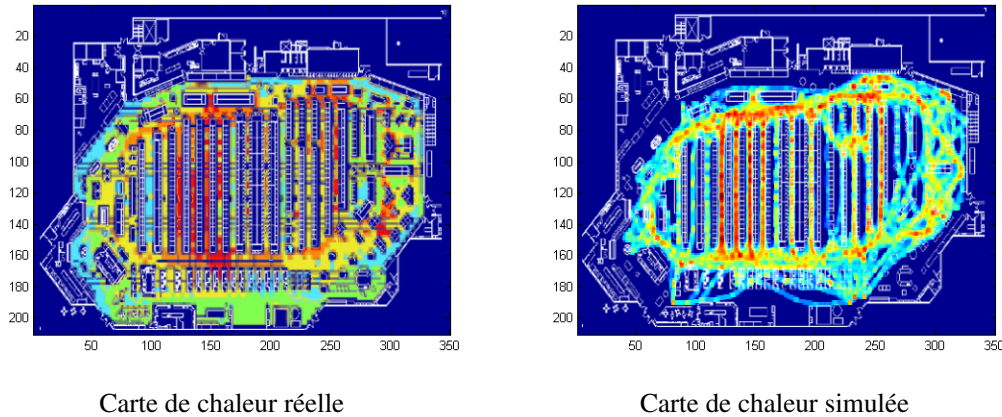


FIGURE 2.14 – Comparaison réel / simulé sur la base de cartes de chaleur

L'ensemble de ces résultats tend à montrer que le modèle de comportement proposé et paramétré avec les données réelles à notre disposition est relativement réaliste tant du point de vue du déplacement des clients que de leurs achats. Rappelons que ces déplacements résultent à la fois des points d'intérêts de la liste L_n , mais aussi du comportement (supplétif) d'achat qui peut pousser un agent à visiter d'autres rayons du magasin voire à écourter sa visite lorsque la pression temporelle est trop importante.

Une telle simulation peut ainsi être utilisée de manière exploratoire pour tester des scénarios inédits. Dans la thèse d'Antoine Sylvain, le simulateur a pu être utilisé pour évaluer l'impact d'une campagne marketing visant à augmenter le flux de visite en magasin. Les résultats obtenus ont ainsi montré qu'une telle campagne permettrait de faire augmenter, comme espéré, le chiffre d'affaires global du magasin, mais au détriment du montant moyen du panier client.

2.4 Conclusion

Même lorsqu'elles sont développées avec des approches comportementales, les simulations sociales restent difficiles à construire. De nombreux travaux, issus de la communauté « multi-agent », ont permis de formaliser et d'outiller le processus de développement de telles simulations. Cependant, peu de contributions ont porté sur les premières étapes du processus, à savoir l'observation du monde réel et son abstraction. Dans ce chapitre, nous avons présenté une approche qui vise à la fois à tirer parti des capacités actuelles à collecter des informations sur les comportements et interactions humaines (via les caméras et objets connectés) et des possibilités offertes par les algorithmes de fouille de données. L'état de l'art sur ces algorithmes montre qu'ils permettent d'obtenir des modèles purement descriptifs de comportements humains par rapport à une situation observée. En revanche, ils ne permettent pas de monter en abstraction pour obtenir des modèles explicatifs des phénomènes ob-

servés (i.e. des modèles permettant de reproduire les prises de décisions qui ont conduit aux actions effectivement observées).

L'approche présentée dans ce chapitre a été développée dans le cadre des thèses d'Imen Saffar [Saffar, 2013] et d'Antoine Sylvain [Sylvain, 2016] et a donné lieu à plusieurs publications : 1 journal actuellement en seconde révision [Doniec *et al.*, 2019], 2 conférences internationales [Saffar *et al.*, 2011], [Sylvain *et al.*, 2014] et 1 conférence nationale : [Saffar *et al.*, 2012]. Elle se restreint ainsi aux activités humaines impliquant des déplacements d'individus et s'applique donc à des agents situés pour lesquels le modèle de déplacement est fixé a priori et repose sur trois niveaux de décisions : stratégique, tactique et opérationnel. Ce modèle de déplacement peut être alimenté via l'usage d'algorithmes de clustering sur des données réelles qui fournissent points d'intérêts, trajectoires et paramètres cinématiques pour différentes typologies d'agents. Nous obtenons ainsi un modèle descriptif de déplacement permettant aux agents de la simulation de reproduire les dynamiques observées dans le monde réel. Comme discuté dans la thèse d'Imen Saffar, la simulation obtenue ne permet pas d'investiguer sur le contexte et les causes qui ont conduit aux phénomènes observés.

Dans la thèse d'Antoine Sylvain, nous avons donc considéré au-dessus du comportement de déplacement, un comportement supplétif capable de générer des points d'intérêts de l'environnement en entrée du niveau stratégique du comportement de déplacement. Il se veut donc en quelque sorte un modèle explicatif des déplacements des individus. Compte tenu des limitations évoquées en section 2.2.1.3, ce comportement supplétif ne peut être appris et nécessite donc le recours à des connaissances expertes pour expliciter les mécanismes de décision qui sous-tendent les déplacements.

Chapitre 3

Coordination d'agents en environnement dynamique et incertain

Le chapitre 1 a mis en avant les difficultés à piloter un système évolutif notamment lorsque celui-ci est constitué d'entités physiquement distribuées dont le comportement global réclame une coordination.

La coordination est une des problématiques majeures dans les systèmes multi-agents. Regroupant l'ensemble des processus permettant d'atteindre un objectif déterminé elle constitue l'une des facettes de l'interaction pouvant exister entre les agents du système. Étudiée dès le début des années 80, la coordination a fait l'objet de nombreux travaux : coordination dans un contexte compétitif [Mandiau *et al.*, 2008], approches réactives [Tlig *et al.*, 2013], planification [Dimopoulos et Moraitis, 2006], etc.

Une tendance qui s'est dégagée ces dernières années vise à définir la coordination dans un cadre plus formel [Lesser et Corkill, 2014] de manière à :

- obtenir des mécanismes plus génériques utilisables dans différents contextes et indépendants de l'application finale,
- avoir une meilleure analyse et compréhension de la complexité algorithmique inhérente aux problèmes de coordination,
- étudier plus finement l'optimalité des mécanismes de coordination.

Dans ce chapitre, nous considérons deux cadres formels que sont les problèmes de satisfaction de contraintes (CSP) (et leur variante d'optimisation sous contraintes appelée les COP) et les processus de décision markoviens.

3.1 La coordination vue comme un problème de satisfaction de contraintes

La programmation par contrainte est un outil classique de l'intelligence artificielle consistant à représenter un problème de décision sous forme de variables liées entre elles par des relations (appelées contraintes) venant restreindre les valeurs applicables à ces variables. Plus formellement, un problème de satisfaction de contraintes (CSP - Constraint Satisfaction Problem) est un triplet (X, D, C) où X désigne un ensemble fini de variables x_i , chacune associée à un domaine $dom(x_i) \in D$ représentant l'ensemble fini et discret des valeurs possibles pour x_i [Kumar, 1992]. Ces variables sont sujettes à

un certain nombre de contraintes appartenant à l'ensemble C . La résolution d'un CSP consiste alors à affecter à chaque variable x_i une valeur appartenant à $dom(x_i)$ et consistante avec l'ensemble des contraintes de C . Trois types d'algorithmes de résolution existent dans la littérature : des algorithmes de recherche exhaustive, basés sur le principe du backtracking, qui vont parcourir l'espace de recherche $dom(x_1) \times \dots \times dom(x_n)$, des algorithmes de filtrage permettant de réduire la taille des domaines et des algorithmes de propagation mariant à la fois recherche exhaustive et filtrage [Rossi *et al.*, 2006].

Les CSP possèdent de nombreuses déclinaisons. Les COP (Constraint Optimization Problem), par exemple, permettent de faire la distinction entre les contraintes dures qui doivent absolument être satisfaites et les contraintes qui peuvent être éventuellement violées. Ces dernières peuvent être vues comme des fonctions de coûts que l'on cherche à maximiser (ou minimiser). CSP et COP possèdent tous deux leurs variantes dans un cadre distribué. Les disCSP (distributed Constraint Satisfaction Problem) [Mandiau *et al.*, 2014] se définissent ainsi comme un quadruplet (X, D, C, A) , extension du triplet (X, D, C) définissant un CSP, et où A désigne un ensemble d'agents A_1, A_2, \dots, A_p dans lequel chaque A_k ($1 \leq k \leq p$) possède : un sous-ensemble de X noté X_{A_k} tel que $\bigcap_{A_k \in A} X_{A_k} = \emptyset$ et un sous-ensemble de C noté C_{A_k} . Du point de vue de l'agent A_k , l'ensemble X_{A_k} désigne ses variables propres tandis que l'ensemble $X \setminus X_{A_k}$ désigne des variables étrangères. Les DCOP [Picard, 2018], quant à eux, peuvent être vus comme des disCSP où chaque contrainte $c_i \in C$ définit un coût associé aux variables impliquées dans cette contrainte.

Dans [Doniec *et al.*, 2008], les CSP avaient été utilisés pour la coordination d'agents en situation de conflits au sein d'une simulation de trafic. Chaque agent devant se coordonner utilisait des contraintes pour décrire les relations le reliant aux autres agents de son environnement. Le filtrage du CSP permettait alors à l'agent d'anticiper les états non désirés (figure 3.1) avant l'exécution d'un mécanisme de coordination à base de jeux [Mandiau *et al.*, 2008].

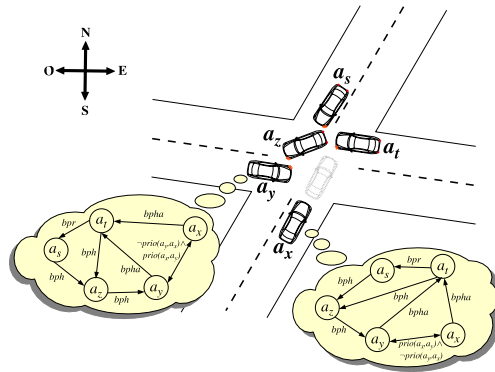


FIGURE 3.1 – Utilisation de CSP pour la coordination d'agents en carrefour [Doniec *et al.*, 2008]

La résolution d'un CSP peut également permettre d'exprimer des problématiques de coordination en considérant les hypothèses suivantes :

- les agents à coordonner disposent à chaque instant t d'un ensemble fini d'états complètement observables et d'un ensemble fini d'actions ;
- la coordination entre ces agents s'exprime comme la séquence d'actions conjointes des agents sur un horizon temporel $[T, T']$;
- l'environnement dans lequel doivent se coordonner les agents peut être soumis à des aléas et

à l'apparition d'évènements non prévus (cf. la définition d'un système évolutif introduite en section 1.1.2.2)

L'approche de coordination via un formalisme à base de contraintes consiste alors à considérer que la séquence d'actions conjointes attendue sur l'horizon $[T, T']$ est obtenue en résolvant une suite de $(T' - T)$ problèmes de satisfaction de contraintes ; chaque solution d'un de ces problèmes correspondant aux actions que devront réaliser les agents à un instant $t \in [T, T']$. Chaque problème possède donc un nombre de variables égal au nombre d'agents à coordonner, les actions possibles de ces agents définissant le domaine de ces variables. Les contraintes de cette suite de problèmes viennent exprimer les objectifs de la coordination, mais aussi les interactions possibles entre les agents (en interdisant par exemple des combinaisons d'actions qui nuiraient à la coordination).

Cette approche est développée dans les deux sous-sections qui suivent au travers deux exemples applicatifs différents : la gestion de ressources sur un système en réseau et l'exploration multi-robot.

3.1.1 Coordination dans les systèmes en réseau

Dans le cadre du projet GEPET-Eau, nous nous sommes intéressés à la gestion de la répartition des volumes d'eau sur les réseaux de voies navigables. Ces réseaux sont composés de biefs sur lesquels il est nécessaire de maintenir un niveau d'eau compris entre une limite haute (notée HNL) et une limite basse (notée LNL) offrant ainsi un rectangle de navigation permettant aux bateaux de circuler sans risque (cf. chapitre 1, section 1.2.4, figure 1.5). Pour tenter de réguler les niveaux dans les biefs, le gestionnaire dispose d'un ensemble d'ouvrages, répartis sur le réseau, tels que des vannes, barrages et pompes.

3.1.1.1 Graphe de flots et CSP

Un tel réseau hydrographique peut être supervisé par un ensemble d'agents en charge d'un ou plusieurs biefs et capables d'agir sur le niveau de ceux-ci via les ouvrages précédemment cités. En première approche, en considérant chaque agent affecté à la conduite d'un bief, la coordination de ces derniers peut se ramener à un problème d'optimisation de flots. Les voies navigables peuvent en effet se représenter sous la forme d'un graphe de transport $G(N, A)$ dans lequel chaque nœud $i, j \in N$ symbolise un bief du réseau et les arcs $(i, j) \in A$ modélisent les échanges d'eau possibles entre ces biefs [Duviella *et al.*, 2018]. Deux nœuds fictifs supplémentaires sont considérés : un nœud source noté o et un nœud puits noté n permettant de matérialiser les volumes d'eau entrants et sortants du réseau considéré. Chaque bief $i \in N$ sera également associé à une capacité d_i bornée par rapport à sa limite haute de navigabilité (HNL) et sa limite basse (LNL). Notons V_i^{NNL} , V_i^{HNL} et V_i^{LNL} les volumes correspondant respectivement au niveau normal de navigation (NNL) dans le bief, au niveau haut (HNL) et au niveau bas (LNL). La capacité d_i peut s'exprimer de manière relative à V_i^{NNL} et ainsi être bornée par (Equation (3.1)) :

$$V_i^{NNL} - V_i^{HNL} \leq d_i \leq V_i^{NNL} - V_i^{LNL} \quad (3.1)$$

Au cours du temps, l'ensemble des capacités d_i du graphe sont amenées à évoluer en fonction :
 — des échanges d'eau contrôlés entre les nœuds (principalement dus à l'usage des écluses et des pompes)

- des échanges d'eau non contrôlés entre le réseau et l'environnement (déversement de rivières naturelles, pluies, rejets et pompages liés à des activités industrielles et humaines, échanges avec les nappes phréatiques)

A chaque arc reliant un nœud i à un nœud j est associé à un flot noté ϕ_{ij} représentant le volume d'eau échangé du bief i vers le bief j . La figure 3.2 présente la modélisation d'une partie d'un réseau réel sous la forme d'un graphe.

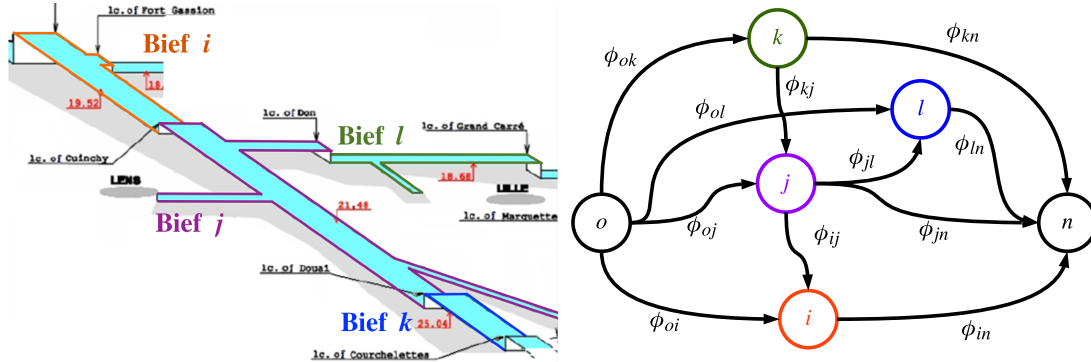


FIGURE 3.2 – Modélisation sous forme d'un graphe d'une partie du réseau des Hauts-de-France (cf. figure 1.4)

Les volumes d'eau qui peuvent s'échanger entre deux biefs vont dépendre des caractéristiques techniques des ouvrages qui les séparent. Par exemple, dans le cadre d'une vanne, le volume d'eau va dépendre de son débit maximal. Pour une écluse, le volume sera dépendant de la taille du sas et du nombre de bassinées¹. Fonction de ces caractéristiques et des types d'échanges d'eau modélisés au niveau des arcs, chaque flot ϕ_{ij} possède donc une capacité maximale et minimale : $l_{ij} \leq \phi_{ij} \leq u_{ij}$ (le détail des calculs des bornes l_{ij} et u_{ij} est présenté dans [Nouasse *et al.*, 2015]).

La dynamique d'un nœud du graphe à un instant k est le résultat d'un bilan volumique impliquant les flots entrants et les flots sortants du nœud : $d_i(k) = \sum \phi_{pi}(k) - \sum \phi_{iq}(k)$, $p, q \in N$. L'objectif est de pouvoir trouver à chaque période de temps k les valeurs des ϕ_{ij} garantissant pour chaque bief de rester dans l'intervalle $[LNL, HNL]$. Pour ce faire, il est possible de considérer un problème de satisfaction de contraintes dans lequel les variables sont l'ensemble des ϕ_{ij} du graphe, chacun des ϕ_{ij} ayant pour domaine l'intervalle $[l_{ij}, u_{ij}]$.

3.1.1.2 Coordination centralisée

Le CSP construit à partir du graphe précédemment introduit peut être ensuite utilisé pour définir un mécanisme de coordination centralisé impliquant un agent « coordinateur » et $|N|$ agents « d'exploitation » en charge de la supervision des niveaux d'eau dans chaque bief du réseau. Du point de vue d'un agent d'exploitation i , ce mécanisme de coordination implique de :

1. mesurer à l'instant k , l'écart $d_i(k)$ entre le volume d'eau présent dans son bief et V_i^{NNL} ;
2. comptabiliser le nombre de bateaux naviguant sur son bief et se dirigeant vers les biefs amont ou aval ;

1. Une bassinée correspond au remplissage et à la vidange du sas d'une écluse lors du passage d'un bateau.

3. communiquer ces deux informations à l'agent « coordinateur » afin qu'il puisse résoudre le CSP décrit ci-avant ;
4. récupérer de l'agent « coordinateur » les consignes de volumes d'eau ϕ_{ix} , x pouvant désigner un bief aval dans le cas d'un écoulement gravitaire ou un bief amont dans le cadre de l'usage de pompes destinées à faire « remonter » des volumes d'eau ;
5. mettre en œuvre ces consignes pour la période $k + 1$.

L'agent coordinateur, quant à lui, devra à chaque période de temps k :

1. récupérer l'ensemble des $d_i(k)$ et des bateaux circulant sur les biefs ;
2. calculer pour chaque ϕ_{ij} du réseau les bornes l_{ij} et u_{ij} ;
3. résoudre le CSP formé par l'ensemble des variables ϕ_{ij} et d_i telles que : $dom(\phi_{ij}) = [l_{ij}, u_{ij}]$ et $\forall p, q \in N, d_i(k) = \sum \phi_{pi}(k) - \sum \phi_{iq}(k)$;
4. transmettre à chaque agent d'exploitation $i \in N$ l'ensemble des ϕ_{ix} à faire transiter vers les biefs x avec lequel il est connecté.

Les résultats de ce mécanisme de coordination peuvent être affinés en essayant de privilégier des solutions du CSP dans lesquels : la ressource en eau est économisée au mieux et les biefs sont au plus près possible de leur niveau normal de navigation (NNL). Deux contraintes additionnelles peuvent donc être considérées : $|d_i - V_i^{NNL}| = 0$ et $|\phi_{ij} - l_{ij}| = 0$. Celles-ci peuvent être utilisées soit pour sur-contraindre le CSP et procéder par relaxations successives pour obtenir une solution [Nouasse *et al.*, 2016], soit pour définir un COP.

Du fait de l'usage d'une approche par contraintes, il nous est possible d'analyser assez facilement la complexité temporelle du mécanisme de coordination. Celle-ci est en effet reliée à la résolution d'un CSP (X, D, C) pour laquelle la complexité est en $\Theta(|C|d^{|X|})$ avec $d = \max_{x_i \in X} |dom(x_i)|$. Dans le cas des voies navigables, il est assez simple d'imaginer le cas le plus défavorable (mais peu réaliste) : un réseau dans lequel chaque bief $i \in N$ est connecté à tous les autres et permettant à la fois un écoulement gravitaire des biefs amont vers les biefs aval et des transferts d'eau de l'aval vers l'amont via des pompes. Un tel réseau forme alors un graphe orienté complet à $N + 2$ nœuds (les N biefs plus les deux nœuds fictifs) et $(N + 2)(N + 1)$ arcs. Le CSP résultant dispose donc de $(N + 2) + (N + 2)(N + 1)$ variables et $(N + 2)(N + 1)$ contraintes. Le temps de calcul nécessaire à tous les agents du système à un instant k pour déterminer les consignes à appliquer à $k + 1$ s'exprime donc en $\Theta(N^2 p^{N^2})$ avec $p = \max_{i,j \in N} |[l_{ij}, u_{ij}]|$. La complexité dépend donc ici du nombre de biefs considéré et du pas de discrétisation utilisé pour représenter l'intervalle $[l_{ij}, u_{ij}]^2$.

Dans nos expérimentations, nous avons pu coordonner, avec des temps de calcul raisonnables (moins de 30 minutes), jusqu'à 12 biefs. Notons que nous considérons ce temps de calcul comme raisonnable au regard de la vitesse d'évolution d'un bief : compte tenu des tailles de biefs (généralement plusieurs kilomètres) et des temps de transfert des volumes d'eau entre deux biefs, l'état d'un bief est souvent considéré comme stable à l'échelle de l'heure. Dans un cas général, il faut donc ramener le temps de résolution du CSP à l'unité de temps considérée pour décrire l'évolution de l'environnement.

Pour certains types de problèmes, une coordination centralisée est parfois difficilement applicable :

- quantité trop importante de données remontées par les agents d'exploitation vers l'agent coordinateur au regard des capacités de communications de ceux-ci,
- problématiques de privacité lors de la centralisation des données,

2. Rappelons que les CSPs et ses variantes considèrent des variables discrètes et non continues.

— coût calculatoire assuré par un seul agent, etc.

Compte tenu de la nature distribuée de certains problèmes, il est souvent intéressant de considérer un modèle impliquant des variables et des contraintes distribuées entre agents. Dans la sous-section suivante, nous présentons l'usage de CSP distribués pour la coordination d'agents robotiques.

3.1.2 Coordination pour l'exploration multi-robot

L'exploration multi-robot impliquant des robots communicants nécessite de prendre en compte les aspects de connectivité lors de la coordination de la flotte. Cette connectivité peut être ponctuelle : les robots planifient entre eux des points de rendez-vous pour fusionner et mettre à jour leurs cartes de l'environnement. Elle peut être aussi permanente obligeant ainsi à maintenir un réseau de communication tout au long de la mission.

3.1.2.1 Exploration aux frontières et contraintes de connectivité

La problématique d'exploration multi-robot sous contraintes de communication consiste à faire évoluer dans un environnement inconnu une flotte de robots à l'intérieur de laquelle chaque membre est capable de communiquer avec le reste de la flotte et un centre de téléopération grâce à un réseau MANET (Mobile Ad Hoc Network) (figure 3.3).

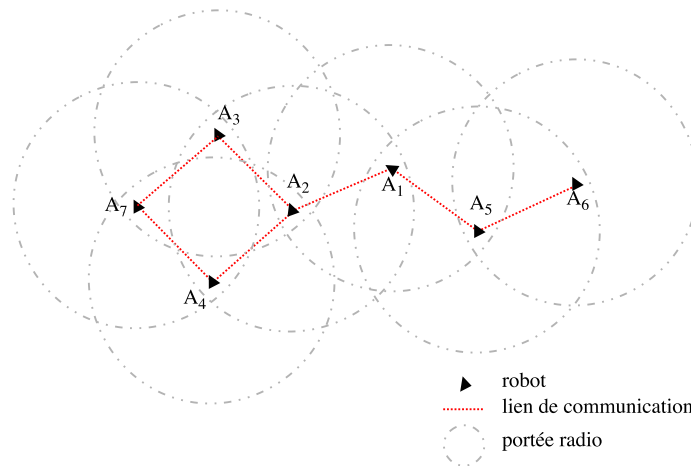


FIGURE 3.3 – Déploiement d'un réseau MANET sur une flotte de robots

Ce réseau doit être maintenu tout au long de la mission d'exploration de manière à ce que :

- les robots puissent s'échanger au cours de la mission des informations permettant à la flotte de disposer d'une carte à jour de l'environnement ;
- un opérateur humain puisse à tout moment visualiser l'état d'avancement de la mission (accès à la carte construite par les robots, accès aux images vues par un robot en particulier) et éventuellement assigner de nouveaux objectifs au groupe.

Suivant le principe d'exploration aux frontières ([Yamauchi, 1998], section 1.2.2.1), la flotte de robots doit donc se déployer au maximum, à la limite des zones de l'environnement encore inexplorées tout

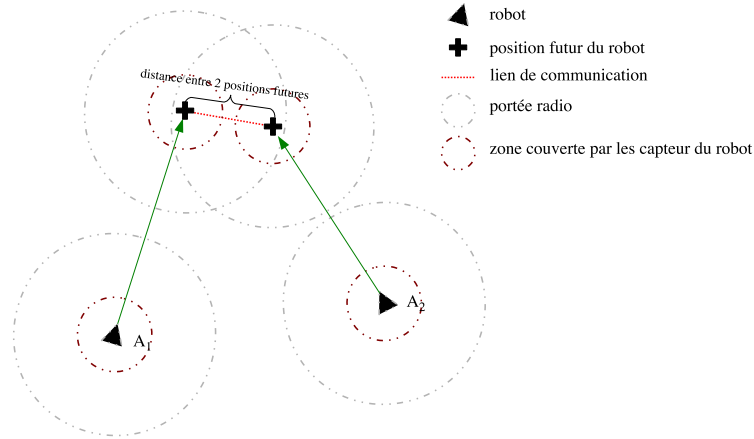


FIGURE 3.4 – Positions futures de deux robots devant se coordonner

en s’assurant de maintenir en place le réseau ad hoc.

La problématique de coordination sous-jacente à ce problème doit donc pouvoir garantir pour chaque robot la réalisation de 3 objectifs contradictoires :

- objectif 1 : ne pas être trop éloigné de manière à rester à portée radio des uns des autres ;
- objectif 2 : être suffisamment éloigné des autres robots pour limiter les recouvrements de zones analysées par les capteurs ;
- objectif 3 : se diriger de préférence vers des régions non encore explorées de l’environnement.

De manière à modéliser ce problème de coordination sous la forme d’un CSP distribué, nous nous fixons deux hypothèses [Doniec *et al.*, 2009] :

- les robots sont capables de se localiser dans l’environnement et d’échanger périodiquement leurs positions et leurs vitesses avec les autres robots de la flotte ;
- chaque robot communique également aux autres la cartographie des zones explorées de manière à maintenir une carte globale à jour pour l’ensemble de la flotte (en particulier la position de la frontière).

Par ailleurs, nous introduisons une discrétisation des actions en représentant les directions possibles de chaque robot sous la forme de points cardinaux : *N, NE, E, SE, S, SO, O, NO*. Connaissant l’ensemble des positions et vitesses de déplacement au sein de la flotte, chaque robot peut estimer pour chacune de ces directions la position future de n’importe quel robot de la flotte (figure 3.4).

Pour pouvoir évaluer si ce déplacement futur peut occasionner une rupture des communications au sein du MANET, il est nécessaire que chaque robot soit à chaque instant conscient de sa connectivité au reste de la flotte. Pour cela, nous utilisons un algorithme d’échange de messages [Le *et al.*, 2009] permettant à chaque robot de se construire une table de connectivité relative à un nœud de référence. Ce nœud peut être choisi parmi les robots en fonction de différents critères : portée wifi supérieure au reste de la flotte, niveau de batterie restant, etc. Nous démontrons dans [Le *et al.*, 2009] que si tous les robots d’une flotte peuvent maintenir un lien de communication (en connexion directe ou via d’autres robots) avec ce nœud de référence, alors la connectivité au sein du MANET est garantie. Ces tables de connectivité permettent en particulier à chaque robot de connaître les robots avec lesquels il doit maintenir une distance suffisamment proche pour garantir une communication avec le nœud de référence. Dans l’exemple de la figure 3.3, le robot A_1 est le nœud de référence. Les robots A_3 et A_7

peuvent construire ainsi les tables suivantes :

— A_3 :	Robot permettant l'accès au nœud de référence	Chemin d'accès complet
	A_2	(A_2, A_1)

— A_7 :	Robot permettant l'accès au nœud de référence	Chemin d'accès complet
	A_3	(A_3, A_2, A_1)
	A_4	(A_4, A_2, A_1)

Pour le robot A_7 , il a ainsi deux possibilités de communiquer avec A_1 : soit en passant par le robot A_3 , soit en passant le robot A_4 .

La construction et la mise à jour de ces tables tout au long de l'exploration permettent d'associer à chaque robot A_i un ensemble AR_{A_i} des robots permettant à A_i de rester connecter au MANET. Dans l'exemple précédent, nous aurions ainsi : $AR_{A_3} = \{A_2\}$ et $AR_{A_7} = \{A_3, A_4\}$.

L'objectif 1 de la coordination peut ainsi s'exprimer sous la forme d'une contrainte d'infériorité entre la distance entre les positions futures (fp) de deux robots (figure 3.4) et la portée radio cr utilisée au sein du MANET :

$$\{\forall A_i \in A, \exists A_j \in AR_{A_i} \text{dist}(fp(A_i, x_i), fp(A_j, x_j)) < cr\}$$

x_i désignant ici la direction future de l'agent A_i .

L'objectif 2 peut également s'exprimer en essayant d'imposer que cette même distance soit inférieure à deux fois le rayon sr de la zone couverte par les capteurs du robot :

$$\forall A_i \in A, \forall A_j \in A \setminus A_i, 2sr < \text{dist}(fp(A_i, x_i), fp(A_j, x_j))\}$$

Enfin, l'objectif 3 peut s'exprimer sous forme d'un ordonnancement des actions (donc des domaines) de chaque agent de manière à ce que la résolution du CSP distribué soit plutôt composé d'actions qui favorisent le rapprochement de la flotte vers la frontière avec les zones non explorées. Dans l'exemple de la figure 3.5, la direction NE serait clairement favorisée, suivie de S , SW , E et SE . Les directions N , NW et W seraient positionnées en fin de domaine du fait de la présence d'un obstacle entre le robot et la frontière³.

3.1.2.2 Coordination distribuée

La coordination de la flotte de robots va consister à résoudre une succession de CSP distribués définis par :

- $A = \{A_1, A_2, \dots, A_p\}$ l'ensemble des p robots de la flotte ;
- $X = \{x_1, x_2, \dots, x_p\}$ l'ensemble des variables correspondant aux déplacements à effectuer, $x_i (1 \leq i \leq p)$ étant le déplacement (direction) du robot A_i ;
- $D = \{dom(x_1), dom(x_2), \dots, dom(x_p)\}$ l'ensemble des domaines des variables de X où chaque $dom(x_i) (1 \leq i \leq p)$ reprend les 8 directions ordonnées suivant la relation d'ordre : $v_1 \preceq v_2 \equiv \text{dist}(fp(A_i, v_1), \text{frontiere}) < \text{dist}(fp(A_i, v_2), \text{frontiere}), (v_1, v_2) \in dom(x_i)^2$;
- $C = C_1 \cup C_2$ avec : $C_1 = \{\forall A_i \in A, \exists A_j \in AR_{A_i} \text{dist}(fp(A_i, x_i), fp(A_j, x_j)) < cr\}$ et $C_2 = \{\forall A_i \in A, \forall A_j \in A \setminus A_i, 2sr < \text{dist}(fp(A_i, x_i), fp(A_j, x_j))\}$

3. Notons que les directions vers lesquelles un obstacle est détecté ne sont pas éliminées des domaines puisque dans certains cas, elles pourraient constituer la seule possibilité de maintenir la connectivité.

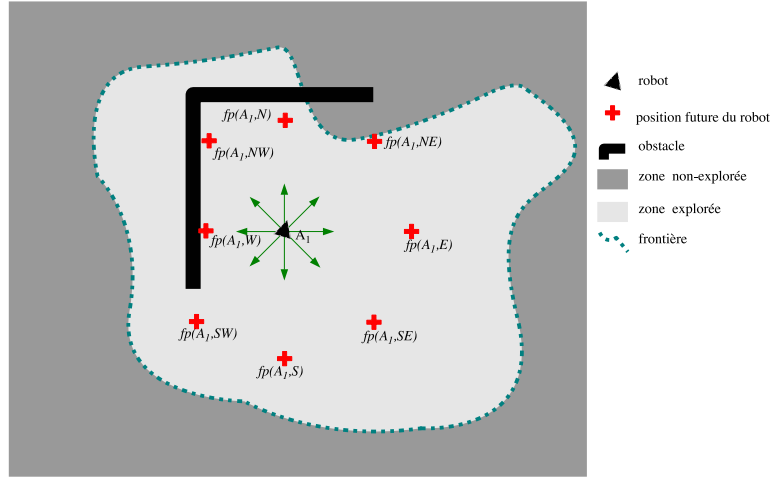


FIGURE 3.5 – Ordonnancement des déplacements possibles pour un robot

L’algorithme 3.1 présente la coordination du point de vue d’un agent A_i . A un instant t donné, ce mécanisme consiste à construire un CSP distribué sur la base des positions des différents robots et des tables de connectivité obtenues via l’algorithme décrit dans [Le *et al.*, 2009]; puis le résoudre pour que chaque robot obtienne une direction vers laquelle se déplacer durant un laps de temps dt . Dès que les robots commencent à opérer leur déplacement, un nouveau CSP distribué peut être construit et résolu de manière à fournir une nouvelle direction à $t + dt$ ⁴. Les lignes 7 à 16 décrivent la manière d’ordonner les actions des robots en tenant compte des éventuels obstacles : les directions sans obstacle sont privilégiées de manière à limiter les manœuvres de contournement et les éventuels blocages dans le cas d’obstacles convexes.

Ce mécanisme a été implémenté et évalué en simulation sur différents environnements plus ou moins structurés et plus ou moins denses en obstacles (figures 3.6). L’efficacité de la coordination a été évaluée en faisant varier le nombre de robots, la portée wifi et en mesurant le temps nécessaire à l’exploration. Sur la figure 3.7, nous pouvons observer que le temps d’exécution diminue avec le nombre de robots et la portée wifi.

La littérature sur la résolution des CSP distribués est relativement riche et de nombreux algorithmes ont été introduits ces quinze dernières années. Notre mécanisme de coordination pouvant fonctionner avec n’importe quel algorithme de résolution de CSP distribué, nous avons également réalisé une étude sur l’impact de l’algorithme de résolution sur l’exploration [Monier *et al.*, 2010, Monier *et al.*, 2011]. Pour cela, trois algorithmes ont été comparés : ABT [Yokoo *et al.*, 1992], AWS [Yokoo, 1995] et DBS [Doniec *et al.*, 2005]. Sur certaines métriques, nous avons constaté des différences notables.

Par exemple, sur la figure 3.8.a, nous pouvons observer que la distance totale parcourue par l’ensemble des robots est significativement inférieure pour les algorithmes ABT et DBS. Ceci peut s’expliquer par la gestion des priorités au sein des différents algorithmes. Tous les algorithmes de résolution de CSP distribués ont besoin d’instaurer un ordre entre les agents de manière à éviter les boucles infinies. Avec ABT et DBS, la priorité entre les robots est fixe (basée sur l’ordre lexicographique des

4. Le paramètre dt peut être ajusté de manière à laisser suffisamment de temps aux robots pour résoudre le disCSP.

ALGORITHME 3.1 - Algorithme de coordination distribué d'une flotte de robots

```

1 : begin
2 :   while qu'une frontière existe do
3 :     mise à jour de la carte et des tables de connectivité
4 :      $X_{A_i} \leftarrow \{x_i\}$ 
5 :      $C_{A_i} \leftarrow \{dist(fp(A_i, x_i), fp(A_j, x_j)) < cr\}_{A_j \in AR_{A_i}}$ 
6 :      $C_{A_i} \leftarrow C_{A_i} \cup \{2 sr < dist(fp(A_i, x_i), fp(A_j, x_j))\}_{A_j \in A \setminus A_i}$ 
7 :      $direction \leftarrow \{N, NE, E, SE, S, SW, W, NW\}$ 
8 :      $\bar{H} \leftarrow \emptyset$ 
9 :     for each  $h \in direction$  do
10 :      if il existe un obstacle dans la direction  $h$  then
11 :         $\bar{H} \leftarrow \bar{H} \cup \{h\}$ 
12 :      end
13 :    end
14 :     $H \leftarrow direction \setminus \bar{H}$ 
15 :     $dom(x_i) \leftarrow \{v_a \in H / \forall (v_a, v_b) \in H^2, a < b \Rightarrow v_a \preceq v_b\}$ 
16 :     $dom(x_i) \leftarrow dom(x_i) \cup \bar{H}$ 
17 :    résoudre le CSP distribué
18 :    avancer en suivant la direction  $x_i$  pendant  $dt$  secondes
19 :  end
20 : end

```

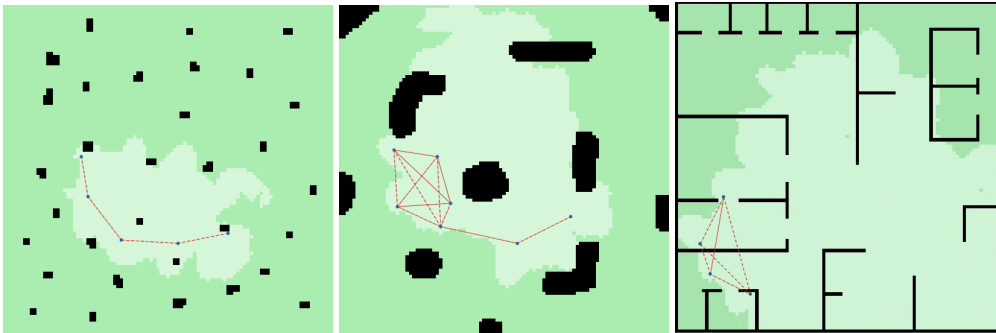


FIGURE 3.6 – Expérimentations sur différents types d'environnements

identifiants de robots) et ne change pas pour tous les CSP distribués résolus au cours de l'exploration. Ainsi le robot de plus haute priorité « guide et influence » la résolution, ce qui tend à lui donner dans le contexte d'exploration multi-robot un rôle de leader. A l'inverse, AWS ne requiert pas de définir au préalable un ordre sur les agents, celui-ci étant déterminé dynamiquement en cours de résolution. Par conséquent, entre deux résolutions successives de CSP distribués, le « leader » peut changer conduisant ainsi à des oscillations de déplacement de la flotte entre deux points opposés de la frontière.

Le nombre de cycles nécessaires à chaque résolution (figure 3.8.c) ainsi que le nombre de contraintes vérifiées (figure 3.8.b) sont également fortement différents d'un algorithme à l'autre, mais n'ont que peu d'incidence sur la coordination. En revanche, le nombre de messages échangés (figure 3.8.e) peut avoir un impact significatif sur l'application. Un algorithme qui fera circuler beaucoup de messages

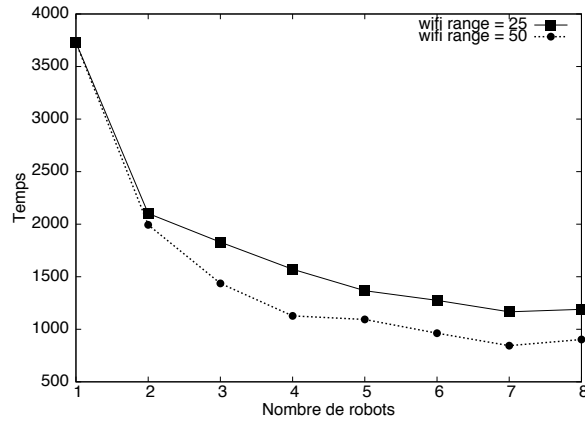


FIGURE 3.7 – Expérimentations : durée de l’exploration en fonction du nombre de robots

sur le réseau aura tendance à plus solliciter les batteries des robots. Ici encore, sur ce critère, AWC se montre moins adapté que les deux autres.

L’ensemble des résultats obtenus sur les différentes expérimentations montrent que la coordination multi-robot pour l’exploration sous contrainte de connectivité est un excellent benchmark pour les algorithmes de résolution de CSP distribués.

3.1.3 Analyse des approches de coordination par contraintes

Pour conclure cette section, nous proposons d’analyser les mécanismes de coordination présentés suivant 4 aspects : la généralité, la complexité, l’adaptabilité aux environnements évolutifs, l’anticipation.

Les problèmes de satisfaction de contraintes constituent un formalisme bien connu en IA. Son expressivité et la richesse des algorithmes de résolution permettent aux mécanismes de coordination d’être extrêmement génériques et de s’appliquer à de nombreux domaines. Grâce aux différentes extensions des CSP, les mécanismes de coordination peuvent être mis en œuvre avec une prise de décision centralisée sur un seul agent « coordinateur » ou complètement distribuée sur l’ensemble des agents du système.

L’usage d’un modèle à base de contraintes offre également la possibilité d’analyser facilement la complexité du mécanisme de coordination résultant. Une telle analyse est beaucoup plus difficile à réaliser sur un mécanisme de coordination ad hoc. Le calcul de complexité détaillé sur l’exemple des voies navigables montre des temps de calcul exponentiels en nombre d’actions et d’agents ; avec en pratique la possibilité de coordonner jusqu’à une douzaine d’agents. Au-delà, il est possible de réduire le nombre d’actions considérées par un agent, mais l’usage de ces mécanismes pour des systèmes large échelle (de l’ordre de la centaine d’agents) reste difficilement envisageable.

Analysons ensuite comment ce type de coordination peut s’accommoder d’environnements dy-

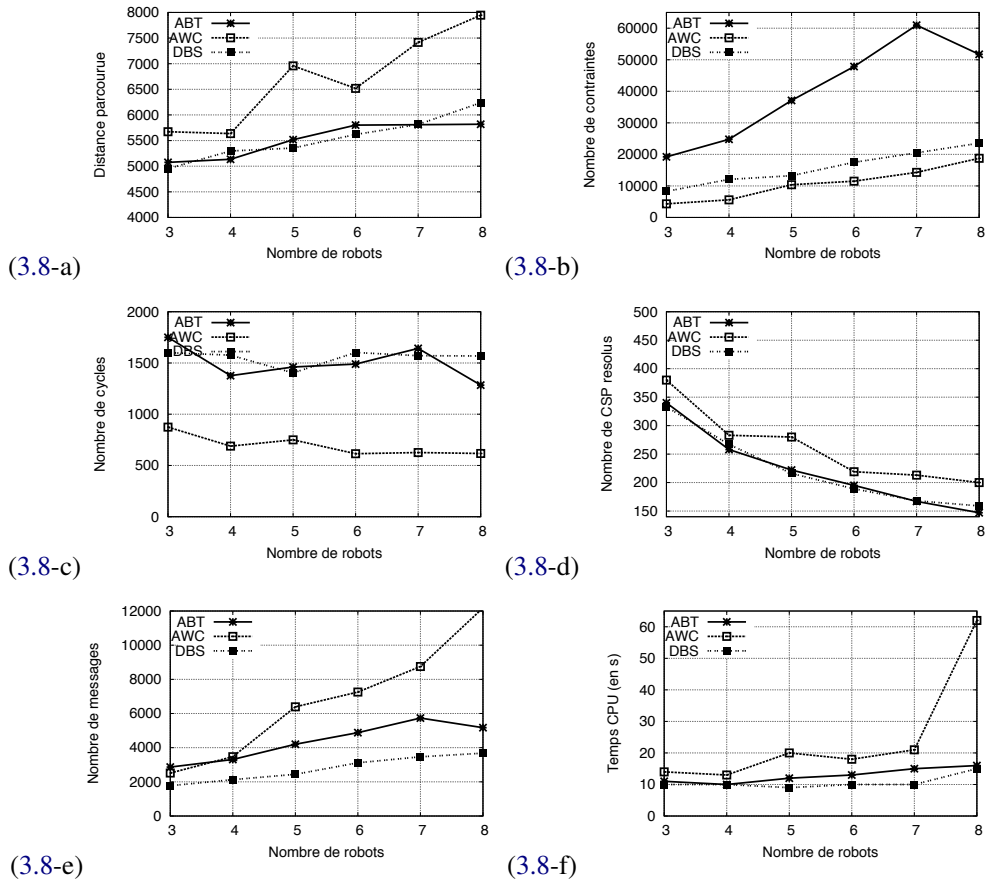


FIGURE 3.8 – Expérimentations : variation du nombre de robots coordonnés

namiques et évolutifs. Les évolutions de l'environnement peuvent être prises en compte entre deux résolutions successives de CSP puisque les décisions des agents à exécuter à l'instant t sont calculées sur la base des informations perçues par les agents à $t - 1$. De fait, toute évolution du système pourra être prise en compte au fur et à mesure que les agents se coordonnent. Ce type de coordination permet donc d'être relativement réactif aux aléas qui peuvent venir perturber les agents. Malheureusement, cette réactivité peut occasionner des discontinuités et des oscillations dans la prise de décision des agents. En effet, sur plusieurs pas de temps successifs, les CSP construits peuvent être légèrement différents (du point de vue des contraintes ou des domaines) et le solveur peut donner une solution différente sur deux pas de temps consécutifs. Dans le cas de l'exploration multi-robot, cela pourrait par exemple se traduire par un robot de dirigeant au nord à l'instant t , pour ensuite se diriger vers le nord est à $t + 1$, puis revenir vers le nord à $t + 2$, etc. Lorsqu'une solution trouvée à t reste valide à $t + 1$, elle doit donc être privilégiée par le solveur pour éviter ce type d'oscillations.

Pour terminer, nous pouvons observer qu'une coordination résultant de résolutions successives de CSP ne permet pas l'anticipation sur le long terme : l'obtention d'une solution à un instant t ne prend pas en compte les effets de cette solution sur la construction des CSP à $t + 1$, $t + 2$, etc. La seconde

partie de ce chapitre s'intéresse donc à la prise de décision séquentielle et aborde la coordination sous l'angle de la planification markovienne.

3.2 La coordination vue sous l'angle de la planification

D'un point de vue mono-agent, la prise de décision séquentielle (également appelée planification) fait référence à la capacité à construire une séquence d'actions qui permettra à l'agent d'aboutir à un objectif fixé. Dans un contexte multi-agent, la planification consiste à obtenir à partir des plans individuels de chaque agent un plan commun décrivant le comportement global du système.

Les premiers travaux dans le domaine des SMA se sont focalisés sur la synchronisation de plans individuels [Georgeff, 1983] puis par la suite sur la construction de plans partagés [Mandiau, 1993]. Plus tard, la planification a été utilisée pour la coordination en environnements fortement dynamiques : application multi-robot [Alami *et al.*, 1998], missions aériennes [Marc, 2005], production industrielle [Giordani *et al.*, 2013], l'intelligence ambiante [Casals *et al.*, 2018]. Les formalismes utilisés dans ces travaux sont variés, allant des simples graphes aux automates hybrides [El Fallah-Seghrouchni *et al.*, 2004] en passant par les modèles STRIPS [Fikes et Nilsson, 1971], PDDL [Ghallab *et al.*, 1998] ou encore ANML [Smith *et al.*, 2008].

La planification mono-agent a, quant à elle, été très étudiée dès les années 60 en utilisant le formalisme des processus de décision markoviens (MDP)[Howard, 1960] dans lequel un agent dispose d'un ensemble d'actions lui permettant d'aboutir (avec une certaine probabilité) à différents états auxquels sont associés des récompenses. Toute sa prise de décision repose sur une maximisation de ces récompenses sur un horizon donné. Grâce à la distribution probabiliste sur les états, les MDP peuvent facilement prendre en compte les événements incertains et ainsi proposer des plans robustes aux aléas. Au début des années 2000, des travaux ont étendu les MDP au contexte multi-agent [Boutilier, 1999], mais contrairement aux approches de planification évoquées plus haut, les MDP et leurs extensions peinent à passer à l'échelle du fait de la complexité calculatoire.

Moyennant certaines hypothèses, elles peuvent tout de même être utilisées dans un contexte réel. Dans la suite, nous présentons deux approches de coordination d'agents à base de modèles markoviens : l'une appliquée à la production décentralisée d'électricité introduite en section 1.2.3 et l'autre à la gestion de la ressource en eau dans les voies navigables présentée en section 1.2.4.

3.2.1 Planification pour le problème d'engagement d'unités (UCP)

Le problème d'engagement d'unités (Unit Commitment Problem (UCP) en anglais) consiste à planifier la production électrique de centrales multi-sources de manière à satisfaire une demande en électricité établie sur un horizon donné (en général 24h). Comme cela a été évoqué au chapitre 1, les techniques de stockage de l'énergie sont à l'heure actuelle encore peu efficaces et relativement coûteuses. Ceci oblige par conséquent à réaliser une planification « en flux tendu » répondant au plus juste à la demande.

Cette planification doit tenir compte des contraintes techniques inhérentes aux centrales multi-sources. Par exemple, les sources contrôlables telles que les groupes diesel ou les turbines à gaz ne peuvent être éteintes avant d'avoir atteint une certaine température. De la même manière, ces mêmes sources requièrent un temps de refroidissement à prendre en compte dans le cas d'un éventuel

redémarrage.

Pour toutes ces raisons, les méthodes qui consisteraient à simplement optimiser indépendamment sur chaque tranche horaire la production de chaque centrale sans tenir compte des effets à long terme de cette production ne sont pas adaptées. Il faut donc anticiper la production en fonction de la demande prévisionnelle et de l'état futur de chaque source. L'UCP consiste donc à définir à moindre coût : (i) les séquences d'allumage et d'extinction de chaque source sur toute la période considérée, (ii) la production de chaque source en marche, de manière à satisfaire la demande en électricité tout en respectant les contraintes techniques des sources.

3.2.1.1 Modélisation mathématique du problème

Notons i une source de production contrôlable sur un intervalle de temps : $[0, t_{max}]$. Lorsque celle-ci est allumée, sa capacité de production est comprise dans l'intervalle discret $[P_i^{min}, P_i^{max}]$. Cette production est soumise à des contraintes d'inertie. L'extinction d'une source ne peut se faire immédiatement après son allumage : pour des raisons de fiabilité, elle doit rester allumée au minimum pendant T_i^{ON} intervalles de temps. De même, une durée minimale d'arrêt T_i^{OFF} doit être respectée avant un nouvel allumage. Les coûts de production des sources contrôlables sont en pratique essentiellement liés au démarrage, à la consommation de carburant durant la production d'énergie et à l'émission de polluants.

A partir de ces éléments, l'UCP se définit comme un n-uplet $(U, P, T, Y, D, \rho_Y, \rho_D, \rho, c)$ pour N sources de production contrôlables, k unités de production non contrôlables (des éoliennes par exemple) avec un horizon de planification H . Dans ce n-uplet :

- U désigne l'ensemble des états d'engagement u . Nous définissons $\vec{u}_h = (u_1^h, \dots, u_N^h)$ comme étant le vecteur d'états d'engagement dans lequel $u_i^h \in \{0, 1\}$ représente l'état d'allumage (on/off) d'une unité i à un instant $h = 0, 1, \dots, H$.
- P est l'ensemble des profils de production p . Nous définissons $\vec{p}_h = (p_1^h, \dots, p_N^h)$ comme étant le vecteur de profils de production où chaque $p_i^h \in \mathbb{R}^+$ représente la puissance générée d'une unité i à un instant $h = 0, 1, \dots, H$.
- T est l'ensemble des périodes t de temps de fonctionnement d'une unité. Nous définissons $\vec{t}_h = (t_1^h, \dots, t_N^h)$ comme étant le vecteur de temps opérationnel dans lequel chaque $t_i^h \in \mathbb{Z} \cap [-T_i^{OFF}, T_i^{ON}]$ représente le temps opérationnel d'une unité i à un instant h .
- Y est l'ensemble des réalisations y des sources de production non contrôlables obtenues à partir de la distribution $\rho_Y(\vec{y}^{h+1} | \vec{y}^h)$. Nous notons $\vec{y}^h = (y_1^h, \dots, y_K^h)$ le vecteur de réalisations où chaque $y_i^h \in \mathbb{R}^+$ représente la réalisation effective d'une source de production non contrôlable k à l'instant h .
- D est l'ensemble des demandes à satisfaire $d^h \in \mathbb{R}^+$ par les sources non contrôlables, obtenues à partir de la distribution $\rho_D(d^{h+1} | d^h)$ à l'instant h .
- $\rho^i : T_i \times U_i \mapsto T_i$ est la fonction de transition spécifiant le prochain état $t_i^{h+1} = \rho^i(t_i^h, u_i^h)$ de i étant donné t_i^h et u_i^h . Cette fonction de transition permet d'exprimer les contraintes opérationnelles de fonctionnement telles que :

$$\rho^i(t_i^h, u_i^h) = \begin{cases} t_i^h & \text{SI } t_i^h = T_i^{ON}, u_i^h = 1 \\ -1 & \text{SI } t_i^h = T_i^{ON}, u_i^h = 0 \\ +1 & \text{SI } t_i^h \leq -T_i^{OFF}, u_i^h = 1 \\ t_i^h + 1 & \text{SI } 0 < t_i^h < T_i^{ON} \\ t_i^h - 1 & \text{SI } -T_i^{OFF} < t_i^h < 0 \end{cases}$$

Par exemple, si une source est en fonctionnement depuis T_i^{ON} unités de temps ($t_i^h = T_i^{ON}$)

et si elle est allumée à l'instant h : $u_i^h = 1$ alors son état reste inchangé. Si en revanche, elle est éteinte à : $u_i^h = 0$ alors qu'elle était en fonctionnement depuis T_i^{ON} unités de temps, l'état passe à -1 et décrémentera jusqu'à atteindre $-T_i^{\text{OFF}}$ sans pouvoir redémarrer.

- $c(\vec{t}, \vec{p}, \vec{u})$ est la fonction de coût dépendant des vecteurs : \vec{u} , \vec{t} et \vec{p} . c est une fonction additive, somme des coûts locaux de chaque unité i : $c_i(t_i^h, p_i^h, u_i^h) = f_i(p_i^h) + \ell_i(t_i^h, u_i^h)$. La fonction $f_i(p_i^h)$ est quadratique et représente le coût de fonctionnement (consommation de carburant) pour produire une puissance p_i . La fonction $\ell_i(t_i^h, u_i^h)$ représente la fonction de coût associée aux changements d'état (arrêt et démarrage) de l'unité de production i . La fonction de coût c est assujettie à deux types contraintes locales :

$$C1 : \forall p_i^h, P_i^{\min} \leq p_i^h \leq P_i^{\max} \quad (3.2)$$

qui s'assure que la puissance à produire par i respecte bien les capacités de i et :

$$C2 : \forall h, \forall \vec{p}_h, \forall \vec{y}_h, g(\vec{p}_h, \vec{y}_h, d^h) = \sum_{i=1}^N p_i^h + \sum_{k=1}^K y_k^h - d^h = 0 \quad (3.3)$$

impliquant de satisfaire la demande sur chaque pas de temps de l'horizon de planification.

3.2.1.2 Approches possibles de coordination pour la résolution distribuée de l'UCP

La résolution de l'UCP a été abordée dans la littérature sous l'angle de la recherche opérationnelle puis plus récemment sous l'angle des DCOP [Kumar *et al.*, 2009, Miller *et al.*, 2012]. Mais comme indiqué en fin de section 3.1.3, de telles approches ne permettent pas de prendre en compte les effets sur le long terme des actions calculées. Le contexte sensible des réseaux électriques nous enjoint à nous tourner vers des approches séquentielles capables d'anticiper sur un horizon temporel les états futurs du système.

La mise en place d'une approche de planification pour l'UCP sous-tend deux tâches distinctes : le calcul d'une politique π (c'est-à-dire une application associant à chaque état du système une action à réaliser) et l'exploitation de cette dernière.

Dans la littérature, deux types de planification sont généralement étudiés : la planification centralisée et la planification distribuée. La planification centralisée repose sur l'utilisation d'un agent central capable de collecter l'ensemble des informations individuelles et propres à chaque source i de sorte à centraliser les données du problème et à le résoudre. À l'inverse, la planification distribuée requiert une participation au calcul des agents en charge des sources et repose sur une succession d'échanges de messages permettant de converger vers une solution.

À l'issue de la tâche de planification, les agents en charge des sources peuvent exploiter la politique résultante. L'exploitation (également appelée contrôle) repose sur l'observation de l'état courant à partir duquel est sélectionnée l'action prescrite par la politique π .

Lorsque π est séparable en politiques individuelles pour chaque source i , il est possible de contrôler chaque unité indépendamment les unes des autres ; autrement dit la sélection de l'action (y_i^h, p_i^h) ne nécessite pas la connaissance des états conjoints de toutes les sources, seule l'observation de l'état de i est nécessaire. On parle alors d'un contrôle réparti. Un tel contrôle permet de s'affranchir des communications entre les sources nécessaires pour reconstruire un état conjoint. Bien sûr, l'obtention d'une politique séparable est souvent coûteuse en termes de temps de calcul.

À l'inverse, l'obtention d'une politique non séparable requiert un temps de calcul plus raisonnable. Mais avec une telle politique, la sélection de l'action appropriée requiert la connaissance de l'état conjoint des sources i . Cette connaissance requiert de la part des sources une communication de l'état observé vers un agent central qui disposera alors de l'état conjoint des sources et pourra sélectionner l'action conjointe appropriée pour enfin la diffuser à l'ensemble des sources. On parle alors de contrôle centralisé.

Le projet AgentVVP étant un projet exploratoire, il a été décidé d'un commun accord au sein du consortium de privilégier de développement d'une planification entièrement distribuée pour un contrôle réparti des sources de production d'une centrale virtuelle multi-sources. Dans la suite, nous présentons donc une méthode de coordination distribuée, basée sur les processus de décision markoviens, permettant à des agents contrôlant des unités de production i de résoudre l'UCP.

3.2.1.3 Formalisation de l'UCP sous forme d'un MMDP

Parmi les différents formalismes markoviens existants (pour une présentation des différentes familles de processus markoviens, nous invitons le lecteur à se référer aux ouvrages suivants : [Sigaud et Buffet, 2008a], [Sigaud et Buffet, 2008b]), les MMDP (processus de décision markoviens multi-agents) semblent les plus adaptés. Formellement, un MMDP [Boutilier, 1999] est un tuple $(\mathcal{I}, S, A, P, R)$ où \mathcal{I} représente un ensemble d'agents collaborant à l'optimisation d'un certain critère. Dans ce modèle, S désigne l'ensemble des états possibles, A l'ensemble des actions, P une fonction de transition $|S| \times |A| \times |S| \rightarrow [0, 1]$ reprenant les probabilités de transition d'un état s à un état s' en utilisant l'action a , R^a associe à cette action une fonction de coût $|S| \times |A|$ pour son exécution dans l'état s .

Résoudre un MMDP consiste à rechercher une politique π permettant d'optimiser un critère de performance (minimisation d'un coût par exemple ou maximisation d'un profit) sur un horizon de planification donné H . Comme pour la plupart des extensions des MDP, le calcul de cette politique est sujet à une explosion combinatoire pour de grandes instances de problèmes.

Dans de nombreux scénarios, chaque agent i peut contrôler son propre processus de décision markovien complètement observable (S^i, A^i, T^i) qui décrit sa dynamique locale. Cela signifie que sa dynamique est indépendante de celle des autres agents. Dans ce cas, le MMDP peut être réécrit comme suit : $(\mathcal{I}, \otimes_i S^i, \otimes_i A^i, \otimes_i T^i, R)$, où l'ensemble des états conjoints S correspond au produit vectoriel sur les états locaux S^i et similairement pour les ensembles A et T . Sous cette hypothèse, un MMDP est plus facilement manipulable. La complexité se trouve également encore réduite lorsque la fonction de coût globale est décomposable sous la forme d'une somme des fonctions de coût locales à chaque agent $i : R^i : S^i \times A^i \mapsto \mathbb{R}$ et dépendant uniquement des états et actions de ce dernier. Lorsque ces deux hypothèses sont réunies, le MMDP est séparable et chaque agent peut résoudre séparément son MDP local comme un problème indépendant tout en communiquant avec les autres pour construire une politique jointe optimale.

Dans notre cas applicatif, le MMDP $(\mathcal{I}, S, A, P, R)$ serait défini par :

- \mathcal{I} l'ensemble des unités de production pilotées par les agents i ;
- S l'ensemble fini d'états $s^h = (s_0^h, s_1^h, \dots, s_N^h)$ tel que $S = T \times Y \times D$. Chaque état local $s_i^h = t_i^h$ de l'agent i correspond à la durée de fonctionnement t_i^h pour tous les intervalles de décisions $h = 0, \dots, H$ et tous les agents $i = 0, \dots, N$. L'état local $s_0^h = (y^h, d^h)$ consiste en un vecteur de réalisation y^h et d'une demande d^h . Nous notons également S_0, \dots, S_N

l'ensemble des états locaux.

- A l'ensemble des actions jointes $a^h = (a_0^h, a_1^h, \dots, a_N^h)$ tel que $A = U \times P$. Chaque action locale $a_i^h = (u_i^h, p_i^h)$ d'un agent i consiste en un état u_i^h (ON / OFF) et un profil de puissance à générer p_i^h . De plus, nous notons A_0, \dots, A_N l'ensemble des actions locales.
- $P : S \times A \times S \rightarrow [0, 1]$ la fonction de transition état à état où $P(s'|s, a)$ est la probabilité d'être dans l'état joint $s' = (t', \rho_Y(y', y), \rho_D(d', d))$ après avoir choisi l'action jointe $a \in A$ dans l'état joint $s = (t, y, d)$ pour tous les $t' \in P(\rho(t, u))$.
- $R : S \times A \rightarrow \mathbb{R}^+$ décrit le coût $R(s^h, a^h)$ associé à l'action a^h effectuée dans l'état s^h . Ce coût est égal à la somme des coûts locaux $\sum_{i=1}^N R_i(s_i^h, a_i^h, i)$.

La résolution du MMDP consiste à trouver une politique jointe, non stationnaire et séparable $\pi = (\pi^0, \pi^1, \dots, \pi^H)$. Chaque règle de décision jointe $\pi^h : S \rightarrow A$ fait correspondre les états joints s^h aux actions jointes a^h . Le but est de choisir une politique jointe π permettant de minimiser la somme de coûts sur un horizon de planification : $\pi = \arg \min_{\pi} E(\sum_{h=0}^H \sum_{i=1}^N R_i(s_i^h, \pi_i[s_i^h]) | s_0, \pi)$ tout en respectant les contraintes $C1$ et $C2$. Dans les approches classiques de résolution de MMDP, l'hypothèse est généralement faite que A et S sont finis et discrets. Ce n'est pas le cas ici, pour l'UCP puisque les actions de A sont des couples où les u_i^h sont discrets, mais les p_i^h sont définis sur \mathbb{R} . Par ailleurs, le calcul de la politique est rendu difficile par la présence de la contrainte $C2$ (équation 3.3) qui est globale à tous les agents.

Une analyse de la fonction de coût fait toutefois apparaître deux propriétés remarquables que nous pouvons exploiter. D'une part, la fonction est quadratique, ce qui offre une large palette de méthodes d'optimisation utilisables. D'autre part, la fonction est séparable (somme de fonctions locales), ce qui permet d'envisager une distribution du processus de planification sur différents agents. La méthode que nous proposons dans la suite s'inspire de l'algorithme DGO initialement proposé par [Tsitsiklis *et al.*, 1986].

3.2.1.4 Adaptation de l'algorithme DGO

L'algorithme DGO est une méthode de résolution distribuée de problèmes d'optimisation quadratique s'appuyant sur un ensemble d'agents contrôlant chacun un sous-ensemble des variables de décision du problème. La version présentée ci-après est adaptée à notre problème : à savoir minimiser une fonction f représentant ici la somme des coûts définie plus haut. Chaque agent i cherche à déterminer $\pi_i^{h,\tau}$ définissant une action a_i^h à exécuter dans un état s_i^h . L'algorithme est itératif : chaque agent cherche à modifier ses propres variables de décision en gardant les valeurs déjà affectées par les autres agents. Cela revient pour chaque agent à résoudre un problème d'optimisation local. L'algorithme DGO original utilise pour ce faire une descente de gradient classique. A la fin d'un cycle τ , les agents communiquent les valeurs assignées à leurs propres variables de décision avec les autres agents du système. L'algorithme continue jusqu'à ce que deux valeurs assignées soient identiques sur deux itérations consécutives. La dernière assignation de chaque variable constitue alors une solution optimale compte tenu de la convexité de la fonction objective.

Pour pouvoir traiter des problèmes d'optimisation quadratique sous contraintes d'égalité (telles que la contrainte C_2 précédemment formulée), il faut changer la descente de gradient classique de DGO par une descente de gradient projetée [Boyd et Vandenberghe, 2004] dans laquelle chaque solution intermédiaire est projetée dans l'espace des solutions admissibles. Cela permet ainsi de s'assurer que les solutions trouvées seront toujours consistantes avec la contrainte C_2 .

Concernant les contraintes C_1 , leur prise en compte dans la projection obligerait à recalculer cette

ALGORITHME 3.2 - Algorithme DGO [Tsitsiklis *et al.*, 1986] exécuté pour un agent i

1 : Calcul de la solution initiale $\pi_i^{h,\tau}$ avec $\tau = 0$
2 : Communiquer la solution initiale aux agents $j \neq i$
2 : **while** $\pi_i^{h,\tau+1} = \pi_i^{h,\tau}$ **do**
3 : Calcul de la direction de la descente : $\beta_i^{h,\tau} = -\nabla f$
4 : Calcul du pas : $\alpha_i^{h,\tau} = \frac{(\beta_i^{h,\tau})^\top (\beta_i^{h,\tau})}{(\beta_i^{h,\tau})^\top (\nabla^2 f) (\beta_i^{h,\tau})}$
5 : Mise à jour de la solution courante : $\forall j, \forall s_j^h, \pi_{i,j}^{h,\tau+1}(s_j^h) \begin{cases} \pi_{i,j}^{h,\tau+1}(s_j^h) & \text{si } i \neq j, \\ \pi_{i,i}^{h,\tau}(s_j^h) + \alpha_i^{h,\tau}(s_j^h) \cdot \beta_i^{h,\tau}(s_j^h) & \text{sinon.} \end{cases}$
6 : Communiquer la solution mise à jour aux agents $j \neq i$
7 : $\tau = \tau + 1$
7 : **end**

dernière à chaque itération de DGO, ce qui induit des coûts calculatoires potentiellement prohibitifs. Pour contourner ces coûts, nous pouvons remarquer que les contraintes C_1 sont locales et propres à chaque source de production. Chaque agent peut donc essayer de satisfaire localement ses contraintes de type C_1 tout en se coordonnant avec les autres agents via un protocole permettant de gérer les éventuels conflits.

Le protocole proposé dans [Dibangoye *et al.*, 2015] repose sur une énumération des différents types de conflits pouvant apparaître entre agents. Posons $\underline{\mathcal{I}}_j^{h,\tau}$ l'ensemble des agents i dont le profil à générer $p_{i,j}^{h,\tau}$ (τ étant une itération de l'algorithme DGO) partagé avec l'agent $j \notin \underline{\mathcal{I}}_j^{h,\tau}$ est le profil minimal de l'agent i : $p_i^{h,\tau}$. De la même manière, nous définissons $\bar{\mathcal{I}}_j^{h,\tau}$ l'ensemble des agents i dont le profil à générer $p_{i,j}^{h,\tau}$ partagé avec l'agent $j \notin \bar{\mathcal{I}}_j^{h,\tau}$ est le profil maximum de l'agent i : $\bar{p}_i^{h,\tau}$.

Nous exprimons la condition de terminaison de l'algorithme en définissant la notion d'agent terminal. Un agent i est dit terminal si en déviant son profil de génération courant, le vecteur de profil de génération est soit sous-optimal soit non consistant avec les contraintes C_1 et C_2 . Autrement dit, un agent i sera terminal si et seulement si l'une des trois conditions suivantes est vérifiée :

- $i \in \underline{\mathcal{I}}_i^{h,\tau}$
- $i \in \bar{\mathcal{I}}_i^{h,\tau}$ et $\underline{\mathcal{I}}_i^{h,\tau} = \emptyset$
- $\beta_i^{h,\tau} = 0$ et $\bar{\mathcal{I}}_i^{h,\tau'} = \underline{\mathcal{I}}_i^{h,\tau'} = \emptyset, \forall \tau' \geq \tau$

Ainsi, si un agent i est terminal alors son profil de production sera :

- $p_i^{h,\tau}$ si $i \in \underline{\mathcal{I}}_i^{h,\tau}$
- $\bar{p}_i^{h,\tau}$ si $i \in \bar{\mathcal{I}}_i^{h,\tau}$ et $\underline{\mathcal{I}}_i^{h,\tau} = \emptyset$
- $\bar{p}_i^{h,\tau}$ si $\beta_i^{h,\tau} = 0$ et $\bar{\mathcal{I}}_i^{h,\tau'} = \underline{\mathcal{I}}_i^{h,\tau'} = \emptyset, \forall \tau' \geq \tau$

Dès qu'un agent découvre qu'il est un agent terminal, il partage cette information incluant son profil de génération avec les agents restants. Ensuite, il attend jusqu'à ce que les autres agents restants terminent à leur tour pour obtenir le vecteur de profil de génération de l'ensemble des agents. Lorsqu'un agent terminal reçoit tous les messages de terminaison des autres agents, il met fin au calcul.

Pour la résolution des conflits, nous définissons la notion de problème résiduel $\mathcal{I} \setminus \underline{\mathcal{I}}_i^{h,\tau}$ (respectivement $\mathcal{I} \setminus \bar{\mathcal{I}}_i^{h,\tau}$) comme une restriction du problème original sur les variables associées aux agents de l'ensemble $\mathcal{I} \setminus \underline{\mathcal{I}}_i^{h,\tau}$ (respectivement $\mathcal{I} \setminus \bar{\mathcal{I}}_i^{h,\tau}$). Illustrons cette notion de problème résiduel en consi-

dérant trois agents devant se répartir une demande d^τ). Supposons que l'agent 1 soit terminal avec un profil $p^{1\tau}$, le problème résiduel consiste à répartir la demande restante $d^{\tau+1} = d^\tau - p_1^\tau$ entre les agents 2 et 3.

Les conflits apparaissent lorsqu'un agent non terminal dispose d'un vecteur de profil qui ne satisfait pas les contraintes globales et $\mathcal{I}_i^{h,\tau} = \emptyset$. Supposons qu'un agent j soit non terminal et dispose d'un vecteur de profil $\tilde{p}_j^{h,\tau}$ tel que :

- les contraintes globales soient non satisfaites
- $\mathcal{I}_i^{h,\tau} \neq \emptyset$

Le profil $p_j^{h,\tau+1}$ peut alors être obtenu en résolvant le problème résiduel sur $\mathcal{I} \setminus \mathcal{I}_h, \tau + 1^j$.

Dans la suite, nous noterons P-DPGO l'adaptation de DGO avec le protocole de communication décrit ci-avant.

3.2.1.5 Expérimentations

P-DPGO a été testé sur différents scénarios en utilisant la plateforme « Energie répartie » du L2EP (<https://www.epmlab.eu/index.php/en/>). Le cœur de cette plateforme est constitué d'un simulateur temps réel de réseaux électriques (RT-Lab [Dufour *et al.*, 2005]) s'exécutant sur un ensemble de PC reliés à un réseau Ethernet. Celui-ci permet de reproduire le comportement d'une grande variété de composants d'un réseau : générateurs contrôlables (groupes diesel, turbines à gaz), systèmes de production d'énergie renouvelable (éoliennes, panneaux photovoltaïques), convertisseurs de puissance, réseaux de transport haute et basse tension, etc. La supervision du réseau est assurée par le logiciel PCVue qui permet de collecter et visualiser la dynamique de la simulation. P-DPGO a été implémenté en utilisant la plateforme de développement Jade (Java Agent DEvelopment framework) [Bellifemine *et al.*, 2007]. Pour reproduire au mieux les conditions de décentralisation, chaque agent est déployé sur un des PC faisant tourner le simulateur. Les trois outils : RT-Lab, PCVue et Jade sont reliés à un serveur OPC qui permet l'échange des données. Pour coller au plus près de conditions réelles de déploiement, chaque agent possède son propre client OPC pour récupérer ses données. La figure 3.9 donne une vue d'ensemble de l'architecture utilisée pour les expérimentations. P-DPGO reçoit une demande en électricité à satisfaire à partir d'un certain nombre de sources de production. Il calcule (hors-ligne) une politique qui associe à chaque état conjoint du système un profil de génération pour chaque source. Cette politique prend en compte tous les états possibles des sources y compris les pannes. L'exploitation de la politique se fait en ligne en fonction de l'état courant observé pour chaque source de production.

Les expérimentations se basent sur un réseau électrique classiquement utilisé dans la littérature : le IEEE 14 nœuds sur lequel ont été ajoutées des centrales virtuelles multi-sources (CMS). La figure 3.10 présente un exemple d'implantation de 3 CMS. Ces centrales sont toutes composées d'éoliennes et de groupes diesel. Toutes les éoliennes sont identiques et ont une capacité de production de 2 MW.

Trois modèles de groupe diesel sont présents. Leurs caractéristiques⁵ diffèrent, mais ils ont tous un profil de génération p^i compris entre : $0.10p_{max}^i \leq p^i \leq 0.95p_{max}^i$. Les fonctions de coût de chaque groupe sont quadratiques et les groupes présentant les capacités de production les plus élevées sont les plus coûteux à utiliser (cf. tableau 3.1).

Les principales sources d'incertitude qui peuvent apparaître lors de la production par les centrales

5. reprises de groupes diesel existants au sein du catalogue du constructeur Caterpillar

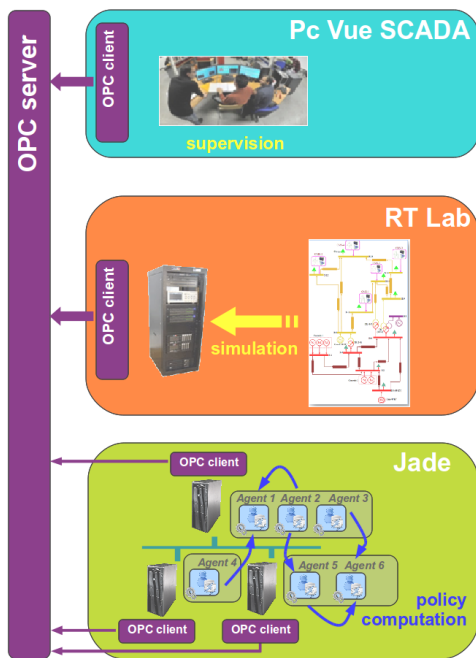


FIGURE 3.9 – Plateforme « Energie répartie »

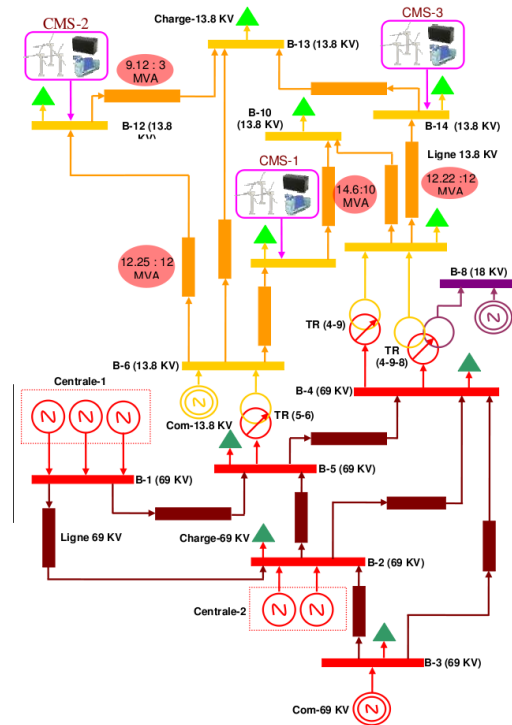


FIGURE 3.10 – Réseau IEEE 14 nœuds

p_{\max}^i	coût d'utilisation en fonction du profil de génération p^i	T^{ON}	T^{OFF}
10800 W	$5.13p^{i2} - 10.19p^i + 29.53$	80 min	80 min
6300 W	$5.00p^{i2} - 10.00p^i + 29.72$	80 min	80 min
5400 W	$4.94p^{i2} - 9.92p^i + 29.79$	80 min	80 min

TABLEAU 3.1 – Caractéristiques des groupes diesel

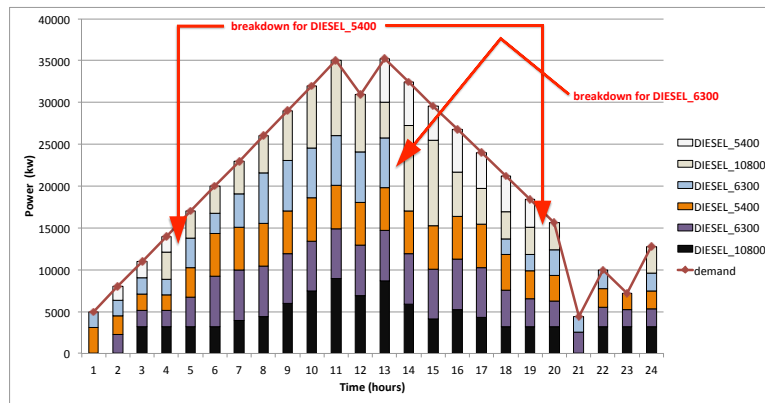


FIGURE 3.11 – Utilisation de l'algorithme P-DPGO avec 6 sources de production et l'apparition de pannes

multi-sources sont :

- les erreurs de prédiction météorologique qui peuvent conduire à surestimer la capacité de production des éoliennes
- les pannes des groupes diesel qui peuvent survenir après de longue période de fonctionnement ou d'arrêt

Notons que dans les deux cas, il s'agit de pallier à un défaut de production et de trouver dans la politique générée une alternative s'appuyant sur les sources de production restantes.

Différents scénarios ont été testés [Dibangoye *et al.*, 2015]. L'un d'eux introduit des pannes sur différents groupes diesel. La figure 3.11 présente le résultat de l'application d'une politique calculée par P-DPGO sur un horizon de planification de 24h pour répondre à une demande. Cette demande résulte d'une demande initiale à laquelle a été retranchée la part produite en éolien. Chaque histogramme présente la contribution des différents groupes diesel sur une heure. Trois pannes surviennent sur la période : deux pannes à répétition (à $h = 5$ et $h = 20$) pour le groupe diesel 5400 et une panne à $h = 14$ pour le groupe diesel 6300. La figure 3.12 présente le coût de l'exploitation de la politique dans le cas normal (sans panne) et dans le cas des trois défaillances précédemment décrites. Sans surprise, ce deuxième cas engendre un coût de production légèrement supérieur du fait du recours à des groupes diesel moins économiques que ceux tombés en panne.

D'autres scénarios ont consisté à faire varier l'horizon de planification h afin de montrer l'intérêt d'une décision séquentielle. Dans de nombreux cas, on observe une différence en termes de coût : plus l'horizon de planification est grand, plus la politique est riche en solutions utilisant les groupes diesel les moins onéreux. La figure 3.13 illustre ce résultat en présentant l'écart de coût pour une utilisation de P-DPGO avec $h = 1$ (ce qui signifie que la politique est recalculée toutes les heures pour une durée d'une heure) et $h = 24$. Pour certaines demandes, l'usage d'un horizon de planification petit

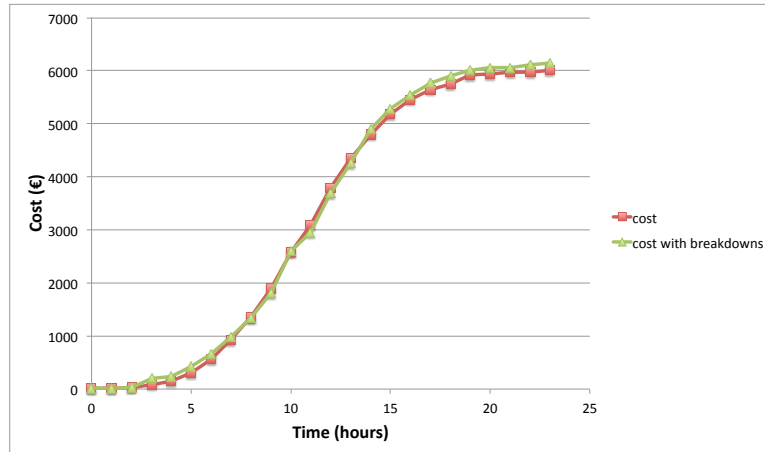


FIGURE 3.12 – Coût cumulé de la production

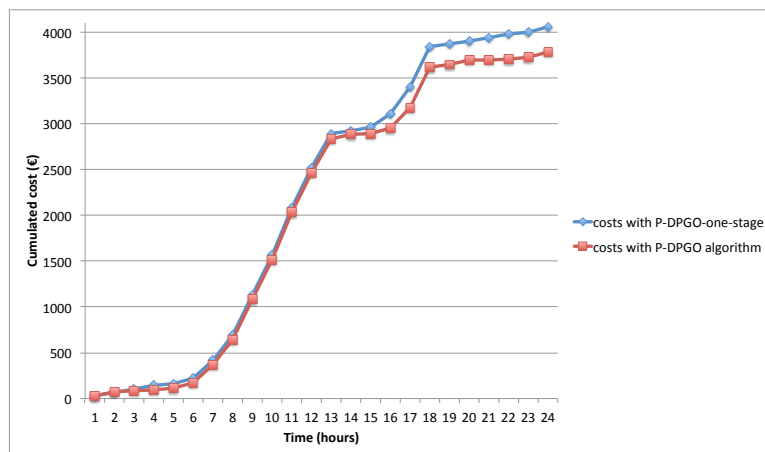


FIGURE 3.13 – Différence de coûts pour des horizons de planification $h = 1$ et $h = 24$

peut conduire à des défauts de production. C'est le cas à la figure 3.14 où P-DPGO a été utilisé avec $h = 1$ (ce qui revient pour l'algorithme à n'avoir aucune visibilité sur la demande future). Sur ce scénario à 3 sources de production, la demande augmente brusquement entre 15h et 16h. La demande étant relativement faible à 15h, la politique calculée pour cette tranche horaire (sans connaissance de l'augmentation de la demande 1h plus tard) préconise l'utilisation d'un seul groupe diesel (5400) pour la satisfaire. A 16h, seuls les groupes diesel 6300 et 5400 sont utilisables et il est trop tard pour démarrer le dernier groupe. La demande étant supérieure à la capacité des deux groupes diesel démarrés, celle-ci ne peut pas être satisfaite.

Comme toutes les approches de planification markovienne, le passage à l'échelle de P-DPGO dépend du nombre d'états du problème considéré. Celui-ci est égal au produit cartésien des états locaux de chaque source de production. Ces états locaux dépendent directement des caractéristiques physiques des sources. Dans le cas d'un groupe diesel i , c'est la discrétisation (plus ou moins fine) de l'intervalle $[-T_i^{OFF}, T_i^{ON}]$ qui va conditionner ce nombre d'états locaux. Nous pouvons également

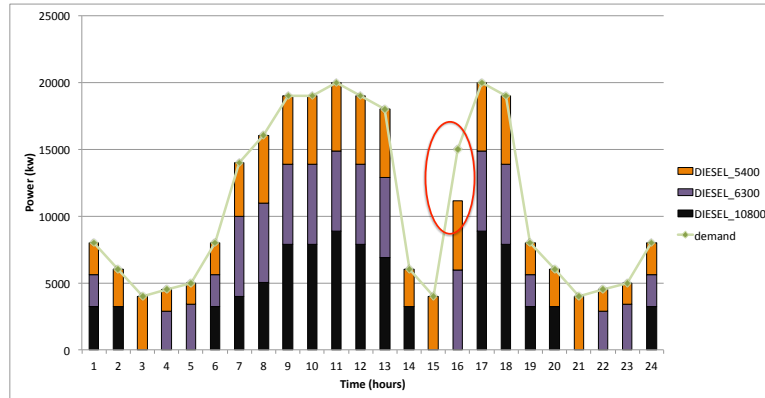


FIGURE 3.14 – Non-satisfaction de la demande avec un horizon de planification $h = 1$

noter que la distribution physique des agents sur des machines différentes n'accélère pas le calcul de la politique, elle permet juste de respecter la nature distribuée de l'UCP dans le cadre des Smart Grid et engendre, par construction du protocole de communication, un grand nombre de messages échangés. Par exemple, dans le scénario à 6 groupes diesel présentés en figure 3.11, pas moins de 5 millions de messages échangés ont été nécessaires pour converger vers la politique.

En pratique, le passage à l'échelle d'approches markoviennes nécessite de pouvoir exploiter des hypothèses et des structures particulières de l'application visée. Dans la sous-section suivante, nous présentons la problématique de gestion de la ressource en eau dans les voies navigables ; application pour laquelle la structure du problème permet l'utilisation d'une sous-classe des MDP.

3.2.2 Planification pour la gestion de la ressource en eau dans les voies navigables

Nous avons vu en sections 1.2.4 et 3.1.1, que la navigation sur un réseau de canaux requiert de maintenir le niveau d'eau entre une limite haute et une limite basse (*HNL* et *LNL*). Par ailleurs, du fait de la présence d'écluses permettant de compenser les déclivités naturelles, la navigation est consommatrice d'eau. De ce fait, le nombre de bateaux empruntant le réseau peut être considéré comme une demande à satisfaire. Pour pouvoir répondre à cette demande, le gestionnaire dispose d'un ensemble d'ouvrages d'art (vannes, barrages, pompes) répartis sur le réseau et lui permettant de déplacer de l'eau à l'intérieur de celui-ci. L'ensemble de ces écluses, vannes, barrages et pompes constituent donc des points de transfert entre les biefs du réseau.

La demande de navigation (i.e. nombre de bateaux empruntant le réseau) étant en général connue par le gestionnaire, il est possible de planifier à l'avance la meilleure suite d'actions en chaque point de transfert du réseau pour satisfaire la demande tout en prenant en compte des événements extérieurs au système (pluie, sécheresse, rejets d'eau sauvage, etc) pouvant venir affecter les niveaux d'eau.

Compte tenu des tailles de biefs (parfois plusieurs centaines de kilomètres), des volumes d'eau échangés, de leur temps de transfert et de propagation dans le réseau, des vitesses de déplacement des bateaux, il est possible de considérer un horizon de planification relativement lointain (de l'ordre de la semaine) avec des pas de temps assez grands (de l'ordre de la demi-journée) permettant de tenir compte de périodes de navigation le jour et d'inactivité la nuit.

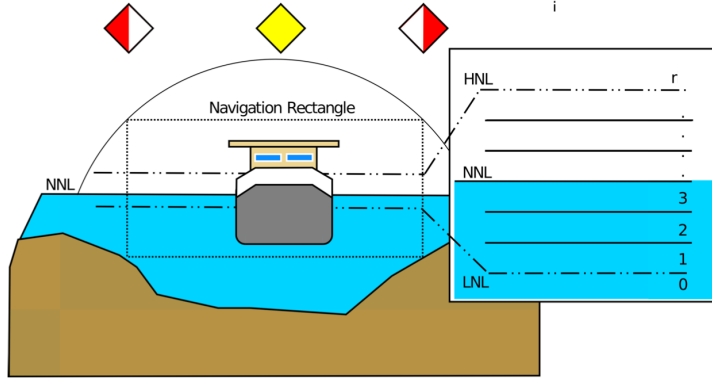


FIGURE 3.15 – Discrétisation d'un bief

3.2.2.1 Modélisation mathématique du problème

De manière à planifier sous incertitude, nous utilisons le formalisme des MDP classiques pour la modélisation mathématique du problème ⁶.

Pour cela, chaque bief du réseau est discrétisé de manière à pouvoir représenter l'ensemble des états du système. Les volumes d'eau contenus dans chaque bief sont donc divisés en intervalles réguliers entre LNL et HNL, augmentés de deux intervalles de taille infinie représentant les valeurs en dehors du rectangle de navigation (figure 3.15). Une discrétisation similaire se retrouve au niveau des différents points de transfert du réseau (pour des raisons de simplification et d'homogénéisation, les transferts d'eau au niveau de ces points sont considérés de manière volumique plutôt que sous forme de débit). Les perturbations pouvant être subies par les biefs sont dépendantes du temps et nécessitent donc d'intégrer le temps dans l'état du système.

L'ensemble S des états du modèle est donc la combinaison de tous les intervalles possibles de chaque bief pour chaque pas de temps :

$$S = \{0, \dots, T\} \times \prod_{i=1}^N d_i$$

où d_i représente le volume d'eau contenu dans le bief i ⁷ discrétisé sous forme d'intervalles :

$$d_i = \{] - \infty, V_i^{LNL} [, [V_i^{LNL}, V_i^{LNL} + p], \dots, [V_i^{HNL} - p, V_i^{HNL} [,]V_i^{HNL}, +\infty [\}$$

avec p un pas de discrétisation.

De manière similaire, l'ensemble A des actions possibles du système se définit comme le produit des intervalles de volume des points de transferts. Ici les actions sont indépendantes du temps, cela nous donne donc l'ensemble :

6. Par souci de simplicité, nous reprenons une bonne partie des notations introduites en section 3.1.1.

7. Par rapport à la modélisation introduite en section 3.1.1, nous considérons directement le volume en absolu et non en relatif par rapport au volume du niveau normal V_i^{NNL} .

$$A = \prod_{\substack{i=0 \\ j=0}}^N \phi_{ij}$$

avec ϕ_{ij} représentant un volume discret transféré entre le bief i et le bief j .

La discrétisation des ϕ_{ij} est établie de manière à s'assurer que l'exécution d'une action a , éventuellement combinée à une perturbation v , garantisse d'être dans trois états possibles : l'intervalle attendu s' ou au pire l'intervalle juste avant : s'^- ou juste après s'^+ . La probabilité de transition se définit donc comme suit :

$$P(s'|s, a + v) = \begin{cases} p_+ & si & s + a + v \in s'^+ \\ p_- & si & s + a + v \in s'^- \\ 0 & sinon \end{cases}$$

avec :

- $s, s' \in S$,
- $a \in A$,
- v une perturbation,
- p_+ la probabilité d'être dans l'intervalle attendu,
- p_- la probabilité d'être dans l'intervalle juste après,
- p_- la probabilité d'être dans l'intervalle juste avant,
- $p_+ + p_+ + p_- = 1$.

La fonction de transition globale est ensuite définie par :

$$T(s, a, s') = \sum_{v \in \mathcal{V}} \left(P(v) \times \prod_{s \in S} P(s'|s, a + v) \right)$$

Cette formalisation naturelle sous forme d'un MDP montre rapidement ses limites en termes de passage à l'échelle puisqu'en considérant un réseau à 8 biefs et 7 écluses avec une discrétisation en neuf intervalles pour les biefs et six pour les écluses, la taille de la fonction de transition est de l'ordre de 10^{21} [Desquesnes *et al.*, 2016a].

Pour pouvoir passer à l'échelle, il est donc nécessaire de contourner ces limitations spatiales en considérant des extensions et sous-classes des MDP classiques. Dans la littérature, deux familles d'approches sont connues pour le passage à l'échelle : les approches de factorisation et les approches de distribution. Les MDP factorisés [Boutilier *et al.*, 1995] qui visent à représenter de manière plus compacte les fonctions de transition et de récompense n'ont pas été retenus dans la thèse de Guillaume Desquesnes [Desquesnes, 2018]. En effet, dans une modélisation factorisée, l'évolution de chaque bief devrait être calculée de manière séparée, ce qui ne correspond pas à la réalité physique du problème. A l'inverse, les approches visant à distribuer le MDP initial en sous-MDP ont retenu notre attention et en particulier les travaux de [Nair *et al.*, 2005] qui visent à exploiter les propriétés de problèmes « en réseau » pour faciliter la distribution de calculs. L'algorithme OCLP proposé ci-après s'inspire de cette approche.

3.2.2.2 Proposition de l'algorithme OCLP

Les travaux de Nair [Nair *et al.*, 2005] ont introduit une classe particulière nommée ND-POMDP (Networked Distributed Partially Observable MDP) dans laquelle :

- les transitions et les observations sont indépendantes (chaque agent a une connaissance complète de son environnement),
- la fonction de récompense est représentée sous la forme d'un graphe dans lequel les nœuds correspondent aux agents et les arcs expriment des récompenses locales dépendantes des états et actions des deux agents impliqués.

L'approche proposée par Nair (LID-JESP) pour résoudre cette classe de problème s'inspire de l'algorithme DBA (Distributed Breakout Algorithm) [Yokoo et Hirayama, 2001] utilisé dans les CSP distribués. Suivant le principe de coévolution utilisé dans DBA, chaque agent tente d'améliorer sa politique en respectant les autres politiques proposées par son voisinage. Au démarrage de l'algorithme, chaque agent débute avec une politique initiale quelconque. Au fur et à mesure des politiques reçues de ses voisins, chaque agent peut calculer l'amélioration de sa politique locale en évaluant les états atteignables avec les actions jointes possibles (issues des politiques du voisinage). Si l'amélioration calculée est la plus grande du voisinage, alors l'agent retient cette politique et la diffuse à ces voisins.

Les ND-POMDP et l'algorithme LID-JESP ne sont pas applicables directement au cas des voies navigables puisque le problème ne peut être divisé en parties dont les évolutions sont totalement indépendantes : l'évolution d'un bief dépend en grande partie du bief amont. La thèse de Guillaume Desquesnes introduit donc une nouvelle approche, nommée OCLP dans laquelle les capacités de contrôle du réseau sont distribuées sur un ensemble d'agents possédant une vue locale du ou des biefs qu'il gère [Desquesnes *et al.*, 2016c, Desquesnes *et al.*, 2017c]. Contrairement à LID-JESP, les agents n'ont pas de connaissance complète : ils optimisent localement leur politique en se coordonnant avec les autres agents. La coordination entre agents reprend les principes initiés par l'algorithme DBA : le calcul des politiques locales est itératif et se base sur le principe de coévolution qui consiste ici à n'autoriser la modification que d'une seule politique par voisinage et itération (ceci afin de limiter des modifications simultanées et potentiellement conflictuelles).

L'algorithme OCLP se présente donc sous la forme d'un protocole de communication permettant aux agents de converger vers une politique globale satisfaisant les contraintes locales de chaque bief.

Un agent α exécutant l'algorithme va commencer par initialiser sa politique locale π_{α_0} ainsi qu'un compteur qui servira de condition d'arrêt. Ce compteur est initialisé avec la plus grande distance (en nombre de voisins) séparant deux agents au sein du système. Ensuite, à chaque itération it , l'agent α échange sa politique courante avec son voisinage⁸. Comme deux agents n'ont pas forcément les mêmes états locaux, la politique transmise par α à un autre agent β de son voisinage est adaptée à la connaissance commune entre les deux agents. Cette adaptation consiste à associer à chaque état partagé par α et β une distribution des probabilités sur les actions pouvant être exécutées par α . En parallèle l'agent α recevra également de son voisinage des politiques adaptées. Ces politiques adaptées lui permettront de mettre à jour son MDP local noté MDP_{α}^{it} . Cette mise à jour consiste à redéfinir la fonction de transition en fonction des distributions de probabilités sur les actions des politiques reçues. Sur la base de ce nouveau MDP, l'agent va recalculer une politique locale optimale π' et estimer le gain de cette nouvelle politique par rapport à la politique courante π_{α}^{it} . Ce gain est calculé à l'aide d'une simple fonction heuristique soustrayant les fonctions de valeur des deux politiques.

8. Notons que l'approche OCLP est synchrone : l'exécution d'une itération requiert des points de synchronisations nécessaires : échange des politiques et des gains qui mettent en attente les agents si tous les messages ne sont pas arrivés.

ALGORITHME 3.3 - OCLP pour un agent α

```

1 : Création d'une politique locale initiale  $\pi_\alpha^0 : S_\alpha \rightarrow A_\alpha$ 
2 :  $d \leftarrow$  distance maximale entre deux agents
3 :  $counter \leftarrow d$ 
4 :  $it \leftarrow 0$ 
5 : répéter
6 :   pour chaque agent  $\beta$  du voisinage de  $\alpha$ 
7 :     adapter  $\pi_\alpha^{it}$  à la connaissance commune entre  $\alpha$  et  $\beta$  pour obtenir  $\pi_{\alpha,\beta}^{it}$ 
8 :     envoyer  $\pi_{\alpha,\beta}^{it}$  à  $\beta$ 
9 :   pour chaque agent  $\beta$  du voisinage de  $\alpha$ 
10 :    recevoir  $\pi_{\beta,\alpha}^{it}$  de  $\beta$ 
11 :    mettre à jour  $MDP_\alpha^{it}$  en  $MDP_\alpha^{it+1}$  en utilisant les politiques reçues
12 :    calculer  $\pi'_\alpha$  la politique optimale de  $MDP_\alpha^{it+1}$ 
13 :     $g_\alpha^{it} \leftarrow$  gain( $\pi'_\alpha, \pi_\alpha^{it}$ ) de  $MDP_\alpha^{it+1}$ 
14 :    échanger  $g_\alpha^{it}$  avec chaque voisin
15 :     $G_\alpha^{it} \leftarrow$  l'ensemble des gains du voisinage
16 :    si  $g_\alpha^{it} = \max G_\alpha^{it}$  alors
17 :       $\pi_\alpha^{it} \leftarrow \pi'_\alpha$ 
18 :    sinon  $\pi_\alpha^{it} \leftarrow \pi_\alpha^{it}$ 
19 :    si  $g_\alpha^{it} > 0$  alors
20 :       $counter \leftarrow d$ 
21 :    sinon  $counter \leftarrow counter - 1$ 
22 :    échanger  $counter$  avec chaque voisin
23 :     $Counters^{it} \leftarrow$  l'ensemble des compteurs du voisinage
24 :     $counter \leftarrow \max Counters^{it}$ 
25 :     $it \leftarrow it + 1$ 
26 : jusqu'à  $counter = 0$ 

```

Le gain g_α^{it} est ensuite échangé par les agents de manière à ce que chaque agent dispose de l'ensemble G_α^{it} des gains de son voisinage. Si l'agent α se retrouve être celui à avoir proposé le gain le plus élevé alors il conserve sa politique améliorée π'_α tandis que les autres agents du voisinage restent sur leur politique courante. De même, si le gain est positif l'agent α remet son compteur à la valeur d , sinon il est décrémenté de 1. Les compteurs sont échangés entre voisins et ceux-ci ne conservent que le plus grand du voisinage. L'algorithme s'arrête pour l'agent α lorsque son compteur atteint 0. Sa politique courante sera alors celle qu'il exploitera pour gérer son bief durant l'horizon de planification.

3.2.2.3 Expérimentations

L'algorithme OCLP a été évalué sur différents réseaux soumis à diverses conditions (sécheresse, étiage, événements pluvieux imprévus). Nous présentons dans la suite une des expérimentations menées dans le cadre de la thèse de Guillaume Desquesnes. Le réseau considéré est un sous-ensemble du réseau des voies navigables des Hauts-de-France, composé de 7 biefs [Desquesnes *et al.*, 2017b]. Deux d'entre eux (le bief Douai-Don-Cuinchy et Cuinchy-Fontinettes), de par leur position géographique et leurs caractéristiques physiques, sont extrêmement sensibles aux variations de niveaux d'eau et sont régulièrement sujets à des inondations (cf. figure 1.7). Leur volume optimal (correspondant au>NNL) est autour des 9 millions de m^3 et l'écart toléré avec le LNL ou le HNL n'est que de l'ordre de 1,5 %.



FIGURE 3.16 – Localisation du réseau considéré

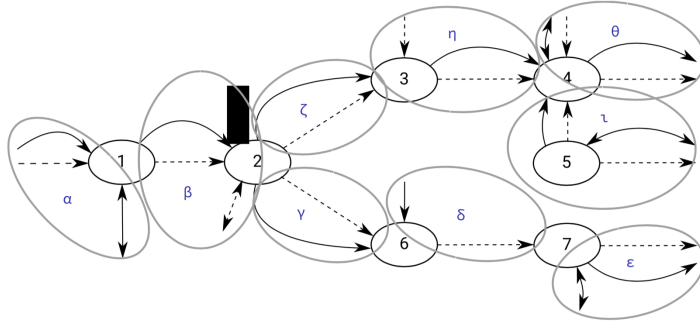


FIGURE 3.17 – Décomposition en agents du réseau considéré

La figure 3.16 présente la localisation du réseau considéré. Celui-ci est décomposé en 9 agents, noté de α à ι et représenté sous forme d'ovales gris sur la figure 3.17. Fonction des capacités de transfert des biefs et de leur discrétisation, le nombre d'actions pour ces 9 agents passe de 100 pour l'agent γ à 1634 pour l'agent ι . L'horizon de planification considéré est d'une demi-semaine (soit 7 périodes de 12 heures). Avec une discrétisation des biefs en une douzaine d'intervalles, le nombre d'états par agent varie entre 1054 (pour l'agent ι) et 1632 (pour l'agent γ). Ce réseau, avec une approche centralisée basée sur un MDP classique, implique une modélisation faisant apparaître 2×10^8 états et 2×10^{23} actions.

Trois différents scénarios ont été considérés avec ce réseau :

- scénario 1 : pas d'évènement climatique extrême et pas de trafic la nuit permettant ainsi au réseau de restabiliser l'ensemble des biefs autour du NNL (ce scénario correspond aux contraintes actuelles d'exploitation des réseaux par VNF)
- scénario 2 : conditions climatiques normales et trafic autorisé la nuit avec une répartition du trafic équivalente entre le jour et la nuit (ce scénario futuriste est envisagé par VNF pour répondre à une possible hausse du trafic)
- scénario 3 : trafic autorisé la nuit et grosse perturbation pluvieuse (de l'ordre de 230000 m^3) sur le bief 2

Sur les trois scénarios, nous considérons un état initial du bief au niveau NNL.

La figure 3.18 présente l'exploitation d'une politique calculée dans le cadre du scénario 1. Chaque bief (de 1 à 7) est associé à une courbe exprimant l'écart relatif au NNL durant 7 pas de temps de 12 heures. Sur l'axe des ordonnées, le 0 correspond ainsi au NNL, +100% d'écart correspond au HNL et -100% d'écart au LNL. Le bief 6 est ainsi celui qui fluctue le plus autour de son NNL : il atteint presque 50% d'écart relatif au pas de temps 1 et dépasse les 25% aux pas de temps 3 et 6. Sur l'ensemble des biefs, l'écart moyen est de l'ordre de 11% avec des écarts.

Dans un contexte de navigation en continu (scénario 2) qui peut sembler un peu plus compliqué, nous pouvons observer sur la figure 3.19 que finalement les résultats sont également bons (voire légèrement meilleurs : 8% d'écart au NNL en moyenne). Ces résultats laissent à penser que la période de repos de nuit était finalement une contrainte pour certains biefs vis-à-vis des échanges contrôlés avec l'extérieur du réseau.

Sur le troisième scénario, la quantité d'eau considérée peut amener le niveau du bief 2 à sortir du rectangle de navigation. Deux planifications ont été testées : une première dans laquelle aucune per-

3.2. La coordination vue sous l'angle de la planification

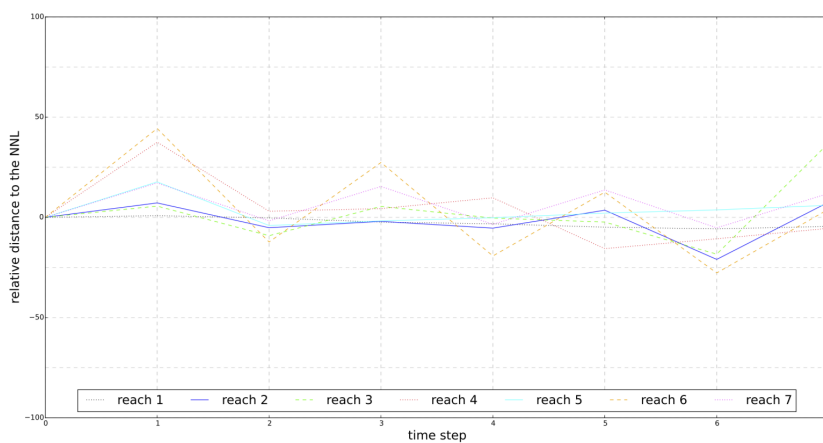


FIGURE 3.18 – Scénario 1 : navigation autorisée sur 12h et pas de pluie

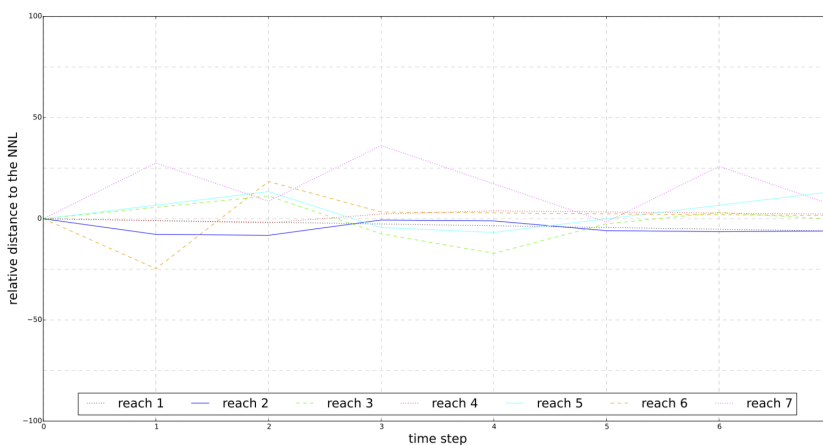


FIGURE 3.19 – Scénario 2 : navigation autorisée sur 24h et pas de pluie

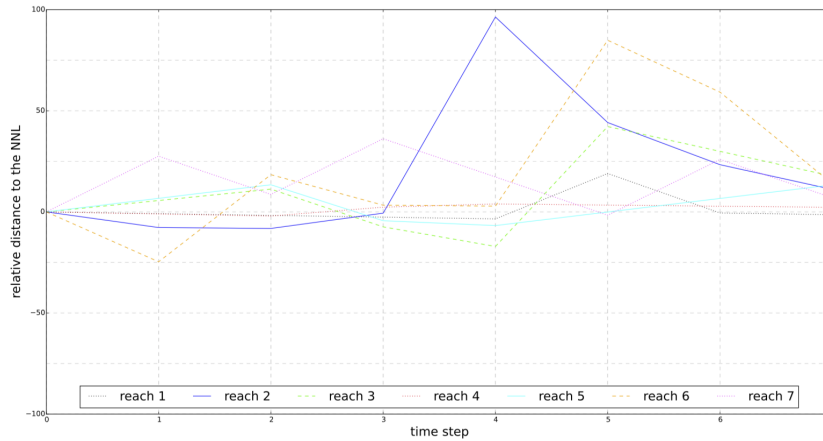


FIGURE 3.20 – Scénario 3 : navigation autorisée sur 24h et occurrence d'un évènement pluvieux intense non anticipé

turbation v n'est prévue dans le modèle (voir la probabilité de transition P présentée en 3.2.2.1) et une seconde où la perturbation est anticipée et intégrée au modèle. La figure 3.20 présente l'exploitation de la première planification. Sans anticipation du phénomène pluvieux, le bief 2 est fortement affecté. On note ainsi qu'au pas de temps 4, celui-ci atteint la limite de navigation haute (HNL). L'exploitation de la politique parvient à ramener le niveau vers le NNL à partir du pas de temps 5 avec pour conséquence d'augmenter le niveau du bief avant (bief 6). Avec anticipation, on retrouve la même dynamique de rétablissement avec un impact moins fort (figure 3.21). Lorsque cette même politique est utilisée et que l'évènement pluvieux ne se produit pas (figure 3.22), le bief 2 se vide dans les biefs aval, mais sans pour autant les perturber ; cela afin que le bief reste à une distance raisonnable de son NNL.

L'algorithme OCLP étant itératif, nous nous sommes intéressés à l'évolution de la « qualité » de la politique au fur et à mesure des itérations successives de l'algorithme. La figure 3.23 a été obtenue en enregistrant des politiques jointes disponibles à la fin de chaque itération et les exploitant en simulation. Ces enregistrements ont été faits pour 50000 simulations avec des conditions initiales à chaque fois différentes et choisies aléatoirement. Nous obtenons ainsi l'évolution du coût d'exploitation de la politique jointe (qui représente en fait sa « qualité ») en fonction de la durée de fonctionnement de l'algorithme (exprimée en itérations). Nous constatons que la politique jointe s'améliore de manière notable sur les 15 premières itérations. Les itérations suivantes continuent d'améliorer la qualité de la politique, mais les gains associés sont minimes. Finalement, il serait possible de considérer qu'à partir d'un certain seuil l'algorithme peut être arrêté sans attendre que les agents mettent fin, d'eux-mêmes, à la résolution.

Dans les résultats des figures 3.18 à 3.21, la décomposition en agents est plutôt intuitive (1 agent par point de transferts) et respecte la distribution « naturelle » du problème. Il est cependant possible d'adopter d'autres décompositions et en particulier de faire varier le nombre d'agents. Le tableau 3.2 présente les résultats de différentes compositions. Comme attendu, augmenter le nombre d'agents permet de réduire la taille de la fonction de transition et donc le temps de résolution. Globalement, la décomposition influe très peu sur la qualité de la politique puisqu'au final l'écart moyen au NNL varie entre 15 et 17 % sans détecter de tendance à une amélioration de la qualité avec un nombre d'agents

3.2. La coordination vue sous l'angle de la planification

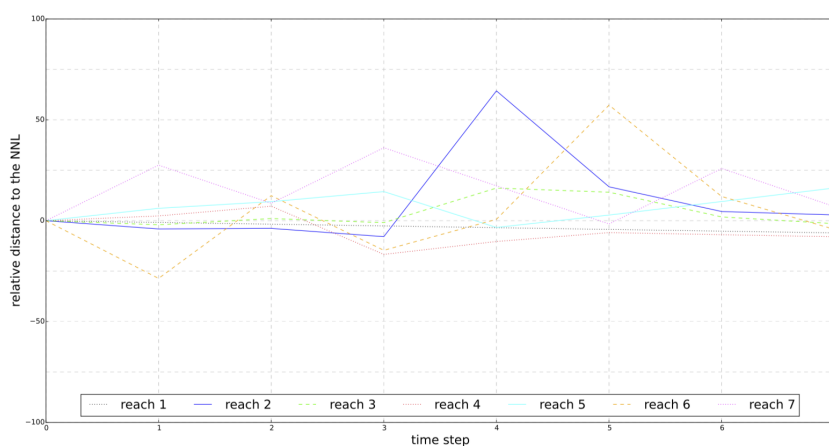


FIGURE 3.21 – Scénario 3 : navigation autorisée sur 24h et occurrence d'un évènement pluvieux intense anticipé

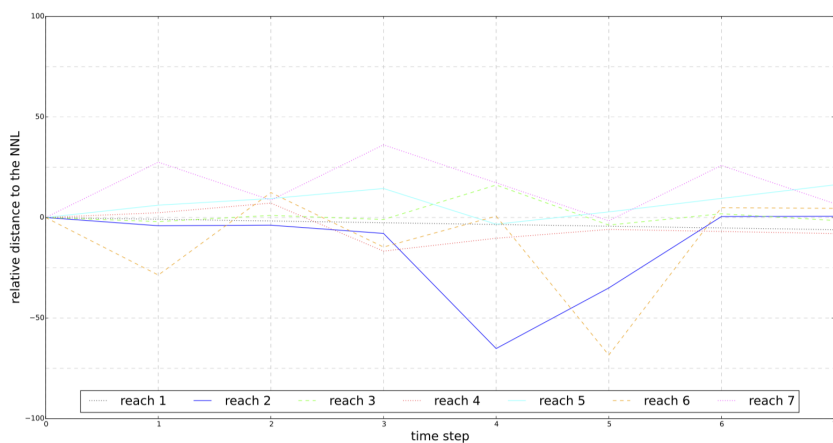


FIGURE 3.22 – Scénario 3 : navigation autorisée sur 24h avec un évènement pluvieux intense anticipé qui finalement ne se produit pas

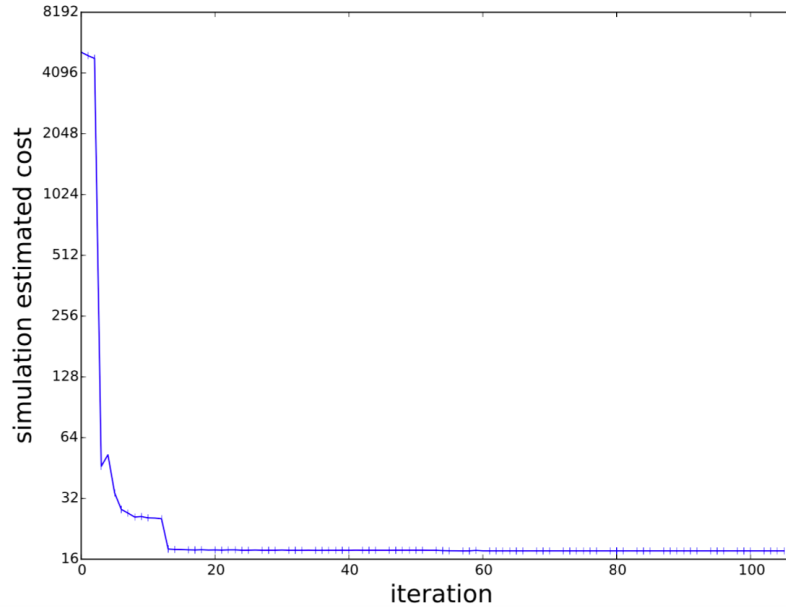


FIGURE 3.23 – Evolution du coût d'une politique jointe en fonction des itérations de l'algorithme

plus petit. Ce résultat est intéressant puisqu'intuitivement, on aurait pu imaginer observer un lien entre l'augmentation du nombre d'agents et une diminution de la qualité globale de la politique. En effet, lorsque le nombre d'agents augmente, la vision du réseau de chaque agent se réduit, augmentant ainsi sa dépendance aux politiques de son voisinage et le besoin de coordination. L'explication tient vraisemblablement au fait que les agents optimisent mieux localement lorsqu'ils sont nombreux et que cela compense l'optimisation globale entre agents qui est censée être moins bonne avec un nombre d'agents élevé.

nombre d'agents	nombre de transitions	temps (s)	écart moyen au NNL (en %)
6	5 799 936	591	17.13
7	2850816	167	15.38
8	475136	68	16.45
9	434560	35	17.27
10	353024	32	17.16
11	291968	30	17.83
12	230912	31	17.27
13	169856	30	16.64
14	108800	26	15.57

TABLEAU 3.2 – Variation du nombre d'agents pour la décomposition du réseau

Comme nous l'avons fait avec les approches de coordination par CSP, nous concluons cette section par une analyse des approches de coordination par planification.

3.2.3 Analyse des approches de coordination par planification

Pour cette analyse, nous reprenons les critères déjà utilisés en section 3.1.3 : la généralité, la complexité, l'adaptabilité aux environnements évolutifs, l'anticipation.

En comparaison avec les CSP, les MDP et leur variante offrent un peu moins de souplesse et d'expressivité. Prenons l'exemple du problème d'engagement d'unités de production : les contraintes des sources de production n'ont pas pu être exprimées de manière déclarative, mais ont du être intégrées au sein des fonctions de transition et de coûts. D'une manière générale, plus le problème dispose de contraintes à exprimer, plus la fonction de transition risque d'être compliquée et sa taille élevée.

Du point de vue de la complexité, nous avons pu voir sur les deux exemples étudiés : UCP et voies navigables, que le passage à l'échelle en nombre d'agents à coordonner nécessite de réduire le nombre d'états possibles par agent. La complexité est également dépendante des propriétés attendues sur la politique planifiée : séparabilité, prise en compte des incertitudes, etc.

L'adaptabilité de la coordination aux événements incertains dans la planification markovienne est conditionnée par leur prise en compte dès la construction du modèle. Sur le cas des voies navigables, nous avons par exemple montré que les crues étaient bien mieux anticipées lorsque les états correspondants font partie du modèle.

3.3 Conclusion

Dans ce chapitre, nous avons abordé la coordination d'agents en environnement évolutif suivant deux approches :

- une coordination au fil de l'eau utilisant un réseau de contraintes résolu à pas de temps régulier pour déterminer les actions à réaliser par les agents ;
- une planification pré-calculée permettant aux agents d'exploiter en ligne une politique jointe.

Ces travaux ont été essentiellement développés dans le cadre de la thèse de Guillaume Desquesnes [Desquesnes, 2018] et des projets AgentVPP et GEPETeau. Ils ont donné lieu à diverses publications dont principalement 4 journaux : [Duvilla *et al.*, 2018], [Dibangoye *et al.*, 2015], [Desquesnes *et al.*, 2016c], [Desquesnes *et al.*, 2017c], 10 conférences internationales : [Desquesnes *et al.*, 2017b], [Desquesnes *et al.*, 2016a], [Nouasse *et al.*, 2016], [Nouasse *et al.*, 2015], [Dibangoye *et al.*, 2013], [Dibangoye *et al.*, 2012a], [Monier *et al.*, 2010], [Monier *et al.*, 2011], [Doniec *et al.*, 2009], [Le *et al.*, 2009] et 2 conférences nationales : [Desquesnes *et al.*, 2017a], [Desquesnes *et al.*, 2016b].

Les approches de coordination présentées ont pu être testées sur des applications réelles impliquant jusqu'à une quinzaine d'agents. Au-delà, nous avons pu constater des temps de calcul non compatibles avec une exploitation en conditions réelles (de l'ordre de la journée pour les approches P-DPGO et OCLP, de l'ordre de la dizaine d'heures pour les CSP). Cette limite en nombre d'agents n'est pas liée aux protocoles de communications utilisés dans les différents algorithmes ; elle découle de la complexité inhérente des problèmes traités par les agents. Ainsi, dans les deux cas applicatifs étudiés (Smart Grid et réseaux de navigation), nous avons pu observer que pour passer à l'échelle en nombre d'agents, il faut compenser l'augmentation du nombre d'états liée à ces agents par une discrétisation moins fine des différentes variables du modèle. Il y a donc dans les deux cas un compromis à trouver entre la précision⁹ de la solution obtenue (plus la discrétisation est grossière, plus les solutions

9. Sans remettre en cause la qualité de la solution.

obtenues seront approximées) et le temps disponible pour le calcul d'une politique.

Au-delà des analyses respectives faites pour ces approches, nous pouvons ajouter que celles-ci disposent de propriétés intéressantes au regard des problématiques de confidentialité. En effet, dans les trois approches présentées (CSP distribués, P-DPGO et OCLP), chaque agent dispose d'un problème local qu'il peut résoudre sans être obligé de diffuser les informations propres à ce problème (contraintes dans le cas d'un CSP, états et fonction de transitions dans le cas d'un processus markovien) aux autres agents. Cette capacité à préserver la confidentialité des informations propres à chaque agent est intéressante pour des problèmes impliquant une coordination compétitive. Dans ce cas, les agents sont motivés par des intérêts propres et parfois en contradiction avec les intérêts des autres agents ; ils n'ont donc pas envie de mettre en commun des connaissances sur leur dynamique interne et leurs objectifs locaux. Une illustration parfaite de ce cas serait une centrale multi-sources qui viendrait agréger des unités de production de différents fournisseurs d'électricité. Les caractéristiques techniques des unités de production (éoliennes, groupes diesel) sont typiquement des informations que les fournisseurs d'électricité ne souhaiteraient pas partager avec leurs concurrents. Dans ce contexte compétitif, l'algorithme OCLP, par exemple, pourrait être utilisé sans aucune modification.

Chapitre 4

Conclusion et perspectives

Ce dernier chapitre fait la synthèse des contributions exposées aux chapitres 2 et 3 sur les problématiques de simulation et de conduite de systèmes évolutifs.

4.1 Synthèse des travaux de recherche

Les travaux présentés dans ce manuscrit s’inscrivent dans le domaine des systèmes multi-agents et ont été décrits suivant deux axes de recherche que sont : la conception d’agents dirigée par les données (chapitre 2) et la coordination d’agents en environnement dynamique et incertain (chapitre 3).

4.1.1 Contribution autour de la conception d’agents dirigée par les données

Le premier axe est motivé par les difficultés de mise en œuvre des simulations comportementales à base d’agents pour des phénomènes fortement dynamiques et évolutifs. Plusieurs étapes du processus de modélisation requièrent une expertise humaine : identification et caractérisation des entités à simuler, observation de leurs comportements et interaction. Ces tâches sont difficiles, en particulier lorsqu’il s’agit de simulations sociales et humaines. Dans les thèses d’Imen Saffar et d’Antoine Sylvain, nous avons exploré l’usage des techniques d’apprentissage non supervisé pour aider le concepteur de la simulation à construire ses modèles d’agents. En première approche, nous nous sommes restreints aux phénomènes impliquant des déplacements d’individus. Sous l’hypothèse de disposer de données d’observation des phénomènes à simuler, nous avons montré que des techniques de clustering permettaient de faire apparaître et de sélectionner des classes de comportements bien différenciées et statistiquement représentatives. Chacune des classes retenues peut ensuite être associée à une typologie d’agents qui viendra peupler la simulation.

En ce qui concerne la retranscription des comportements de déplacement propres à chaque classe au sein des agents, nous nous sommes fixés un modèle de déplacement reposant sur trois niveaux de décision (stratégique, tactique et opérationnel). Nous avons montré que pour des simulations simples, ce modèle pouvait être également alimenté par un apprentissage sur les données. Pour des simulations plus complexes, nous avons proposé l’usage d’un comportement supplétif permettant de représenter les motivations qui sous-tendent le modèle de déplacement. Au regard de l’état de l’art récent sur l’apprentissage et la reconnaissance d’activité, aucune approche ne peut être utilisée pour apprendre

ces motivations.

Les contributions scientifiques sur cet axe ont été appliquées dans le cadre de l'industrie du commerce et de la grande distribution avec l'objectif de construire une simulation d'un magasin de type supermarché. La méthodologie d'agentification dirigée par les données a pu être appliquée durant la thèse d'Antoine Sylvain sur des données réelles composées de trajectoires et de tickets de caisse. Ces trajectoires ont permis d'identifier via les méthodes de clustering des grandes catégories de consommateurs en fonction de leur comportement de déambulation. Ce modèle de déplacement ainsi appris a été complété par un comportement supplétif destiné à adapter les stratégies de déplacement apprises en fonction des motivations d'achats des clients.

Ce comportement supplétif se matérialise par un modèle d'intention d'achat permettant de générer, pour le modèle de déplacement, de nouveaux points d'intérêt. Ce modèle d'intention d'achat se base sur des travaux en psychologie et marketing qui ont identifié les facteurs prépondérants et déclencheurs de l'acte d'achat. D'un point de vue agent, ces facteurs nous ont permis de proposer un mécanisme d'attraction client-produit qui peut : déclencher l'acte d'achat, mais aussi influencer dynamiquement au cours de la simulation le déplacement propre à chaque typologie de client.

4.1.2 Contribution autour de la coordination d'agents en environnement dynamique et incertain

Le second axe s'inscrit dans une volonté d'aborder les problématiques de coordination multi-agent avec des approches génériques de l'IA classique. La mise en place d'un mécanisme de coordination requiert, entre autres, une modélisation des états propres à chaque agent et des dynamiques des agents. Deux approches sont alors bien adaptées pour ces modélisations : les CSP (Constraint Satisfaction Problem) et les MDP (Markov Decision Process).

Dans le cadre de mes travaux en DEA et en thèse, je m'étais intéressé à l'utilisation des CSP (classiques ou distribués) comme support à la coordination d'agents [Doniec *et al.*, 2008], [Doniec *et al.*, 2005]. J'ai poursuivi ensuite ces travaux en considérant des problèmes de coordination en environnement dynamique et incertain. Ainsi, dans le cadre du projet GEPETeau, nous avons proposé une coordination centralisée des biefs utilisant des résolutions successives de CSP. L'objectif est de piloter différents ouvrages (vannes, pompes, écluses) afin de pouvoir garantir un certain niveau d'eau à l'intérieur de chaque bief pour permettre la navigation. La capacité du mécanisme à réagir aux événements incertains (crues, sécheresse) provient de la mise à jour fréquente et de la résolution répétée du CSP.

Dans le cadre de la thèse de Tuan Van Le, des CSP distribués ont été utilisés pour l'exploration multi-robot sous contrainte de connectivité. Chaque robot de la flotte se déplace vers les zones non explorées de l'environnement et communique avec le reste de la flotte via un MANET (Mobile Ad Hoc Network). La coordination doit permettre aux robots de concilier deux buts antagonistes : s'éloigner suffisamment des uns des autres de manière à ne pas explorer une même zone tout en restant à distance suffisante pour maintenir la connectivité au sein du MANET. Ici encore, l'expressivité d'un modèle à base de contraintes permet de facilement modéliser les actions des robots et l'influence de leur dynamique de déplacement sur le MANET. Dans le cadre de la thèse de Pierre Monier, nous avons montré que ce problème d'exploration multi-robot pouvait servir de benchmark pour l'évaluation des performances des algorithmes de résolution de CSP distribués.

L'inconvénient majeur d'une coordination à base de contraintes réside dans la difficulté à anticiper

sur le long terme. Nous avons donc investigué des approches de coordination par planification dans le cadre du projet AgentVPP, puis de la thèse de Guillaume Desquesnes. Dans un contexte incertain, les approches markoviennes sont les plus adaptées, mais elles souffrent malheureusement de « la malédiction de la dimensionnalité » [Bellman, 1961] et peinent à passer à l'échelle du fait de la taille en mémoire du modèle (états et fonction de transition). La distribution de l'approche markovienne telle que présentée en section 3.2.2 permet de considérer la coordination d'un plus grand nombre d'agents tout en garantissant l'obtention d'une politique globale de bonne qualité.

4.2 Perspectives

Compte tenu des éléments présentés, des thèses et projets en cours à la date de rédaction de ce manuscrit, mon projet de recherche reste focalisé sur l'usage des systèmes multi-agents pour la simulation et le pilotage de systèmes évolutifs. Les perspectives exposées ci-dessous sont regroupées suivant les deux axes de recherche utilisés tout au long de ce document : la simulation multi-agent de systèmes évolutifs et la coordination multi-agent pour leur pilotage. La dernière partie de cette section présente une convergence possible entre ces deux axes.

4.2.1 La simulation multi-agent de systèmes évolutifs

4.2.1.1 Adaptation en ligne de simulations multi-agents

Les travaux présentés au chapitre 2 se positionnent comme une aide au concepteur de la simulation lui permettant de plus facilement « agentifier » les phénomènes à reproduire. Cette aide repose sur : l'utilisation de données passées et figées et un apprentissage hors-ligne non supervisé sur ces données. Les typologies d'agents ainsi apprises correspondent donc à une situation observée dans le passé. Si le phénomène à reproduire est continuellement en évolution (et pas seulement sur la période de temps pour laquelle les données ont été collectées), il est nécessaire de prévoir un mécanisme d'adaptation permettant de faire évoluer en continu la simulation.

Cela revient à proposer une approche intégrée permettant d'adapter en ligne la simulation sur la base de données collectée en temps réel. La mise en place d'une telle approche nécessite de pouvoir :

- confronter en temps réel les phénomènes observés (et les données collectées) avec les résultats de la simulation ;
- quantifier, à l'issue de cette confrontation, la dérive du (ou des) modèle(s) d'agents ;
- décider, sur la base de cette dérive, d'adapter ou non la simulation ;
- déterminer la nature de l'adaptation : apparition et/ou disparition de classes d'agents, variation de la répartition des classes d'agents en entrée de la simulation, adaptation des motivations et comportements propres à chaque classe d'agents, etc.

Les problématiques sous-jacentes sont :

- Comment caractériser la dérive de la simulation ?
- A quel niveau doit-on la quantifier (microscopique, macroscopique) ?
- Sur quels critères doit-on décider d'adapter les modèles d'agents en place dans la simulation ?
- Dans le contexte de phénomènes émergents, comment identifier les classes d'agents responsables de l'écart entre le réel et le simulé ?
- Comment apprendre de manière incrémentale des classes d'agents à partir de flots de données ?

Notons que certaines de ces problématiques sont abordées dans d'autres domaines tels que le diagnostic ou l'apprentissage dynamique. La réutilisation d'approches développées dans ces cadres peut donc s'envisager.

Dans le cadre du CPER ELSAT / ORIO¹, j'ai commencé à aborder la problématique d'adaptation de population d'agents dans un contexte de simulation de trafic routier. Cette adaptation concerne les caractéristiques propres à la conduite (vitesse désirée, respect du code) mais aussi les origines / destinations desquelles vont découler les flux de trafic au sein du réseau simulé. Pour adapter en ligne les origines / destinations des agents, il faut pouvoir être capable de mesurer simultanément sur le terrain les flux de trafic. Cette tâche est difficile et coûteuse, car elle oblige à équiper la voirie de dispositifs de comptages (boucles magnétiques, radars) si possible en grands nombres afin de garantir une bonne couverture du réseau simulé. Dans [Li *et al.*, 2019a, Li *et al.*, 2019b], nous avons montré qu'il était possible de remplacer ces dispositifs physiques de comptage par un modèle de flux appris à partir de données de temps parcours fournis par des services en ligne tels que Uber Movement (<https://movement.uber.com/?lang=fr-FR>) ou Google Maps (<https://developers.google.com/maps/documentation/?hl=fr>). Ces modèles de flux peuvent ensuite être utilisés durant une simulation de trafic pour adapter dynamiquement la demande de trafic (i.e. la population d'agents à générer) sur le réseau étudié. Dans le prolongement de ces travaux, une thèse est actuellement en cours² et s'intéresse à l'adaptation de population d'agents pour la simulation de la multimodalité à l'échelle d'un territoire.

4.2.1.2 Simulation comportementale à large échelle

Dans le cadre de la thèse d'Antoine Sylvain, nous avons soulevé des problèmes de passage à l'échelle lors de la simulation de grands centres commerciaux fréquentés. Il s'agit d'un problème assez courant en simulation multi-agent : plus le modèle d'agent et les comportements associés sont riches et détaillés, plus ils sont coûteux à exécuter en termes de temps de calcul et plus il est difficile de simuler un grand nombre d'agents. Ces dernières années, certaines plateformes multi-agents ont été déclinées en des versions exécutables sur des clusters de calculs haute performance (par exemple la plateforme Repast HPC³). Le problème est actuel comme le montrent de récents travaux [Bhattacharya *et al.*, 2019, Park *et al.*, 2018, Moreno *et al.*, 2019, Wijerathne *et al.*, 2018] ou encore le programme de recherche de la DARPA : SocialSim⁴ qui vise à développer des simulations multi-agents massives de comportements sociaux en lignes [Blythe *et al.*, 2019].

Une manière d'aborder ce problème (sans étendre les capacités de calculs disponibles pour la simulation) consiste à créer des simulations multi-niveaux [Espíe *et al.*, 2006, Gaud *et al.*, 2008] (ou multi-échelles) permettant, selon le besoin, d'ajuster le niveau de détails de la simulation et par conséquent sa rapidité d'exécution. Deux niveaux en particulier peuvent être considérés : un niveau microscopique faisant appel à des modèles de comportements « individus-centrés » et relativement détaillés, et un niveau macroscopique considérant plutôt des flux d'individus. L'objectif est de pouvoir jongler avec ses deux niveaux de détails de manière dynamique en cours de simulation.

D'un point de vue pratique, cela revient à « immerger un petit ensemble d'agents » avec un com-

1. ORIO (Observation des peRformances des Infrastructures urbaines et de la mObilité) est un projet du CPER ELSAT 2020 (Ecomobilité, Logistique, Sécurité et Adaptabilité dans les Transports)

2. Azise Diallo (2018-2021) - Simulation multi-agent pour la mobilité du futur

3. https://repast.github.io/repast_hpc.html

4. <https://www.darpa.mil/program/computational-simulation-of-online-social-behavior>

portement microscopique riche et détaillé au sein d'une simulation macroscopique. Ce petit ensemble d'agents peut alors être vu comme une « bulle » microscopique permettant à l'utilisateur de la simulation d'observer finement les phénomènes locaux qui s'y produisent. Tous les phénomènes en dehors de cette bulle (et donc non visualisé par l'utilisateur) peuvent être considérés à un niveau macroscopique.

La principale difficulté d'une telle approche réside dans le passage du microscopique au macroscopique et inversement. Il faut en effet garantir que les phénomènes locaux au sein de la bulle sont consistants avec le comportement macroscopique de la simulation à l'extérieur de la bulle. En particulier, il faut s'assurer que les agents présents dans la bulle sont bien représentatifs du phénomène simulé dans sa globalité au niveau macroscopique. Dans le cas d'une simulation d'agents situés, il faut par exemple gérer les créations et la destruction des agents qui entrent ou sortent de la bulle. On retombe ici sur la problématique de constitution d'une population d'agents réaliste au regard du phénomène à simuler. Dans ce cadre, les approches par apprentissage présentées en section 2.2.2.2 peuvent certainement être adaptées pour faciliter la génération des agents (notamment en termes de typologie et de caractéristiques) à la frontière de la bulle microscopique.

4.2.2 Coordination multi-agent pour le pilotage de systèmes évolutifs

4.2.2.1 Identification des classes de problèmes pouvant bénéficier de l'hybridation entre processus markoviens et problèmes de satisfaction de contraintes

A notre connaissance, les premières idées visant à introduire la notion d'incertitude dans la programmation par contraintes remontent aux débuts des années 2000 avec la proposition par T. Walsh des CSP stochastiques [Walsh, 2002] abordant les problèmes de décision sous incertitude via la programmation par contraintes. Dans la même lignée, Fowler et Brown ont proposé en 2003 une autre classe de CSP : les BCSPs dédiés à l'allocation dynamique de ressources avec incertitude [Fowler et Brown, 2003]. L'article [Verfaillie et Jussien, 2005] donne, sur cette période, une vision d'ensemble des différents travaux visant à combiner raisonnement incertain et CSP.

Depuis, d'autres travaux ont proposé des hybridations entre des modèles markoviens et la programmation par contraintes. Dans [Besse et Chaib-Draa, 2007], les auteurs partent des CSP dynamiques (succession de CSP classiques, chacun résultant du précédent par l'ajout ou la suppression d'une contrainte [Dechter et Dechter, 1988]) et en décrivent l'évolution (le passage d'un CSP à un autre) par un processus de décision markovien. La classe résultante appelée MaCSP (markovian CSP) est appliquée à la résolution de problème d'allocation de ressource contrainte. Plus récemment, les auteurs de [Savaux *et al.*, 2019] ont proposé une extension des CSP distribués appelés (UDisCSP - *Utilitarian Distributed Constraint Satisfaction Problem*) prenant en compte la notion de privacité au travers l'expression de coûts associés à la révélation d'informations personnelles. Les auteurs montrent une équivalence entre les UDisCSP et les jeux stochastiques partiellement observables qui sont une extension des POMDP.

Dans l'algorithme OCLP présenté au chapitre précédent, nous nous sommes inspirés d'une approche utilisée en résolution de CSP distribués pour établir le protocole de communication entre agents. L'hybridation entre planification markovienne et programmation par contraintes peut également se concevoir en termes de pré-calculs permettant de simplifier l'espace d'états d'un MDP. Par exemple, dans [Dibangoye *et al.*, 2012b] nous avons utilisé des COP pour éviter l'énumération exhaustive des règles de décision accessibles à chaque état d'un dec-MDP.

D'une manière générale, l'hybridation entre planification markovienne et programmation par contraintes peut donc avoir plusieurs finalités différentes. Comme le montrent les résultats dans [Di-bangoye *et al.*, 2012b], elle peut clairement permettre d'augmenter la taille des instances de certains problèmes. Il nous semble donc intéressant d'identifier et d'étudier les différentes classes de problèmes pour lesquelles les COP ou les CSP permettraient de réduire l'espace mémoire et/ou la complexité calculatoire.

4.2.2.2 Découplage des raisonnements internes aux agents et des mécanismes de coordination

Les algorithmes de résolution de CSP distribués utilisés en section 3.1.2 considèrent deux types de contraintes : des contraintes intra-agents appliquées aux variables propres d'un agent et des contraintes inter-agents définies entre des variables de différents agents. Les algorithmes de résolution de CSP distribués ne tiennent compte que des contraintes inter-agents pour l'échange des messages entre les agents ; les contraintes intra-agents peuvent être gérées par n'importe quel algorithme classique de filtrage et de recherche exhaustive.

Cette indépendance possible entre le raisonnement interne des agents et le mécanisme de coordination pourrait également s'obtenir dans l'algorithme OCLP proposé dans la thèse de G. Desquesnes. Dans les expérimentations menées, chaque problème local résolu par un agent était modélisé sous la forme d'un MDP. Cependant, moyennant la mise en place d'un format d'échange standard de la politique locale à chaque agent, n'importe quel outil de planification pourrait être utilisé en local par les agents.

Cette propriété de découplage entre mécanismes de décision individuelle et collective est intéressante puisqu'elle facilite l'interopérabilité tout en préservant une certaine forme de confidentialité : lors des échanges entre les agents, ces derniers ne sont amenés à ne communiquer que sur les actions qu'ils ont localement planifiées sans dévoiler leurs variables d'état interne.

En outre, cette propriété permet de plus facilement aborder les problèmes nécessitant une coordination compétitive dans laquelle les agents sont motivés par des buts individuels parfois antagonistes, mais cherchent tout de même à se coordonner de manière à garantir un certain fonctionnement global.

4.2.2.3 Coordination en environnement évolutif et web sémantique

Dans la définition proposée en section 1.1.2.2 (page 9) et dans les différents travaux exposés par la suite, nous avons considéré que les différents événements qui pouvaient survenir et affecter l'environnement seraient automatiquement et instantanément connus des agents. La détection de ces événements et la diffusion de cette connaissance à l'ensemble des agents est une problématique en soi. Elle a notamment été abordée dans le cadre de l'intelligence ambiante [Cook *et al.*, 2009, Preuveneers *et al.*, 2004] et des bâtiments intelligents [Robles et Kim, 2010, Kumar *et al.*, 2013]. Pour ces applications, les technologies utilisées se basent généralement sur le web sémantique, les ontologies et le traitement d'événements complexes [Stavropoulos *et al.*, 2012, Chen *et al.*, 2014, Kucera *et al.*, 2014].

Une perspective intéressante est donc l'étude de l'impact de ces technologies sur les mécanismes de coordination mise en œuvre dans un environnement évolutif. Cette piste fait actuellement l'objet d'une thèse⁵ en collaboration avec l'université Rey Juan Carlos de Madrid. L'application considérée

5. Qasim Khalid (2017-2020) -Multi agent system Coordination and Cooperation applied to Smart Building

est le guidage de personnes lors de l'évacuation d'un bâtiment intelligent [Khalid *et al.*, 2019b]. La thèse vise à utiliser une représentation sémantique du bâtiment et de ces occupants pour le développement d'une architecture distribuée de recommandation de chemins en tenant compte du contexte et des évènements incertains qui peuvent se produire durant l'évacuation d'un bâtiment (propagation d'un incendie, perte d'accès aux escaliers de secours, obstruction d'un couloir, etc). La recommandation des chemins aux différentes personnes doit tenir compte des possibles congestions et nécessite donc la mise en place de mécanismes de coordination permettant une répartition optimale des personnes à évacuer sur les différentes sorties de secours.

Les mécanismes de coordination que nous cherchons à développer doivent pouvoir prendre en compte les caractéristiques des personnes notamment en termes de déplacement (personnes âgées ou à mobilité réduite) et liens sociaux (familles). En première approche, nous formons des groupes partageant des caractéristiques communes (ou complémentaires ; cas de personnes à mobilité réduite devant être accompagnées) [Khalid *et al.*, 2019a] qui se voient ensuite affectée à un itinéraire d'évacuation tenant compte des déplacements des autres groupes.

4.2.3 Vers une approche intégrée mêlant coordination et simulation multi-agent

Les deux thèmes de recherche exposés dans ce manuscrit étaient déjà présents dans mes travaux de thèse : la simulation comportementale de conducteur en carrefour y était abordée comme un problème de coordination multi-agent. Les travaux exposés dans ce manuscrit sur ces deux thèmes sont en revanche complètement décorrélés. A terme, ils pourraient à nouveau converger en considérant des problèmes nécessitant l'usage de simulations dans le processus de coordination d'agents.

Pour illustrer cette idée, considérons à nouveau la problématique de guidage dans les espaces intelligents. Dans l'article [Lujak et Doniec, 2018], nous avons présenté une approche multi-agent pour le guidage de clients en magasin. Chaque client est représenté par un agent déployé sur le smartphone de celui-ci. Cet agent « client » communique avec des agents « infrastructure » qui en temps réel scrutent le trafic dans les rayons et détectent les congestions. Sur la base de ces informations et d'une liste de courses renseignée par le client, chaque agent « client » calcule (ou met à jour) une trajectoire optimale permettant au client d'acheter les produits de sa liste tout en limitant les ralentissements dus aux congestions. La trajectoire ainsi calculée est remontée aux agents « infrastructure » pour leur permettre d'affiner leur vision de l'occupation du magasin.

Ces informations pourraient être encore plus affinées en utilisant une simulation de magasin telle que présentée en section 2.3.6. L'outil pourrait en effet fournir aux agents « infrastructure » des prédictions plus précises sur les flux de clients et l'occupation des rayons. En retour, les données de déplacement collectées par les smartphones pourraient venir enrichir en temps réel la simulation. Dans cette approche intégrée, la coordination des agents « client » se baserait sur les résultats fournis par la simulation comportementale et celle-ci pourrait bénéficier de nouvelles informations remontant des agents « clients » pour adapter en ligne les modèles de comportement utilisés.

Bibliographie

- [Abar *et al.*, 2017] ABAR, S., THEODOROPOULOS, G. K., LEMARINIER, P. et O’HARE, G. M. (2017). Agent based modelling and simulation tools : A review of the state-of-art software. *Computer Science Review*, 24:13 – 33.
- [Abolhasan *et al.*, 2004] ABOLHASAN, M., WYSOCKI, T. et DUTKIEWICZ, E. (2004). A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1):1 – 22.
- [Adu-Kankam et Camarinha-Matos, 2018] ADU-KANKAM, K. O. et CAMARINHA-MATOS, L. M. (2018). Towards collaborative virtual power plants : Trends and convergence. *Sustainable Energy, Grids and Networks*, 16:217 – 230.
- [Ahuja *et al.*, 1990] AHUJA, R. K., MEHLHORN, K., ORLIN, J. et TARJAN, R. E. (1990). Faster algorithms for the shortest path problem. *Journal of ACM*, 37(2).
- [Ajib *et al.*, 2018] AJIB, B., LEFTERIU, S., CAUCHETEUX, A., LECOEUICHE, S. et GAUVRIT, J. (2018). Prediction of standardized energy consumption of existing buildings based on hybrid systems modeling and control. In *57th IEEE Conference on Decision and Control, CDC 2018, Miami, FL, USA, December 17-19, 2018*, pages 3880–3886.
- [Alami *et al.*, 1998] ALAMI, R., INGRAND, F. F. et QUTUB, S. (1998). A scheme for coordinating multi-robots planning activities and plans execution. In *ECAI*, pages 617–621.
- [Albanese *et al.*, 2008] ALBANESE, M., CHELLAPPA, R., MOSCATO, V., PICARIELLO, A., SUBRAHMANIAN, V. S., TURAGA, P. et UDREA, O. (2008). A constrained probabilistic petri net framework for human activity detection in video. *IEEE Transactions on Multimedia*, 10(6):982–996.
- [Amblard, 2010] AMBLARD, F. (2010). Construire des sociétés artificielles pour comprendre les phénomènes sociaux réels. *Nouvelles perspectives en sciences sociales : Revue internationale de systémique complexe et d’études relationnelles*, 5(2):69–77.
- [Amblard et Phan, 2006] AMBLARD, F. et PHAN, D. (2006). *Modélisation et simulation multi-agents*. science informatique et SHS. Hermes Lavoisier.
- [Aylott et Mitchell, 1998] AYLOTT, R. et MITCHELL, V.-W. (1998). An exploratory study of grocery shopping stressors. *International Journal of Retail & Distribution Management*, 26(9):362–373.
- [Baqueiro *et al.*, 2009] BAQUEIRO, O., WANG, Y. J., MCBURNEY, P. et COENEN, F. (2009). Integrating data mining and agent based modeling and simulation. In *Advances in Data Mining. Applications and Theoretical Aspects*, pages 220–231, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Bar-Yam, 1997] BAR-YAM, Y. (1997). *Dynamics of Complex Systems*. Perseus Books.

BIBLIOGRAPHIE

- [Barbosa et Leitão, 2011] BARBOSA, J. et LEITÃO, P. (2011). Simulation of multi-agent manufacturing systems using agent-based modelling platforms. *In 2011 9th IEEE International Conference on Industrial Informatics*, pages 477–482. IEEE.
- [Bauer et al., 2001] BAUER, B., MALLER, J. P. et ODELL, J. (2001). Agent uml : A formalism for specifying multiagent software systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(03):207–230.
- [Beauchemin et Barron, 1995] BEAUCHEMIN, S. S. et BARRON, J. L. (1995). The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–466.
- [Bellifemine et al., 2007] BELLIFEMINE, F. L., CAIRE, G. et GREENWOOD, D. (2007). *Developing Multi-Agent Systems with JADE*. Wiley.
- [Bellman, 1961] BELLMAN, R. (1961). *Adaptive Control Processes*. Princeton University Press.
- [Bera et al., 2015] BERA, A., KIM, S. et MANOCHA, D. (2015). Efficient trajectory extraction and parameter learning for data-driven crowd simulation. *In Proceedings of the 41st Graphics Interface Conference*, pages 65–72. Canadian Information Processing Society.
- [Bernon et al., 2003] BERNON, C., GLEIZES, M.-P., PEYRUQUEOU, S. et PICARD, G. (2003). Adelfe : A methodology for adaptive multi-agent systems engineering. *In Proceedings of the 3rd International Conference on Engineering Societies in the Agents World III, ESAW'02*, pages 156–169, Berlin, Heidelberg. Springer-Verlag.
- [Besse et Chaib-Draa, 2007] BESSE, C. et CHAIB-DRAA, B. (2007). A markovian model for dynamic and constrained resource allocation problems. *In Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1846.
- [Beuthe et al., 2012] BEUTHE, M., JOURQUIN, B., URBAIN, N., BRUINSMA, F., LINGEMANN, I., UBBELS, B. et HEUMEN, E. V. (2012). Estimating the impacts of water depth and new infrastructures on transport by inland navigation : A multimodal approach for the rhine corridor. volume 54, pages 387 – 401.
- [Bhattacharya et al., 2019] BHATTACHARYA, P., EKANAYAKE, S., KUHLMAN, C. J., LEBIERE, C., MORRISON, D., SWARUP, S., WILSON, M. L. et ORR, M. G. (2019). The matrix : An agent-based modeling framework for data intensive simulations. *In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1635–1643. International Foundation for Autonomous Agents and Multiagent Systems.
- [Blanc-Rouchossé et al., 2018] BLANC-ROUCHOSSÉ, J.-B., BLAVETTE, A., CAMILLERI, G. et GLEIZES, M.-P. (2018). Electric Vehicles Fleet for Frequency Regulation Using a Multi-Agent System (regular paper). *In International Conference on Practical Applications of Agents and Multiagent Systems (PAAMS 2018), Toledo, Spain*, numéro 10978, pages 84–96.
- [Blythe et al., 2019] BLYTHE, J., FERRARA, E., HUANG, D., LERMAN, K., MURIC, G., SAPIENZA, A., TREGUBOV, A., PACHECO, D., BOLLENBACHER, J., FLAMMINI, A. et al. (2019). The darpa socialsim challenge : Massive multi-agent simulations of the github ecosystem. *In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1835–1837.
- [Bobick et Davis, 2001] BOBICK, A. F. et DAVIS, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3): 257–267.

- [Boser *et al.*, 1992] BOSER, B. E., GUYON, I. M. et VAPNIK, V. N. (1992). A training algorithm for optimal margin classifiers. *In Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.
- [Boucher *et al.*, 2009] BOUCHER, A., CANAL, R., CHU THANH, Q., DROGOUL, A., GAUDOU, B., VAN TUAN, L., VICTOR, M., NGUYEN, N. V., NGUYEN VU, Q. A., TAILLANDIER, P., FRANÇOIS, S. et STINCKWICH, S. (2009). The AROUND project : Adapting robotic disaster response to developing countries. *In Safety, Security & Rescue Robotics (SSRR), 2009 IEEE International Workshop*.
- [Boutilier, 1999] BOUTILIER, C. (1999). Sequential optimality and coordination in multiagent systems. *In Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'99*, pages 478–485.
- [Boutilier *et al.*, 1995] BOUTILIER, C., DEARDEN, R. et GOLDSZMIDT, M. (1995). Exploiting structure in policy construction. *In Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95*, pages 1104–1111.
- [Boyd et Vandenberghe, 2004] BOYD, S. et VANDENBERGHE, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.
- [Boé *et al.*, 2009] BOÉ, J., TERRAY, L., MARTIN, E. et HABETS, F. (2009). Projected changes in components of the hydrological cycle in french river basins during the 21st century. *Water Resources Research*, 45(8).
- [Calo *et al.*, 2017] CALO, S. B., VERMA, D. C. et BERTINO, E. (2017). Distributed intelligence : Trends in the management of complex systems. *In Proceedings of the 22Nd ACM on Symposium on Access Control Models and Technologies, SACMAT '17 Abstracts*, pages 1–7, New York, NY, USA. ACM.
- [Cao, 2009] CAO, L. (2009). *Data Mining and Multi-agent Integration*. Springer Publishing Company, Incorporated.
- [Cao *et al.*, 2012] CAO, L., WEISS, G. et YU, P. S. (2012). A brief introduction to agent mining. *Autonomous Agents and Multi-Agent Systems*, 25(3):419–424.
- [Casals *et al.*, 2018] CASALS, A., EL FALLAH-SEGHRUCHNI, A. et BRANDÃO, A. A. F. (2018). Augmented agents : Contextual perception and planning for bdi architectures. *In EL FALLAH-SEGHRUCHNI, A., RICCI, A. et SON, T. C., éditeurs : Engineering Multi-Agent Systems*, pages 38–55. Springer International Publishing.
- [Castel *et al.*, 1996] CASTEL, C., CHAUDRON, L. et TESSIER, C. (1996). What is going on ? a high level interpretation of sequences of images. *Proc. ECCV Workshop on Conceptual Descriptions from Images*, pages 13–27.
- [Cedras et Shah, 1995] CEDRAS, C. et SHAH, M. (1995). Motion-based recognition a survey. *Image and Vision Computing*, 13(2):129 – 155.
- [Cervenka et Trencansky, 2007] CERVENKA, R. et TRENCANSKY, I. (2007). *The Agent Modeling Language - AML : A Comprehensive Approach to Modeling Multi-Agent Systems (Whitestein Series in Software Agent Technologies and Autonomic Computing)*. Birkhauser Basel.
- [Chen *et al.*, 2014] CHEN, C. Y., FU, J. H., SUNG, T., WANG, P.-F., JOU, E. et FENG, M.-W. (2014). Complex event processing for the internet of things and its applications. *In 2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1144–1149. IEEE.

BIBLIOGRAPHIE

- [Clausen *et al.*, 2017] CLAUSEN, A., UMAIR, A., DEMAZEAU, Y. et JØRGENSEN, B. N. (2017). Agent-based integration of complex and heterogeneous distributed energy resources in virtual power plants. In DEMAZEAU, Y., DAVIDSSON, P., BAJO, J. et VALE, Z., éditeurs : *Advances in Practical Applications of Cyber-Physical Multi-Agent Systems : The PAAMS Collection*, pages 43–55.
- [Cook *et al.*, 2009] COOK, D. J., AUGUSTO, J. C. et JAKKULA, V. R. (2009). Ambient intelligence : Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298.
- [Cook et Krishnan, 2015] COOK, D. J. et KRISHNAN, N. C. (2015). *Activity Learning : Discovering, Recognizing, and Predicting Human Behavior from Sensor Data*. Wiley.
- [Cossentino, 2005] COSSENTINO, M. (2005). *From Requirements to Code with the PASSI Methodology*. Idea Group Inc., Hershey, PA, USA.
- [Cossentino *et al.*, 2008] COSSENTINO, M., FORTINO, G., GARRO, A., MASCILLARO, S. et RUSSO, W. (2008). Passim : a simulation-based process for the development of multi-agent systems. *Int. J. Agent-Oriented Softw. Eng.*, 2(2):132–170.
- [Cossentino *et al.*, 2010] COSSENTINO, M., GAUD, N., HILAIRE, V., GALLAND, S. et KOUKAM, A. (2010). Aspecs : an agent-oriented software process for engineering complex systems. *Autonomous Agents and Multi-Agent Systems*, 20(2):260–304.
- [Cumenal, 2017] CUMENAL, D. (2017). Dynamique des systèmes complexes concepts et méthodologie. *Techniques de l'ingénieur Théorie et management des systèmes complexes*.
- [Davidson *et al.*, 2008] DAVIDSON, E. M., MCARTHUR, S. D. J., YUEN, C. et LARSSON, M. (2008). Aura-nms : Towards the delivery of smarter distribution networks through the application of multi-agent systems technology. In *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–6.
- [de Hoog *et al.*, 2010] de HOOG, J., CAMERON, S. et VISSER, A. (2010). Selection of rendezvous points for multi-robot exploration in dynamic environments. In *Workshop on Agents in Realtime and Dynamic Environments, International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
- [De La Higuera, 2000] DE LA HIGUERA, C. (2000). Current trends in grammatical inference. In *Advances in Pattern Recognition*, pages 28–31. Springer.
- [Dechter et Dechter, 1988] DECHTER, R. et DECHTER, A. (1988). *Belief maintenance in dynamic constraint networks*. University of California, Computer Science Department.
- [Demazeau, 1995] DEMAZEAU, Y. (1995). From interactions to collective behaviour in agent-based systems. In *Proceedings of the 1st European Conference on Cognitive Science*, pages 117–132, St Malo.
- [Dempster *et al.*, 1977] DEMPSTER, A. P., LAIRD, N. M. et RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.
- [Desquesnes, 2018] DESQUESNES, G. (2018). *Distribution de Processus Décisionnels Markoviens pour une gestion prédictive d'une ressource partagée : application aux voies navigables des Hauts-de-France dans le contexte incertain du changement climatique*. Thèse de doctorat, Ecole nationale supérieure Mines-Télécom Lille Douai.
- [Desquesnes *et al.*, 2016a] DESQUESNES, G., LOZENGUEZ, G., DONIEC, A. et DUVELLA, E. (2016a). Dealing with large mdps, case study of waterway networks supervision. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 48–59. Springer.

- [Desquesnes *et al.*, 2016b] DESQUESNES, G., LOZENGUEZ, G., DONIEC, A. et DUVELLA, É. (2016b). Mdp à grande échelle : étude de cas des voies navigables. *In Reconnaissance des formes et intelligence artificielle (RFIA 16), Clermont-Ferrand*.
- [Desquesnes *et al.*, 2016c] DESQUESNES, G., LOZENGUEZ, G., DONIEC, A. et DUVELLA, É. (2016c). Planning large systems with mdps : case study of inland waterways supervision. *AD-CAIJ : Advances in Distributed Computing and Artificial Intelligence Journal*, 5(4):71–84.
- [Desquesnes *et al.*, 2017a] DESQUESNES, G., LOZENGUEZ, G., DONIEC, A. et DUVELLA, E. (2017a). Coordination distribuée et hors-ligne de planifications locales. *In Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite des systèmes (JFPDA 17), Caen*.
- [Desquesnes *et al.*, 2017b] DESQUESNES, G., LOZENGUEZ, G., DONIEC, A. et DUVELLA, E. (2017b). Distributed mdp for water resources planning and management in inland waterways. *In 20th World Congress of the International Federation of Automatic Control, Toulouse*, volume 50, pages 6576–6581. Elsevier.
- [Desquesnes *et al.*, 2017c] DESQUESNES, G., LOZENGUEZ, G., DONIEC, A. et DUVELLA, E. (2017c). Vers une distribution des mdp à grande échelle : étude de cas des voies navigables. *Revue d'Intelligence Artificielle*, 31(1-2):183.
- [Dibangoye *et al.*, 2012a] DIBANGOYE, J. S., AMATO, C. et DONIEC, A. (2012a). Scaling up decentralized mdps through heuristic search. *In Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 217–226. AUAI Press.
- [Dibangoye *et al.*, 2012b] DIBANGOYE, J. S., AMATO, C. et DONIEC, A. (2012b). Scaling up decentralized mdps through heuristic search. *In Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pages 217–226.
- [Dibangoye *et al.*, 2013] DIBANGOYE, J. S., AMATO, C., DONIEC, A. et CHARPILLET, F. (2013). Producing efficient error-bounded solutions for transition independent decentralized mdps. *In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 539–546. International Foundation for Autonomous Agents and Multiagent Systems.
- [Dibangoye *et al.*, 2015] DIBANGOYE, J. S., DONIEC, A., FAKHAM, H., COLAS, F. et GUILLAUD, X. (2015). Distributed economic dispatch of embedded generation in smart grids. *Engineering Applications of Artificial Intelligence*, 44:64–78.
- [Dichtl *et al.*, 2018] DICHTL, J., FABRESSE, L., LOZENGUEZ, G. et BOURAQADI, N. (2018). Poly-map : A 2d polygon-based map format for multi-robot autonomous indoor localization and mapping. *In Intelligent Robotics and Applications - 11th International Conference, ICIRA 2018, Newcastle, NSW, Australia, August 9-11, 2018, Proceedings, Part I*, pages 120–131.
- [Dimeas et Hatziargyriou, 2005] DIMEAS, A. L. et HATZIARGYRIOU, N. D. (2005). Operation of a multiagent system for microgrid control. *IEEE Transactions on Power Systems*, 20(3):1447–1455.
- [Dimopoulos et Moraitis, 2006] DIMOPOULOS, Y. et MORAITIS, P. (2006). Multi-agent coordination and cooperation through classical planning. *In Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, pages 398–402. IEEE Computer Society.
- [Djanatliev et German, 2013] DJANATLIEV, A. et GERMAN, R. (2013). Large scale healthcare modeling by hybrid simulation techniques using anylogic. *In Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques, SimuTools '13*, pages 248–257.

BIBLIOGRAPHIE

- [Domingos et Pazzani, 1997] DOMINGOS, P. et PAZZANI, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130.
- [Doniec *et al.*, 2009] DONIEC, A., BOURAQADI, N., DEFOORT, M., LE, V. T. et STINCKWICH, S. (2009). Distributed constraint reasoning applied to multi-robot exploration. In *ICTAI 2009, 21st IEEE International Conference on Tools with Artificial Intelligence, Newark, New Jersey, USA, 2-4 November 2009*, pages 159–166.
- [Doniec *et al.*, 2019] DONIEC, A., LECOEUICHE, S., MANDIAU, R. et SYLVAIN, A. (2019). Purchase intention based agent for customer behaviors. *Information Sciences, Seconde révision en cours*.
- [Doniec *et al.*, 2008] DONIEC, A., MANDIAU, R., PIECHOWIAK, S. et ESPIÉ, S. (2008). Anticipation based on constraint processing in a multi-agent context. *Autonomous Agents and Multi-Agent Systems*, 17(2):339–361.
- [Doniec *et al.*, 2005] DONIEC, A., PIECHOWIAK, S., MANDIAU, R. *et al.* (2005). A discsp solving algorithm based on sessions. In *FLAIRS Conference*, pages 666–670.
- [Dufour *et al.*, 2005] DUFOUR, C., ABOURIDA, S. et BELANGER, J. (2005). Hardware-in-the-loop simulation of power drives with rt-lab. In *2005 International Conference on Power Electronics and Drives Systems*, volume 2, pages 1646–1651.
- [Duviella *et al.*, 2018] DUVIELLA, E., DONIEC, A. et NOUASSE, H. (2018). Adaptive water-resource allocation planning of inland waterways in the context of global change. *Journal of Water Resources Planning and Management*, 144(9).
- [Dziczkowski *et al.*, 2011] DZICZKOWSKI, G., DONIEC, A. et LECOEUICHE, S. (2011). Triggering rules for conversational agents in trading situations. In *ICAART (1)*, pages 382–387.
- [El Fallah-Seghrouchni *et al.*, 2004] EL FALLAH-SEGHROUCHNI, A., DEGIRMENCIYAN-CARTAULT, I. et MARC, F. (2004). Modelling, control and validation of multi-agent plans in dynamic context. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 44–51. IEEE Computer Society.
- [Espíe *et al.*, 2006] ESPIÉ, S., GATTUSO, D. et GALANTE, F. (2006). Hybrid traffic model coupling macro- and behavioral microsimulation. In *Transportation Research Board 85th Annual Meeting*.
- [Feron et Monti, 2017] FERON, B. et MONTI, A. (2017). An agent based approach for virtual power plant valuing thermal flexibility in energy markets. In *2017 IEEE Manchester PowerTech*, pages 1–6.
- [Fikes et Nilsson, 1971] FIKES, R. E. et NILSSON, N. J. (1971). STRIPS : A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208.
- [Fowler et Brown, 2003] FOWLER, D. W. et BROWN, K. N. (2003). Branching constraint satisfaction problems and markov decision problems compared. *Annals of Operations Research*, 118(1-4):85–100.
- [Fox *et al.*, 2000] FOX, D., BURGARD, W., KRUPPA, H. et THRUN, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344.
- [Galán *et al.*, 2009] GALÁN, J. M., IZQUIERDO, L. R., IZQUIERDO, S. S., SANTOS, J. I., DEL OLMO, R., LÓPEZ-PAREDES, A. et EDMONDS, B. (2009). Errors and artefacts in agent-based modelling. *Journal of Artificial Societies and Social Simulation*, 12(1):1.
- [García *et al.*, 1989] GARCIA, C. E., PRETT, D. M. et MORARI, M. (1989). Model predictive control : theory and practice - a survey. *Automatica*, 25(3):335–348.

- [Garcia-Ojeda *et al.*, 2008] GARCIA-OJEDA, J. C., DELOACH, S. A., ROBBY, R., OYENAN, W. H. et VALENZUELA, J. (2008). O-mase : A customizable approach to developing multiagent development processes. In *Proceedings of the 8th International Conference on Agent-oriented Software Engineering VIII, AOSE'07*, pages 1–15, Berlin, Heidelberg. Springer-Verlag.
- [Gaud *et al.*, 2008] GAUD, N., GALLAND, S., GECHTER, F., HILAIRE, V. et KOUKAM, A. (2008). Holonic multilevel simulation of complex systems : Application to real-time pedestrians simulation in virtual urban environment. *Simulation Modelling Practice and Theory*, 16(10):1659–1676.
- [Georgeff, 1983] GEORGEFF, M. (1983). Communication and interaction in multi-agent planning. In *Proceedings of the Third AAAI Conference on Artificial Intelligence, AAAI'83*, pages 125–129.
- [Ghallab *et al.*, 1998] GHALLAB, M., HOWE, A., KNOBLOCK, C., MCDERMOTT, D., RAM, A., VELOSO, M., WELD, D. et WILKINS, D. (1998). PDDL - the planning domain definition language. *AIPS-98 planning committee*, 3:14.
- [Ghorbel *et al.*, 2016] GHORBEL, E., BOUTTEAU, R., BOONAERT, J., SAVATIER, X. et LECOEUICHE, S. (2016). Vers une reconnaissance en ligne d'actions à partir de caméras rgb-d. In *Reconnaissance de Formes et Intelligence Artificielle*.
- [Gilbert, 2008] GILBERT, G. N. (2008). *Agent-based models*. Quantitative applications in the social sciences. Sage.
- [Gilbert, 1995] GILBERT, N. (1995). *Artificial societies : The computer simulation of social life*. UCL Press.
- [Giordani *et al.*, 2013] GIORDANI, S., LUJAK, M. et MARTINELLI, F. (2013). A distributed multi-agent production planning and scheduling framework for mobile robots. *Computers & Industrial Engineering*, 64(1):19–30.
- [Gorelick *et al.*, 2007] GORELICK, L., BLANK, M., SHECHTMAN, E., IRANI, M. et BASRI, R. (2007). Actions as space-time shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12).
- [Grewal *et al.*, 1998] GREWAL, D., KRISHNAN, R., BAKER, J. et BORIN, N. (1998). The effect of store name, brand name and price discounts on consumers' evaluations and purchase intentions. *Journal of Retailing*, 74(3):331 – 352. Research Perspective on Retail Pricing.
- [Grignard *et al.*, 2013] GRIGNARD, A., TAILLANDIER, P., GAUDOU, B., VO, D. A., HUYNH, N. Q. et DROGOU, A. (2013). Gama 1.6 : Advancing the art of complex agent-based modeling and simulation. In BOELLA, G., ELKIND, E., SAVARIMUTHU, B. T. R., DIGNUM, F. et PURVIS, M. K., éditeurs : *PRIMA 2013 : Principles and Practice of Multi-Agent Systems*, pages 117–131, Berlin, Heidelberg. pringer Berlin Heidelberg.
- [Guiot *et al.*, 2014] GUIOT, F., HENRY, L. et GALLAND, S. (2014). Multiagent model for the simulation of drivers with the jasim platform. In *3èmes Journées des Jeunes Chercheurs de l'UTBM (IngéDoc 2014)*. Doceo, UTBM, UTBM Press.
- [Hadid *et al.*, 2018] HADID, B., DUVIELLA, E. et LECOEUICHE, S. (2018). Improvement of a predictive data-driven model for rainfall-runoff global characterization of aa river. In *IEEE 4th International Forum on Research and Technology for Society and Industry, RTSI 2018, Palermo, Italy, September 10-13, 2018*, pages 1–6.
- [Hartigan et Wong, 1979] HARTIGAN, J. A. et WONG, M. A. (1979). Algorithm as 136 : A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1): 100–108.

BIBLIOGRAPHIE

- [Hassan, 2010] HASSAN, S. (2010). *Towards a Data-driven Approach for Agent-Based Modelling : Simulating Spanish Postmodernisation*. Thèse de doctorat, Universidad Complutense de Madrid, Madrid, Spain.
- [Haykin et al., 2009] HAYKIN, S. S. et al. (2009). *Neural networks and learning machines*. New York : Prentice Hall,.
- [Helbing et Molnár, 1995] HELBING, D. et MOLNÁR, P. (1995). Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286.
- [Höferlin et al., 2011] HÖFERLIN, M., HÖFERLIN, B., WEISKOPF, D. et HEIDEMANN, G. (2011). Interactive schematic summaries for exploration of surveillance video. *In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11*.
- [Holzinger, 2018] HOLZINGER, A. (2018). From machine learning to explainable ai. *In 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, pages 55–66.
- [Hongeng et Nevatia, 2003] HONGENG, S. et NEVATIA, R. (2003). Large-scale event detection using semi-hidden markov models. *In Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1455–1462.
- [Hoogendoorn et Bovy, 2004] HOOGENDOORN, S. et BOVY, P. (2004). Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B : Methodological*, 38(2): 169 – 190.
- [Hopkinson et al., 2006] HOPKINSON, K., WANG, X., GIOVANINI, R., THORP, J., BIRMAN, K. et COURRY, D. (2006). Epochs : a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components. *IEEE Transactions on Power Systems*, 21(2):548–558.
- [Hou et al., 2017] HOU, L., WAN, W., HWANG, J.-N., MUHAMMAD, R., YANG, M. et HAN, K. (2017). Human tracking over camera networks : a review. *EURASIP Journal on Advances in Signal*, 2017(1).
- [Howard, 1960] HOWARD, R. A. (1960). Dynamic programming and markov processes.
- [Hu, 1962] HU, M.-K. (1962). Visual pattern recognition by moment invariants, computer methods in image analysis. *IRE Transactions on Information Theory*, 8.
- [Huang et al., 2009] HUANG, K., SRIVASTAVA, S. K., CARTES, D. A. et SUN, L.-H. (2009). Market-based multiagent system for reconfiguration of shipboard power systems. *Electric Power Systems Research*, 79(4):550 – 556.
- [Jhuang et al., 2007] JHUANG, H., SERRE, T., WOLF, L. et POGGIO, T. (2007). A biologically inspired system for action recognition. *In 2007 IEEE 11th International Conference on Computer Vision*, pages 1–8.
- [Johnson, 1967] JOHNSON, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3): 241–254.
- [Karami et al., 2016] KARAMI, A., FLEURY, A., BOONAERT, J. et LECOEUQUE, S. (2016). User in the loop : Adaptive smart homes exploiting user feedback - state of the art and future directions. *Information*, 7(2):35.
- [Khalid et al., 2019a] KHALID, Q., FERNÁNDEZ, A., LUJAK, M. et DONIEC, A. (2019a). A group evacuation method for smart buildings. *In Technological Innovation for Industry and Service Systems : 10th IFIP WG 5.5/SOCOLNET Advanced Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2019, Costa de Caparica, Portugal, May 8–10, 2019, Proceedings*, page 206. Springer.

- [Khalid *et al.*, 2019b] KHALID, Q., FERNÁNDEZ, A., LUJAK, M. et DONIEC, A. (2019b). A two-phase context-aware approach to emergency evacuation in smart buildings. *In International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 234–245. Springer.
- [Kim *et al.*, 2013] KIM, S.-C., JEONG, Y.-S. et PARK, S.-O. (2013). Rfid-based indoor location tracking to ensure the safety of the elderly in smart home environments. *Personal and ubiquitous computing*, 17(8):1699–1707.
- [Klein, 2009] KLEIN, F. (2009). *Contrôle d'un Système Multi-Agents Réactif par Modélisation et Apprentissage de sa Dynamique Globale*. Thèse de doctorat, Université Nancy II.
- [Kohavi, 1995] KOHAVI, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *In Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95*, pages 1137–1143.
- [Kohonen, 1988] KOHONEN, T. (1988). Neurocomputing : Foundations of research. chapitre Self-organized Formation of Topologically Correct Feature Maps, pages 509–521. MIT Press, Cambridge, MA, USA.
- [Kubera *et al.*, 2011] KUBERA, Y., MATHIEU, P. et PICAULT, S. (2011). Ioda : an interaction-oriented approach for multi-agent based simulations. *Autonomous Agents and Multi-Agent Systems*, 23(3): 303–343.
- [Kucera *et al.*, 2014] KUCERA, A., PITNER, T. *et al.* (2014). Smart buildings : Semantic web technology for building information model and building management system. *In 2014 International Conference on Data and Software Engineering (ICODSE)*, pages 1–6. IEEE.
- [Kumar *et al.*, 2009] KUMAR, A., FALTINGS, B. et PETCU, A. (2009). Distributed constraint optimization with structured resource constraints. *In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '09*, pages 923–930.
- [Kumar, 1992] KUMAR, V. (1992). Algorithms for constraint-satisfaction problems : A survey. *AI magazine*, 13(1):32–32.
- [Kumar *et al.*, 2013] KUMAR, V., FENSEL, A. et FRÖHLICH, P. (2013). Context based adaptation of semantic rules in smart buildings. *In Proceedings of International Conference on Information Integration and Web-based Applications & Services*, page 719. ACM.
- [Lallement, 2008] LALLEMENT, J. (2008). *Time pressure effects on information processing*. Thèse de doctorat, Université de Tours / Région Centre.
- [Larson *et al.*, 2005] LARSON, J. S., BRADLOW, E. T. et FADER, P. S. (2005). An exploratory look at supermarket shopping paths. *International Journal of Research in Marketing*, 22(4):395 – 414.
- [Le *et al.*, 2009] LE, V., BOURAQADI, N., STINCKWICH, S., MORARU, V. et DONIEC, A. (2009). Making networked robots connectivity-aware. *In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3502–3507. IEEE.
- [Lesser et Corkill, 2014] LESSER, V. et CORKILL, D. (2014). Challenges for multi-agent coordination theory based on empirical observations. *In Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 1157–1160.
- [Li *et al.*, 2019a] LI, J., BOONAERT, J., DONIEC, A. et LOZENGUEZ, G. (2019a). Toward reliable estimations of urban traffic flows from machine learning and floating car data. *In 15th World Conference on Transport Research (WCTR 2019)*, Mumbai, Inde.
- [Li *et al.*, 2019b] LI, J., BOONAERT, J., DONIEC, A. et LOZENGUEZ, G. (2019b). Traffic flow multi-model with machine learning method based on floating car data. *In 6th International Conference on*

BIBLIOGRAPHIE

- Control, Decision and Information Technologies, CoDIT 2019, Paris, France, April 23-26, 2019*, pages 512–517.
- [Li *et al.*, 2015] LI, T., CHANG, H., WANG, M., NI, B., HONG, R. et YAN, S. (2015). Crowded scene analysis : A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(3):367–386.
- [Lujak *et al.*, 2017] LUJAK, M., BOURAQADI, N., DONIEC, A., FABRESSE, L., FLEURY, A., KARAMI, A. et LOZENGUEZ, G. (2017). Towards robots-assisted ambient intelligence. In *Multi-Agent Systems and Agreement Technologies - 15th European Conference, EUMAS 2017, and 5th International Conference, AT 2017, Évry, France, December 14-15, 2017, Revised Selected Papers*, pages 490–497.
- [Lujak et Doniec, 2018] LUJAK, M. et DONIEC, A. (2018). Towards distributed real-time coordination of shoppers’ routes in smart hypermarkets. In *Agreement Technologies - 6th International Conference, AT 2018, Bergen, Norway, December 6-7, 2018, Revised Selected Papers*, pages 223–238.
- [Mandiau, 1993] MANDIAU, R. (1993). *Contribution à la modélisation des univers multi-agents : génération d’un plan partagé*. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis.
- [Mandiau *et al.*, 2008] MANDIAU, R., CHAMPION, A., AUBERLET, J., ESPIÉ, S. et KOLSKI, C. (2008). Behaviour based on decision matrices for a coordination between agents in a urban traffic simulation. *Applied Intelligence*, 28(2):121–138.
- [Mandiau *et al.*, 2014] MANDIAU, R., VION, J., PIECHOWIAK, S. et MONIER, P. (2014). Multi-variable distributed backtracking with sessions. *Applied Intelligence*, 41(3):736–758.
- [Marc, 2005] MARC, F. (2005). *Planification multi-agent sous contraintes dans un contexte dynamique : applications aux simulations aériennes*. Thèse de doctorat, Paris 6.
- [McArthur *et al.*, 2004] MCARTHUR, S. D. J., STRACHAN, S. M. et JAHN, G. (2004). The design of a multi-agent transformer condition monitoring system. *IEEE Transactions on Power Systems*, 19(4):1845–1852.
- [Mercure *et al.*, 2016] MERCURE, J.-F., POLLITT, H., BASSI, A. M., VIÑUALES, J. E. et EDWARDS, N. R. (2016). Modelling complex systems of heterogeneous agents to better design sustainability transitions policy. *Global environmental change*, 37:102–115.
- [Merkuryeva et Bolshakovs, 2010] MERKURYEVA, G. et BOLSHAKOV, V. (2010). Vehicle schedule simulation with anylogic. In *2010 12th International Conference on Computer Modelling and Simulation*, pages 169–174.
- [Miller *et al.*, 2012] MILLER, S., RAMCHURN, S. D. et ROGERS, A. (2012). Optimal decentralised dispatch of embedded generation in the smart grid. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS ’12*, pages 281–288.
- [Monier *et al.*, 2010] MONIER, P., DONIEC, A., PIECHOWIAK, S. et MANDIAU, R. (2010). Metrics for the evaluation of discsp : Some experiments on multi-robot exploration. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2010, Toronto, Canada, August 31 - September 3, 2010*, pages 370–373.
- [Monier *et al.*, 2011] MONIER, P., DONIEC, A., PIECHOWIAK, S. et MANDIAU, R. (2011). Comparison of DCSP algorithms : A case study for multi-agent exploration. In *Advances on Practical*

- Applications of Agents and Multiagent Systems - 9th International Conference on Practical Applications of Agents and Multiagent Systems, PAAMS 2011, Salamanca, Spain, 6-8 April 2011*, pages 231–236.
- [Moore *et al.*, 1999] MOORE, D. J., ESSA, I. A. et HAYES, M. H. (1999). Exploiting human actions and object context for recognition tasks. *In Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 80–86.
- [Moreno *et al.*, 2019] MORENO, A., RODRÍGUEZ, J. J., BELTRÁN, D., SIKORA, A., JORBA, J. et CÉSAR, E. (2019). Designing a benchmark for the performance evaluation of agent-based simulation applications on hpc. *The Journal of Supercomputing*, 75(3):1524–1550.
- [Mwasilu *et al.*, 2014] MWASILU, F., JUSTO, J. J., KIM, E.-K., DO, T. D. et JUNG, J.-W. (2014). Electric vehicles and smart grid interaction : A review on vehicle to grid and renewable energy sources integration. *Renewable and sustainable energy reviews*, 34:501–516.
- [Nagata et Sasaki, 2002] NAGATA, T. et SASAKI, H. (2002). A multi-agent approach to power system restoration. *IEEE Transactions on Power Systems*, 17(2):457–462.
- [Nair *et al.*, 2005] NAIR, R., VARAKANTHAM, P., TAMBE, M. et YOKOO, M. (2005). Networked distributed pomdps : A synthesis of distributed constraint optimization and pomdps. *In Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1, AAAI'05*, pages 133–139.
- [North *et al.*, 2013] NORTH, M. J., COLLIER, N. T., OZIK, J., TATARA, E. R., MACAL, C. M., BRAGEN, M. et SYDELKO, P. (2013). Complex adaptive systems modeling with repast simphony. *Complex Adaptive Systems Modeling*, 1(1):3.
- [Nouasse *et al.*, 2016] NOUASSE, H., DONIEC, A., LOZENGUEZ, G., DUVIELLA, E., CHIRON, P., ARCHIMÈDE, B. et CHUQUET, K. (2016). Constraint satisfaction problem based on flow graph to study the resilience of inland navigation networks in a climate change context. *IFAC-PapersOnLine*, 49(12):331 – 336. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016.
- [Nouasse *et al.*, 2015] NOUASSE, H., RAJAOARISOA, L. H., DONIEC, A., DUVIELLA, E., CHUQUET, K., CHIRON, P. et ARCHIMÈDE, B. (2015). Study of drought impact on inland navigation systems based on a flow network model. *In XXV International Conference on Information, Communication and Automation Technologies, ICAT 2015, Sarajevo, Bosnia and Herzegovina, October 29-31, 2015*, pages 1–6.
- [Olson *et al.*, 2013] OLSON, E., STROM, J., GOEDDEL, R., MORTON, R., RANGANATHAN, P. et RICHARDSON, A. (2013). Exploration and mapping with autonomous robot teams. *Commun. ACM*, 56(3):62–70.
- [Pantic *et al.*, 2007] PANTIC, M., PENTLAND, A., NIJHOLT, A. et HUANG, T. S. (2007). Human computing and machine understanding of human behavior : A survey. *In Artificial Intelligence for Human Computing*, pages 47–71. Springer Berlin Heidelberg.
- [Park *et al.*, 2018] PARK, B. H., AZIZ, H. A., MORTON, A. et STEWART, R. (2018). High performance data driven agent-based modeling framework for simulation of commute mode choices in metropolitan area. *In 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3779–3784. IEEE.
- [Pavón et Gómez-Sanz, 2003] PAVÓN, J. et GÓMEZ-SANZ, J. (2003). Agent oriented software engineering with ingenias. *In Multi-Agent Systems and Applications III*, pages 394–403, Berlin, Heidelberg. Springer Berlin Heidelberg.

BIBLIOGRAPHIE

- [Pavon *et al.*, 2008] PAVON, J., SANSORES, C. et GOMEZ-SANZ, J. J. (2008). Modelling and simulation of social systems with ingenias. *Int. J. Agent-Oriented Softw. Eng.*, 2(2):196–221.
- [Perkins, 2001] PERKINS, C. E. (2001). Ad hoc networking. chapitre Ad Hoc Networking : An Introduction, pages 1–28. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [Picard, 2018] PICARD, G. (2018). Optimisation sous contraintes distribuée : une introduction au domaine. In *Distribution et Décentralisation - Vingt-sixièmes journées francophones sur les systèmes multi-agents, JFSMA 2018, Métabief, France, 10-12 Octobre 2018.*, pages 43–52.
- [Preuveneers *et al.*, 2004] PREUVENEERS, D., Van den BERGH, J., WAGELAAR, D., GEORGES, A., RIGOLE, P., CLERCKX, T., BERBERS, Y., CONINX, K., JONCKERS, V. et DE BOSSCHERE, K. (2004). Towards an extensible context ontology for ambient intelligence. In *European Symposium on Ambient Intelligence*, pages 148–159. Springer.
- [Proctor et Zandt, 2018] PROCTOR, R. et ZANDT, T. (2018). *Human Factors in Simple and Complex Systems*. Boca Raton : CRC Press.
- [Pudjianto *et al.*, 2007] PUDJIANTO, D., RAMSAY, C. et STRBAC, G. (2007). Virtual power plant and system integration of distributed energy resources. *IET Renewable Power Generation*, 1(1):10–16.
- [Quinlan, 1986] QUINLAN, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.
- [Ravindra Babu *et al.*, 2010] RAVINDRA BABU, T., NARASIMHA MURTY, M. et SUBRAHMANYA, S. V. (2010). Multiagent based large data clustering scheme for data mining applications. In *Active Media Technology*, pages 116–127, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Rehtanz, 2003] REHTANZ, C. (2003). *Autonomous Systems and Intelligent Agents in Power System Control and Operation (Power Systems)*. SpringerVerlag.
- [Remondino et Correndo, 2005] REMONDINO, M. et CORRENDO, G. (2005). Data mining applied to agent based simulation. In *Proceedings of the 19th European Conference on Modelling and Simulation*, Riga, Latvia.
- [Remondino et Correndo, 2006] REMONDINO, M. et CORRENDO, G. (2006). Mabs validation through repeated executing and data mining analysis. *International Journal of Simulation Systems, Science Technology*, 7(6):10–21.
- [Reynolds, 1999] REYNOLDS, C. (1999). Steering behaviors for autonomous characters. In *GDC - Game developers conference*, pages 762 – 782.
- [Robles et Kim, 2010] ROBLES, R. J. et KIM, T.-h. (2010). Context aware tools for smart home development. *International Journal of Smart Home*, 4(1):1–12.
- [Roche *et al.*, 2010] ROCHE, R., BLUNIER, B., MIRAOU, A., HILAIRE, V. et KOUKAM, A. (2010). Multi-agent systems for grid energy management : A short review. In *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*, pages 3341–3346. IEEE.
- [Rodriguez *et al.*, 2015] RODRIGUEZ, S., GALLAND, S. et GAUD, N. (2015). A new perspective on multi-agent environment with {SARL}. *Procedia Computer Science, The 10th International Conference on Future Networks and Communications (FNC 2015) / The 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015) Affiliated Workshops*, 56:526 – 531.
- [Rossi *et al.*, 2006] ROSSI, F., BEEK, P. v. et WALSH, T. (2006). *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA.
- [Rota et Thonnat, 2000] ROTA, N. et THONNAT, M. (2000). Activity recognition from video sequences using declarative models. In *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany, August 20-25, 2000*, pages 673–680.

- [Rouas *et al.*, 2006] ROUAS, J.-L., LOURADOUR, J. et AMBELLOUIS, S. (2006). Audio events detection in public transport vehicle. *In 2006 IEEE Intelligent Transportation Systems Conference*, pages 733–738. IEEE.
- [Rousseuw, 1987] ROUSSEUW, P. (1987). Silhouettes : A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65.
- [Royer et Toh, 1999] ROYER, E. M. et TOH, C.-K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55.
- [Ruiz *et al.*, 2009] RUIZ, N., COBELO, I. et OYARZABAL, J. (2009). A direct load control model for virtual power plant management. *IEEE Transactions on Power Systems*, 24(2):959–966.
- [Russell et Norvig, 2010] RUSSELL, S. J. et NORVIG, P. (2010). *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education.
- [Ryoo et Aggarwal, 2006] RYOO, M. S. et AGGARWAL, J. K. (2006). Recognition of composite human activities through context-free grammar based representation. *In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1709–1718.
- [Saboori *et al.*, 2011] SABOORI, H., MOHAMMADI, M. et TAGHE, R. (2011). Virtual power plant (vpp), definition, concept, components and types. *In 2011 Asia-Pacific Power and Energy Engineering Conference*, pages 1–4. IEEE.
- [Saffar, 2013] SAFFAR, I. (2013). *Vers une agentification de comportements observés : une approche originale basée sur l'apprentissage automatique pour la simulation d'un environnement réel*. Thèse de doctorat, Ecole doctorale Sciences pour l'Ingénieur (Lille), Université de Lille 1, Ecole des Mines de Douai.
- [Saffar *et al.*, 2011] SAFFAR, I., DONIEC, A., BOONAERT, J. et LECOEUUCHE, S. (2011). Multi-agent simulation design driven by real observations and clustering techniques. *In IEEE 23rd International Conference on Tools with Artificial Intelligence, ICTAI 2011, Boca Raton, FL, USA, November 7-9, 2011*, pages 555–560.
- [Saffar *et al.*, 2012] SAFFAR, I., DONIEC, A., BOONAERT, J. et LECOEUUCHE, S. (2012). Conception de simulations multi-agents : vers une modélisation basée sur les observations des comportements réels. *In 18ème Congrès francophone Reconnaissance de formes et Intelligence Artificielle (RFIA 12)*, Lyon.
- [Samek *et al.*, 2017] SAMEK, W., WIEGAND, T. et MÜLLER, K. (2017). Explainable artificial intelligence : Understanding, visualizing and interpreting deep learning models. *CoRR*, abs/1708.08296.
- [Savaux *et al.*, 2019] SAVAUX, J., VION, J., PIECHOWIAK, S., MANDIAU, R., MATSUI, T., HIRAYAMA, K., YOKOO, M., ELMANE, S. et SILAGHI, M. (2019). Privacy stochastic games in distributed constraint reasoning. *Annals of Mathematics and Artificial Intelligence*, pages 1–25.
- [Shannon, 1998] SHANNON, R. E. (1998). Introduction to the art and science of simulation. *In 1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)*, volume 1, pages 7–14. IEEE.
- [Sharma *et al.*, 2018] SHARMA, A. S., LYNCH, E. M., KRISHNAMURTHY, V. et KALNAY, E. (2018). Data-driven Modeling, Prediction and Predictability : The Complex Systems Framework. *AGU Fall Meeting Abstracts*.
- [Shechtman et Irani, 2005] SHECHTMAN, E. et IRANI, M. (2005). Space-time behavior based correlation. *In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 405–412.

BIBLIOGRAPHIE

- [Shoham, 1993] SHOHAM, Y. (1993). Agent-oriented programming. *Artificial intelligence*, 60(1):51–92.
- [Sigaud et Buffet, 2008a] SIGAUD, O. et BUFFET, O. (2008a). *Processus décisionnels de Markov en intelligence artificielle*, volume 1 - principes généraux et applications de IC2 - informatique et systèmes d'information. Lavoisier - Hermes Science Publications.
- [Sigaud et Buffet, 2008b] SIGAUD, O. et BUFFET, O. (2008b). *Processus décisionnels de Markov en intelligence artificielle*, volume 2 - Méthodes avancées et applications de IC2 - informatique et systèmes d'information. Lavoisier - Hermes Science Publications.
- [Singh et al., 2018] SINGH, M., MARATHE, A., MARATHE, M. V. et SWARUP, S. (2018). Behavior model calibration for epidemic simulations. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1640–1648. International Foundation for Autonomous Agents and Multiagent Systems.
- [Skarvelis-Kazakos et al., 2016] SKARVELIS-KAZAKOS, S., PAPADOPOULOS, P., UNDA, I. G., GORMAN, T., BELAIDI, A. et ZIGAN, S. (2016). Multiple energy carrier optimisation with intelligent agents. *Applied Energy*, 167:323 – 335.
- [Smith et al., 2008] SMITH, D. E., FRANK, J. et CUSHING, W. (2008). The anml language. In *The ICAPS-08 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*.
- [Stavropoulos et al., 2012] STAVROPOULOS, T. G., VRAKAS, D., VLACHAVA, D. et BASSILIADES, N. (2012). Bonsai : a smart building ontology for ambient intelligence. In *Proceedings of the 2nd international conference on web intelligence, mining and semantics*, page 30. ACM.
- [Streltchenko et al., 2005] STRELTCHENKO, O., YESHA, Y. et FININ, T. (2005). *Multi-Agent Simulation of Financial Markets*, pages 393–419. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Sylvain, 2016] SYLVAIN, A. (2016). *Modèle comportemental d'agent-client dirigé par des données réelles : application à la simulation d'espaces de vente*. Thèse de doctorat, Ecole doctorale Sciences pour l'Ingénieur (Lille), Université de Lille 1, Ecole des Mines de Douai.
- [Sylvain et al., 2014] SYLVAIN, A., DONIEC, A., MANDIAU, R. et LECOEUICHE, S. (2014). Purchase intention based model for a behavioural simulation of sale space. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03*, pages 318–324. IEEE Computer Society.
- [Tan et al., 2006] TAN, P.-N., STEINBACH, M. et KUMAR, V. (2006). *Introduction to Data Mining*. Pearson Education.
- [Teknomo, 2006] TEKNOMO, K. (2006). Application of microscopic pedestrian simulation model. *Transportation Research Part F : Traffic Psychology and Behaviour*, 9(1):15 – 27.
- [Tesone et al., 2018] TESONE, P., POLITO, G., BOURAQADI, N., DUCASSE, S. et FABRESSE, L. (2018). Dynamic software update from development to production. *Journal of Object Technology*, 17(1):1 :1–36.
- [Thang Pham et al., 2014] THANG PHAM, V., RAÏEVSKY, C. et JAMONT, J.-P. (2014). A multiagent approach using model-based predictive control for an irrigation canal. In *Trends in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection*, pages 87–94. Springer International Publishing.
- [Tisue et Wilensky, 2004] TISUE, S. et WILENSKY, U. (2004). NetLogo : a simple environment for modeling complexity.

- [Tlig *et al.*, 2013] Tlig, M., BUFFET, O. et SIMONIN, O. (2013). Reactive coordination rules for traffic optimization in road sharing problems. *In International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 61–72. Springer.
- [Tsitsiklis *et al.*, 1986] TSITSIKLIS, J., BERTSEKAS, D. et ATHANS, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812.
- [Tufféry, 2005] TUFFÉRY, S. (2005). *Data mining et statistique décisionnelle : l'intelligence dans les bases de données*. Technip.
- [Turaga *et al.*, 2008] TURAGA, P., CHELLAPPA, R., SUBRAHMANIAN, V. S. et UDREA, O. (2008). Machine recognition of human activities : A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488.
- [Verfaillie et Jussien, 2005] VERFAILLIE, G. et JUSSIEN, N. (2005). Constraint solving in uncertain and dynamic environments : A survey. *Constraints*, 10(3):253–281.
- [Wagner, 2003] WAGNER, G. (2003). The agent-object-relationship metamodel : Towards a unified view of state and behavior. *Inf. Syst.*, 28(5):475–504.
- [Walsh, 2002] WALSH, T. (2002). Stochastic constraint programming. *In Proceedings of ECAI*, pages 111–115.
- [Wijerathne *et al.*, 2018] WIJERATHNE, L., PETPRAKOB, W., AGUILAR, L., HORI, M. et ICHMURA, T. (2018). Scalable hpc enhanced agent based system for simulating mixed mode evacuation of large urban areas. *Transportation research procedia*, 34:275–282.
- [Wooldridge *et al.*, 2000] WOOLDRIDGE, M., JENNINGS, N. R. et KINNY, D. (2000). The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312.
- [Yamauchi, 1998] YAMAUCHI, B. (1998). Frontier-based exploration using multiple robots. *In Proceedings of the Second International Conference on Autonomous Agents*, AGENTS '98, pages 47–53.
- [Yang et Chen, 2018] YANG, B. et CHEN, Y. (2018). Evolution model and simulation of logistics outsourcing for manufacturing enterprises based on multi-agent modeling. *Cluster Computing*.
- [Yokoo, 1995] YOKOO, M. (1995). Asynchronous weak-commitment search for solving distributed constraint satisfaction problems. *In International Conference on Principles and Practice of Constraint Programming*, pages 88–102. Springer.
- [Yokoo et Hirayama, 2001] YOKOO, M. et HIRAYAMA, K. (2001). Distributed breakout algorithm for solving distributed constraint satisfaction problems. *ICMAS*.
- [Yokoo *et al.*, 1992] YOKOO, M., ISHIDA, T., DURFEE, E. H. et KUWABARA, K. (1992). Distributed constraint satisfaction for formalizing distributed problem solving. *In 1992 12th International Conference on Distributed Computing System*, pages 614–621. IEEE.
- [Young *et al.*, 2001] YOUNG, R., LESPERANCE, R. et MEYER, W. W. (2001). The gaussian derivative model for spatial-temporal vision : I. cortical model. *Spatial Vision*, 14(3):261–319.
- [Zhang *et al.*, 2016] ZHANG, H., VOROBAYCHIK, Y., LETCHFORD, J. et LAKKARAJU, K. (2016). Data-driven agent-based modeling, with application to rooftop solar adoption. *Autonomous Agents and Multi-Agent Systems*, 30(6):1023–1049.

BIBLIOGRAPHIE
