



HAL
open science

Prise en compte des comportements anticipatifs dans la coordination multi-agent : application à la simulation de trafic en carrefour

Arnaud Doniec

► To cite this version:

Arnaud Doniec. Prise en compte des comportements anticipatifs dans la coordination multi-agent : application à la simulation de trafic en carrefour. Intelligence artificielle [cs.AI]. Université de Valenciennes, 2006. Français. NNT : . tel-01992947

HAL Id: tel-01992947

<https://hal.science/tel-01992947>

Submitted on 24 Jan 2019

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.

THÈSE

présentée à
l'Université de Valenciennes et du Hainaut-Cambrésis
pour l'obtention du

Doctorat de l'Université de Valenciennes et du Hainaut-Cambrésis
spécialité Automatique et Informatique des Systèmes Industriels et Humains

mention: Informatique

par
Arnaud DONIEC

Prise en compte des comportements anticipatifs dans la coordination multi-agent : application à la simulation de trafic en carrefour

thèse soutenue publiquement le 25 Octobre 2006 devant le jury composé de:

Président:

Philippe MATHIEU, Professeur LIFL, Université des Sciences et Technologies de Lille

Rapporteurs:

Marie-Pierre GLEIZES, Maître de conférences Hdr IRIT, Université Paul Sabatier (Toulouse III)
Olivier BOISSIER, Professeur École des Mines de Saint-Etienne

Examineurs:

Domenico GATTUSO, Professeur Università degli Studi Mediterranea di Reggio Calabria

Directeur :

René MANDIAU, Professeur Université de Valenciennes et du Hainaut-Cambrésis

co-Directeurs :

Sylvain PIECHOWIAK, Professeur Université de Valenciennes et du Hainaut-Cambrésis
Stéphane ESPIÉ, Hdr, Directeur de l'unité MSIS INRETS

Remerciements

Je débute cette page en remerciant la région Nord-Pas-de-Calais et l'INRETS pour avoir financé ce travail de recherche au travers ma bourse de thèse.

Je tiens ensuite à remercier Marie-Pierre Gleizes et Olivier Boissier pour avoir accepté de rapporter ce travail malgré leurs responsabilités et leur emploi du temps chargé. Je leur suis très reconnaissant pour leurs lectures attentives du manuscrit : leurs commentaires et leurs questions ont été pour moi très constructifs.

Je remercie également les autres membres du jury : Philippe Mathieu pour avoir accepté de présider durant ma soutenance et Domenico Gattuso pour m'avoir accordé une partie de son temps à l'écriture d'un rapport détaillé.

J'adresse mes remerciements tout particulier aux professeurs René Mandiau et Sylvain Piechowiak pour leur encadrement de qualité tout au long de la thèse. Je leur exprime ma plus profonde gratitude pour leurs encouragements, leur disponibilité et leur soutien durant ces trois années. Merci de m'avoir fait découvrir l'intelligence artificielle et de m'avoir initié au monde de la recherche. Je remercie également Stéphane Espié, directeur de l'unité recherche MSIS et instigateur du projet ARCHISIM, pour m'avoir accueilli et encadré pendant les deux années passées à l'INRETS. Son aide et ses conseils ont largement contribué au bon déroulement de cette thèse.

Je remercie les membres de l'équipe MSIS pour leur sympathie et leur gentillesse. Merci à Fabrice pour ses chansons (et parodies de chanson) et ses visuels 3D toujours au top, merci à Jean-Michel pour ses nombreuses aides au débogage, ses conseils et ses nombreux apéros, merci à Vincent, Lamri, Hakim, Jacky et Pierre pour leurs bonnes humeurs, merci à Joelle, LaeTi et Florence pour avoir apporté leur touche féminine au sein de l'équipe.

Je remercie les membres de l'équipe RAIHM et son responsable Christophe Kolski. Merci à Sophie pour les quelques mois passés dans son bureau et ses encouragements en début de thèse.

Merci également aux moniteurs de la promotion 2004 pour les moments de détente passés lors des formations CIES.

Je tiens aussi à remercier toutes les personnes, rencontrées durant mon cursus et ayant suscité mon intérêt pour l'informatique et la recherche. Un merci tout particulier à Hervé Cordier, Jean-Philippe Delcroix, Vincent Poirriez, Georges Mariano, Dorian Petit, ...

Je termine cette page en remerciant Valérie pour tous ces week-end passés à relire avec moi articles, rapports, chapitres de thèse... Ses encouragements y sont pour beaucoup dans l'aboutissement de ce travail.

Table des matières

Remerciements	i
Table des figures	vii
Introduction générale	1
1 L’approche multi-agent appliquée à la simulation de trafic	5
1.1 Simulation de trafic routier	6
1.1.1 Trafic routier : définition, études et simulation	6
1.1.2 Simulation de trafic : critères de classification	7
1.1.3 Conception des modèles de simulation	9
1.1.4 Modélisation de la tâche de conduite	11
1.2 Approche multi-agent et simulation	12
1.2.1 Les systèmes multi-agents : motivations, concepts et propriétés	12
1.2.2 Éléments d’un système multi-agent	14
1.2.3 Simulation multi-agent	16
1.3 Le modèle de trafic ARCHISIM	17
1.3.1 Positionnement du modèle	17
1.3.2 Architecture informatique	18
1.3.3 Chaîne de production de trafic sous ARCHISIM	22
2 Simulation de trafic : gestion des conflits et coordination multi-agent	27
2.1 Problématique de simulation de trafic en carrefour	28
2.1.1 Conduite en file / franchissement d’intersection	28
2.1.2 Les approches existantes	31
2.1.3 Synthèse	38
2.2 Coordination en environnement dynamique	38
2.2.1 Coordination : généralités	38
2.2.2 Mécanismes de coordination d’agents cognitifs	40

2.2.3	Mécanismes de coordination d'agents réactifs	42
2.2.4	Synthèse	44
2.3	Mécanisme de coordination pour la simulation de trafic en carrefour	45
2.3.1	Fondement théorique	45
2.3.2	Théorie des jeux et systèmes multi-agents	48
2.3.3	Modélisation	50
2.3.4	Résolution des conflits	53
2.3.5	Analyse et critique	56
3	Modèles de comportements opportunistes et anticipatifs	61
3.1	Comportements opportunistes	62
3.1.1	Norme dans les systèmes multi-agents	62
3.1.2	Sélection des agents en amont de la coordination	65
3.1.3	Reconnaissance et perception des priorités	66
3.2	L'anticipation	69
3.2.1	Anticipation et comportement	70
3.2.2	Formalisation de l'anticipation	71
3.2.3	L'anticipation en intelligence artificielle	71
3.2.4	Conclusion	76
3.3	Un modèle d'anticipation fondé sur les réseaux de contraintes	76
3.3.1	Réseaux de contraintes	76
3.3.2	Formalisation de l'anticipation	79
3.3.3	Algorithmes d'anticipation	83
3.3.4	Instanciation du modèle d'anticipation dans le cadre de la simulation com- portementale de trafic	86
4	Application du modèle opportuniste / anticipatif à la simulation de trafic ARCHISIM	93
4.1	Choix d'implémentation	94
4.1.1	Définition des états non désirés	94
4.1.2	Distance d'anticipation	96
4.1.3	Choix de l'algorithme de propagation	98
4.1.4	Complexité en pratique	105
4.2	Intégration des comportements opportunistes et anticipatifs à ARCHISIM	106
4.2.1	Cycle de décision de l'agent	106
4.2.2	Couplage anticipation / coordination	108
4.3	Conclusion	109

5 Évaluations, validations et perspectives	111
5.1 Expérimentations sur la base de scenarii	112
5.1.1 Réduction des phénomènes d'oscillations	112
5.1.2 Anticipation et interblocages en carrefour isolé	114
5.1.3 Succession de carrefours	119
5.2 Simulation d'un carrefour réel et validation	122
5.2.1 Démarche	122
5.2.2 Résultats de simulation	124
5.2.3 Reproductibilité de la simulation	127
5.3 Perspectives	127
5.3.1 Perspectives de recherche dans le cadre d'ARCHISIM	128
5.3.2 Perspectives générales	130
Conclusion générale	133
Bibliographie	137

Table des figures

1.1	Principe de fonctionnement d'un simulateur de trafic	7
1.2	Granularité et niveau de détails des simulations de trafic	8
1.3	Effet accordéon	10
1.4	Émergence dans les systèmes multi-agents	13
1.5	Architecture globale d'ARCHISIM	19
1.6	Couplage entre simulation de trafic et simulateur de conduite	20
1.7	Repérage dans l'environnement routier simulé	21
1.8	Logiciels dr^2 et Sim^2	23
1.9	Outils de simulation développés autour d'ARCHISIM	24
2.1	Différentes situations de fusions et croisements de flux	28
2.2	Situation conflictuelle en conduite en file (insertion sur autoroute)	29
2.3	Conflits en carrefour	30
2.4	Gestion de deux carrefours à feux proches dans le logiciel SimTraffic	32
2.5	Simulation au centre des carrefours vue comme un problème d'ordonnancement	33
2.6	Simulation en carrefour (conduite à gauche) à l'aide d'automate cellulaire [Ruskin et Wang, 2002]	35
2.7	Algorithme de décision pour des mobiles en situation de carrefours [Reece et Shafer, 1993]	36
2.8	Situation d'interblocage nécessitant l'intervention d'un processus centralisé [Paruchuri <i>et al.</i> , 2002]	37
2.9	Algorithme de résolution de conflits spatiaux utilisant le modèle de satisfaction altruisme [Simonin, 2001]	43
2.10	Situations élémentaires en carrefour	52
2.11	Relations entre agents considérées dans le cadre de la coordination en carrefour	53
2.12	Décomposition du mécanisme de coordination utilisé dans ARCHISIM pour la résolution des conflits	55
2.13	Pourcentage de conflits non résolus et d'information connue en fonction du nombre de joueurs [Champion, 2003]	56
2.14	Exemple d'interblocage	58
2.15	Autres exemples d'interblocage	58
3.1	Distances aux conflits en carrefour	65
3.2	Franchissement de Stop et choix des agents pour la coordination	68
3.3	Niveau d'anticipation dans les systèmes artificiels [Butz <i>et al.</i> , 2003]	72
3.4	Mécanisme d'anticipation d'un classeur dans ACS [Gerard, 2002]	73

3.5	L'anticipation selon Rosen	75
3.6	Architecture agent pour l'anticipation préventive [Davidsson, 2003]	75
3.7	Exemple de réseau de contraintes	77
3.8	Arc-consistance	78
3.9	Anticipation linéaire et anticipation arborescente (inspiré de [Davidsson, 2003])	80
3.10	Trois agents évoluant sur une grille	82
3.11	Représentations mentales d'une situation de carrefour	89
4.1	Remontée de queue entre deux carrefours	95
4.2	Zone d'anticipation	97
4.3	Algorithme AC-3 [Mackworth, 1977a]	99
4.4	Algorithme dnAC-4 [Bessière, 1992]	101
4.5	Exemple de propagation de la relation <i>bph</i>	103
4.6	Exemple de propagation de la relation <i>bpha</i>	104
4.7	Exemple de propagation de la relation <i>bpr</i>	104
4.8	Exemple de propagation des relations	105
4.9	Cycle de décision d'un agent	107
5.1	Décélération d'un agent en approche d'un carrefour	113
5.2	Impressions écrans d'une simulation sans anticipation sur un scénario d'interblocages	115
5.3	Impressions écrans d'une simulation avec anticipation sur un scénario d'interblocage	116
5.4	Courbe de vitesse du mobile 13	117
5.5	Courbe de l'accélération du mobile 13	117
5.6	Variation du nombre d'interblocages en fonction du débit simulé	118
5.7	Impressions écrans d'une simulation sans anticipation sur un scénario impliquant deux carrefours successifs	120
5.8	Impressions écrans d'une simulation avec anticipation sur un scénario impliquant deux carrefours successifs	121
5.9	Plan de l'infrastructure du carrefour réel (Italie, Reggio de Calabre)	122
5.10	Photos du carrefour "Roma-Zerbi"	123
5.11	Débits obtenus en simulation pour l'axe nord / sud	124
5.12	Débits obtenus en simulation pour l'axe sud / nord	125
5.13	Débits obtenus en simulation pour l'axe est / ouest	126
5.14	Débits obtenus en simulation pour l'axe ouest / est	126
5.15	Variation de l'EQMR en fonction de la durée du pas de temps	128
5.16	Extension possible du modèle d'anticipation pour la gestion de l'accélération latérale	129

Introduction générale

L'introduction récente des systèmes multi-agents dans la conception d'outils de simulation permet d'améliorer le réalisme et d'élever le niveau de complexité pouvant être simulé. Le trafic routier fait partie des phénomènes complexes qui sont difficilement modélisables de manière analytique. Les outils actuels de simulation de trafic reposent sur des modèles mathématiques faisant intervenir différentes variables cinématiques. Dans ces modèles, le cas des intersections est généralement traité de manière simpliste. L'approche couramment utilisée consiste à se ramener à un problème d'ordonnement de flux de véhicules entre les différentes branches du carrefour. Les simplifications imposées par une telle approche induisent des simulations de comportements peu réalistes notamment à l'intérieur des carrefours.

L'unité de recherche MSIS (Modélisation, Simulation et Simulateur) de l'INRETS (Institut National de Recherche sur les Transports et leur Sécurité) développe depuis une quinzaine d'années des outils de simulation dédiés à l'étude du trafic et du comportement des conducteurs. L'une des problématiques étudiées par l'unité MSIS concerne la conception de simulateurs de conduite capables d'immerger un conducteur réel dans un trafic routier interactif et réaliste. L'approche utilisée à l'INRETS est originale : elle combine l'utilisation d'un modèle dit "comportemental" de trafic (ARCHISIM) avec une architecture distribuée de simulation de conduite (SIM²).

L'approche comportementale utilisée dans ARCHISIM est issue d'une réflexion sur les apports des systèmes multi-agents dans la simulation de trafic routier. Elle consiste à considérer le trafic routier comme un phénomène émergent résultant des actions et interactions entre les différents acteurs de la route : les usagers (automobilistes, motards, piétons, etc), l'infrastructure routière et la signalisation. Le modèle informatique sous-jacent applique les principes des systèmes multi-agents. Chaque acteur du système de trafic est un agent disposant d'objectifs et de caractéristiques propres, et faisant preuve d'autonomie lors de son déplacement sur le réseau routier simulé. La prise de décision de chaque agent est individuelle et se base sur une perception locale de son environnement.

Le travail présenté dans cette thèse s'intègre dans le cadre de l'extension du modèle de trafic d'ARCHISIM pour la gestion des réseaux urbains. Une telle extension nécessite d'être capable de simuler correctement une intersection, or la simulation réaliste des phénomènes de trafic en carrefour est un problème difficile. Un carrefour se présente en effet comme un espace non-structuré à la géométrie parfois complexe où s'effectuent des fusions et des intersections de flux de véhicules.

Des premiers travaux ont été menés sur ce sujet dans le cadre d'une coopération entre l'unité MSIS et l'équipe RAIHM (Raisonnement Automatique et Interaction Homme-Machine) du LAMIH (Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines). Cette première étude a montré que la simulation de trafic en carrefour pouvait se ramener à un problème de coordination multi-agent entre les conducteurs simulés [Champion, 2003].

L'approche la plus courante en coordination multi-agent fait l'hypothèse que les agents sont motivés collectivement par un but commun et coopèrent ensemble à l'accomplissement de celui-ci. La coordination proposée dans [Champion, 2003] se veut plus compétitive en considérant que les conducteurs simulés sont motivés par des objectifs qui leur sont propres mais interagissent tout de même ensemble de manière à éviter les accidents et les blocages. Ce choix est motivé et conforté par différentes études en psychologie de la conduite qui montrent que la tâche de conduite est plutôt de nature individualiste et compétitive.

Le mécanisme de coordination proposé à l'issue de cette première étude repose sur l'identification et la modélisation des conflits entre véhicules simulés dans différentes configurations. L'aspect comportemental des conducteurs est traité en amont de la coordination par l'ébauche d'une reconnaissance des différentes relations de priorité en fonction du contexte. Les expérimentations réalisées à l'issue de cette étude montrent que le modèle ne permet pas de gérer correctement l'entrée en carrefour d'un flux non-prioritaire. Par ailleurs, les phénomènes d'interblocages (véhicules se bloquant mutuellement et entraînant une paralysie complète du trafic) ne sont pas pris en compte.

Nous abordons dans ce mémoire l'introduction de comportements anticipatifs et opportunistes au sein de la simulation. Nous définissons la notion de comportement opportuniste comme un comportement circonstancié, subordonnant les normes et règles du système à l'intérêt momentané de l'agent. Un tel comportement est en parfaite adéquation avec la tâche de conduite mais accentue les risques d'interblocages précédemment évoqués. Nous introduisons donc en complément du comportement opportuniste des capacités anticipatives permettant aux agents d'éviter ces phénomènes d'interblocages.

Le chapitre 1 est consacré à une présentation du contexte de nos travaux à savoir l'approche multi-agent appliquée à la simulation de trafic. Après avoir défini le trafic routier, nous introduisons différentes notions relatives à la classification des outils de simulation et aux modèles sous-jacents. La seconde partie du chapitre vise à introduire les principaux concepts des systèmes multi-agents qui nous permettent d'exposer le modèle de trafic ARCHISIM en troisième partie.

Le chapitre 2 se focalise sur la gestion des conflits dans les simulations de trafic fondées sur une approche multi-agent. Nous présentons en premier lieu la problématique de simulation de trafic en carrefour. La seconde partie est consacrée à la coordination multi-agent : les différents mécanismes existants dans la littérature sont d'abord exposés, en particulier les travaux en planification [Marc, 2005] et en coordination réactive [Simonin, 2001]. La dernière section du chapitre résume et analyse les travaux exposés dans [Champion, 2003] visant à permettre la simulation de trafic en carrefour dans ARCHISIM.

Le chapitre 3 présente notre contribution. Elle consiste en la proposition d'un modèle de sélection d'actions fondé sur des comportements à la fois opportunistes et anticipatifs. La première section détaille la mise en place de comportements opportunistes sur la base d'une reconnaissance égo-centrée du contexte de coordination. La seconde partie du chapitre définit le concept d'anticipation dans un contexte général puis dans le cadre de l'intelligence artificielle. Nous retenons en particulier l'approche de l'anticipation proposée par Rosen pour laquelle nous proposons une formalisation en dernière partie. Celle-ci est fondée sur la manipulation et le traitement de réseaux de contraintes. Un algorithme générique d'anticipation est introduit puis instancié dans le cadre de la simulation de trafic.

Le chapitre 4 détaille l'application du modèle de comportements opportunistes et anticipatifs à la simulation de trafic ARCHISIM. Les différents choix effectués en terme d'implémentation sont exposés et justifiés. Nous analysons l'incidence de ces choix sur la complexité algorithmique de l'ap-

proche. Le couplage entre les algorithmes d'anticipation et de coordination est présenté en seconde section.

Le chapitre 5 est consacré aux différentes expérimentations menées pour l'évaluation de notre contribution. La première section présente un ensemble d'expérimentations fondées sur l'utilisation de différents scénarii permettant d'évaluer les comportements locaux des agents. Ceux-ci sont ainsi placés dans des situations de trafic pour lesquelles un comportement bien précis est attendu. La seconde section aborde la validation du trafic d'un point de vue global. Un carrefour réel est simulé et la comparaison entre des données de trafic réelles et simulées permet de vérifier que les situations de trafic émergeant des comportements locaux sont réalistes. La dernière partie du chapitre présente différentes perspectives à nos travaux de recherche.

Chapitre 1

L'approche multi-agent appliquée à la simulation de trafic

Sommaire

1.1 Simulation de trafic routier	6
1.1.1 Trafic routier : définition, études et simulation	6
1.1.2 Simulation de trafic : critères de classification	7
1.1.3 Conception des modèles de simulation	9
1.1.4 Modélisation de la tâche de conduite	11
1.2 Approche multi-agent et simulation	12
1.2.1 Les systèmes multi-agents : motivations, concepts et propriétés	12
1.2.2 Éléments d'un système multi-agent	14
1.2.3 Simulation multi-agent	16
1.3 Le modèle de trafic ARCHISIM	17
1.3.1 Positionnement du modèle	17
1.3.2 Architecture informatique	18
1.3.3 Chaîne de production de trafic sous ARCHISIM	22

La simulation est devenue depuis plusieurs années un outil incontournable pour tous les problèmes liés à la compréhension des phénomènes complexes, à la prise de décision, à la conception ou encore à la formation.

D'une manière générale, la simulation consiste en une reproduction du comportement dynamique d'un système réel à partir de modèles. Un modèle est une construction mentale abstraite et simplifiée de la réalité permettant d'analyser et de comprendre le système à simuler. Le comportement dynamique d'un système peut être vu comme une succession de changements de ses états. Pendant très longtemps, les comportements dynamiques d'un système simulé ont été décrits à l'aide d'outils mathématiques : équations aux différences pour des changements d'états s'opérant en domaine discret ou équations différentielles dans le cas du continu.

La modélisation par équations à l'avantage de fournir une approche parfaitement formalisée de la simulation : universalité du langage mathématique et outils de résolution associés aux équations. Ce type de modélisation atteint cependant ses limites lorsqu'il s'agit de simuler des phénomènes de

grande ampleur ou lorsque la remise en cause du modèle doit être fréquente (par exemple, lors de l'ajout de nouvelles entités dans la simulation).

L'introduction de techniques issues de l'intelligence artificielle distribuée et des systèmes multi-agents dans la conception des outils de simulation permet de reproduire ces phénomènes s'accommodant mal d'une modélisation mathématique : comportements des mammifères marins [Ramat *et al.*, 1998], phénomènes sociaux [Doran, 1998], déplacements de piétons en environnement urbain ([Grislin-Le Strugeon *et al.*, 2004], [Hanon *et al.*, 2006]).

Ce premier chapitre expose les contextes théoriques et applicatifs de nos travaux. Nous commençons par définir le trafic routier et présentons les différents critères et approches utilisés en simulation de trafic. La seconde partie se focalise sur l'approche multi-agent. Nous présentons les concepts généraux des systèmes multi-agents et leurs applications en simulation. La dernière partie de ce chapitre est consacrée au modèle de trafic ARCHISIM développé par l'INRETS.

1.1 Simulation de trafic routier

1.1.1 Trafic routier : définition, études et simulation

Le trafic routier peut se définir comme l'ensemble des phénomènes complexes résultant du déplacement d'usagers sur un réseau routier de capacité limitée. Le "système de trafic" est caractérisé par une offre constituée d'une infrastructure routière conçue par un opérateur¹ pour répondre à un optimum collectif et par une demande toujours croissante des usagers qui cherchent à atteindre un optimum individuel.

Les études de trafic ont donc pour ambition l'identification et la compréhension de l'ensemble des situations impliquant le couple usager / infrastructure routière. Ces études, menées depuis de nombreuses années ([Greenshields, 1935],[Rothrock et Keefer, 1957]), sont à la base de la prévision et du contrôle du système de trafic. Les applications peuvent aller de la diffusion d'informations en temps-réel (affichage des temps de parcours sur un périphérique) ou prévisionnelle (Bison Futé) à la mise au point de contrôleurs de feux en carrefour pour optimiser le débit des flux de véhicules.

Mener une étude de trafic nécessite de pouvoir recueillir différentes données inhérentes au couple conducteur / véhicule placé dans la situation à étudier. Différentes mesures sont réalisées par le biais d'équipements et de capteurs qui peuvent être placés au niveau de l'infrastructure routière ou directement embarqués sur le véhicule. L'utilisation combinée de ces différents types de mesures permet d'obtenir des indicateurs sur l'activité du conducteur (niveau individuel) et ses effets sur son environnement routier (niveau collectif).

Jusqu'à récemment, de telles études de trafic ne pouvaient être réalisées qu'en conditions réelles (sur route ou sur piste) et nécessitaient un aménagement particulier de l'infrastructure (installation de boucles magnétiques, de caméras de surveillance) et l'instrumentation d'un véhicule. Depuis quelques années, il est possible de réaliser certaines études en laboratoire sur un simulateur de conduite. Les avantages de cette démarche sont nombreux :

- réduction des coûts notamment ceux liés à l'instrumentation du véhicule,
- réduction des risques liés aux expérimentations menées sur route,

1. ensemble des personnes qui définissent l'utilisation de l'infrastructure en introduisant des plans de circulation et de déplacement urbain, des règles d'affectation de trafic, ...

- reproductibilité à l’infini d’une large variété de situations de trafic,
- mise au point simplifiée des protocoles expérimentaux de l’étude.

La simulation de trafic trouve des applications dans de nombreux domaines. Elle peut être utilisée comme un outil d’aide à la décision pour des problèmes d’investissement ayant trait au trafic routier : étude de l’impact de l’aménagement de nouvelles infrastructures routières [Espié *et al.*, 2002] et autoroutières [Piroird, 1997]. La simulation peut être également utilisée comme un outil de développement et d’évaluation facilitant la mise au point de nouveaux équipements automobiles et de systèmes d’aide à la conduite [Auberlet *et al.*, 2003], [Tricot, 2005], [Tricot *et al.*, 2006]. Enfin, une demande très forte s’instaure dans le domaine de la formation, où l’utilisation combinée d’une simulation de trafic et d’un simulateur offre un environnement virtuel de conduite permettant l’apprentissage des bases de la conduite [Flipo, 2000].

Au vue de l’étendue des domaines couverts par la simulation de trafic et de la multitude d’outils de simulation existants aujourd’hui sur le marché, il convient d’introduire quelques éléments de classification qui nous permettront par la suite de situer nos travaux.

1.1.2 Simulation de trafic : critères de classification

Tous les outils de simulation de trafic reposent sur un même principe d’utilisation. Ils prennent en entrée un réseau routier ainsi qu’une demande de trafic et fournissent en sortie des résultats statistiques décrivant de manière quantitative et qualitative les flots de véhicules durant la simulation. La plupart des outils offre en complément de ces résultats numériques une visualisation graphique (2D et/ou 3D) de l’état du réseau au cours de la simulation (figure 1.1).

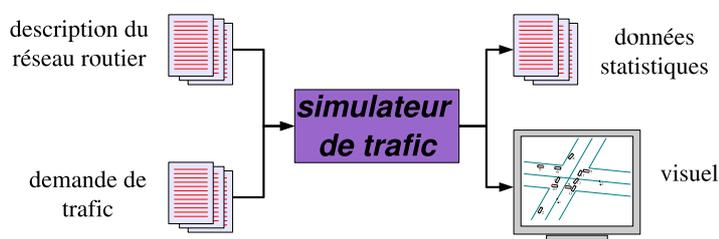


FIGURE 1.1 – Principe de fonctionnement d’un simulateur de trafic

1.1.2.1 Niveau de détails et granularité

Tous les outils de simulation n’offrent pas le même niveau de détails, aussi est-il courant de distinguer trois niveaux de granularité : microscopique, mésoscopique et macroscopique [Cohen, 1993].

Une simulation dite microscopique offre un niveau de détails plus grand qu’une simulation macroscopique (figure 1.2).

Les simulations de type macroscopique s’intéressent aux flux de trafic qu’elles cherchent à caractériser sous la forme de grandeurs physiques comme par exemple le débit et la densité. Ces simulations permettent de décrire l’écoulement du trafic d’un point de vue global.

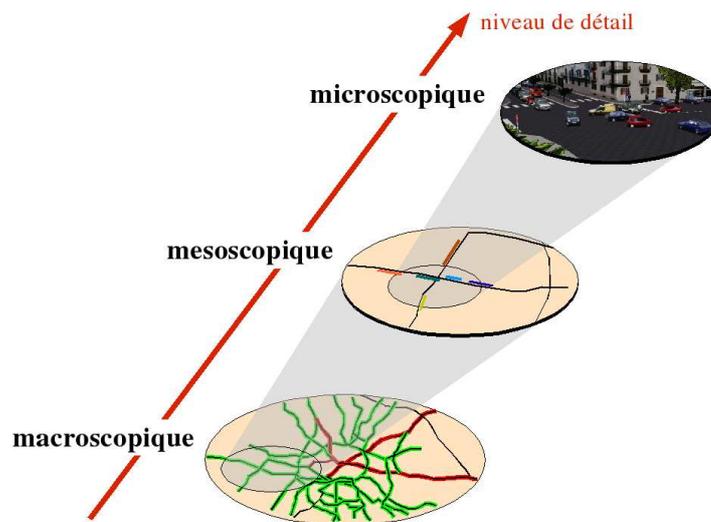


FIGURE 1.2 – Granularité et niveau de détails des simulations de trafic

A l'inverse, les simulations microscopiques considèrent de manière individuelle les différentes entités du flux de trafic. Il en résulte un niveau de détails beaucoup plus élevé permettant de rendre compte de manière relativement fine de l'activité et des interactions des mobiles simulés.

Les simulations mésoscopiques sont un compromis entre le niveau macroscopique et microscopique. Les véhicules ne sont plus considérés de manière individuelle mais sont regroupés selon des caractéristiques de "trafic" proches et homogènes. Un peloton de véhicules sur autoroute sera par exemple simulé comme une seule et unique entité.

1.1.2.2 Gestion du temps

La gestion du temps constitue un second critère de classification pour les outils de simulation de trafic puisque les phénomènes de trafic s'entendent dans le temps. Une représentation continue du temps est difficilement envisageable sur ordinateur ; c'est pourquoi tous les outils de simulation de trafic utilisent une discrétisation du temps.

Certains modèles de simulation sont dits à événement discret. Le calcul des états et des valeurs des différentes variables de la simulation (comme par exemple le calcul de l'accélération d'un véhicule donné) est régi par l'apparition de certains événements déclencheurs d'un changement d'état (exemple : arrivée à un croisement).

Les autres modèles de simulation utilisent une discrétisation homogène du temps sous la forme d'intervalles de longueur égale appelés "pas de temps" ou encore "pas de simulation". A chacun de ces pas, les différents états et variables de la simulation sont recalculés. Ces modèles ont l'avantage d'offrir une représentation fine tout en conservant une certaine simplicité en terme d'implémentation. En revanche, lorsque l'environnement évolue très lentement, d'un pas de simulation à l'autre, de nombreux calculs sont effectués inutilement.

1.1.2.3 Simulation stochastique ou déterministe

L'utilisation de variables aléatoires dans les modèles de simulation permet d'introduire une certaine dispersion autour des phénomènes simulés. De tels modèles sont dits stochastiques. A l'inverse, les simulations déterministes n'ont recours à aucune variable aléatoire, ni fonction probabiliste : tous les calculs sont exacts, prédictibles et reproductibles à l'infini.

Il est rare de pouvoir affirmer qu'un modèle de simulation est purement stochastique ou purement déterministe. Certaines parties du modèle requièrent l'utilisation de variables aléatoires et probabilistes, comme par exemple la répartition de la taille des véhicules autour d'une valeur moyenne dans un peloton. A l'inverse, dans d'autres parties du modèle, il est plus simple (notamment lors de la conception) de n'utiliser que des processus entièrement déterministes.

De nombreux autres critères de classification peuvent être considérés comme par exemple la flexibilité et la généralité du modèle. En effet, dans le domaine de la simulation de trafic routier, de nombreux outils ont été développés pour des besoins bien spécifiques : simulation d'axe urbain, simulation autoroutière, simulation de trajectoire en ronds-points, etc. Cette multiplication des outils montre qu'il est extrêmement difficile de concevoir un modèle générique et facilement extensible. La généralité d'un modèle est à mettre directement en relation avec l'approche utilisée pour concevoir les outils de simulation.

1.1.3 Conception des modèles de simulation

1.1.3.1 Approches mathématiques versus comportementales

Historiquement, les premières études du trafic routier sont dues aux physiciens qui considéraient que l'écoulement de véhicules était assimilable à celui d'un fluide et le réseau routier un ensemble de canalisations. Les premières approches en simulation étaient donc fondées sur des modèles mathématiques.

Celles-ci consistent à mesurer par le biais de capteurs (caméras, boucles magnétiques, etc) un ensemble d'indicateurs et de variables (débit, densité, vitesse moyenne, temps inter-véhiculaire, etc) décrivant la situation observée. Toutes les données ainsi recueillies sont ensuite mises en équation de manière à obtenir une description analytique de la situation mesurée. Suivant la granularité de la simulation, les équations mathématiques sont des lois de circulation de flux (simulation macroscopique) ou des lois de poursuites et de changements de voies (simulation microscopique).

Les principales critiques qu'il est possible d'émettre sur ces approches concernent leur manque de généralité et d'interactivité. Les lois obtenues sont généralement liées aux caractéristiques physiques du tronçon utilisé pour les mesures (longueur, nombre de voies, marquage au sol) faisant totalement abstraction de l'environnement. De plus, une équation est par définition relativement figée et ne peut s'adapter facilement à l'introduction de nouveaux éléments dans la simulation.

Depuis une quinzaine d'années, une approche dite "comportementale" ou "individu-centrée" tente de combler les faiblesses de l'approche mathématique. Le postulat sur lequel repose cette approche est le suivant : le trafic est le résultat de la somme de toutes les actions et interactions des différents éléments de la simulation : usagers (automobilistes, piétons, deux roues), signalisations, infrastructures routières, etc. L'approche comportementale s'attache donc à modéliser le plus finement possible les comportements individuels des différentes entités simulées. Celles-ci s'adaptent en permanence à

la situation routière en prenant en compte l'environnement dans lequel elles évoluent. Les outils de simulation issus de cette approche de conception sont généralement microscopiques.

Les phénomènes de trafic simulés de cette manière sont qualifiés d'émergents et résultent à la fois de l'hétérogénéité des entités simulées (variété des véhicules par exemple) et de l'hétérogénéité des comportements simulés (variété des pratiques individuelles).

Pour illustrer la différence entre l'approche mathématique et l'approche comportementale, considérons la situation de trafic suivante : un peloton de véhicules s'arrête puis redémarre. Le phénomène de trafic observé est un effet "accordéon" résultant de la contraction puis du relâchement du peloton (figure 1.3).

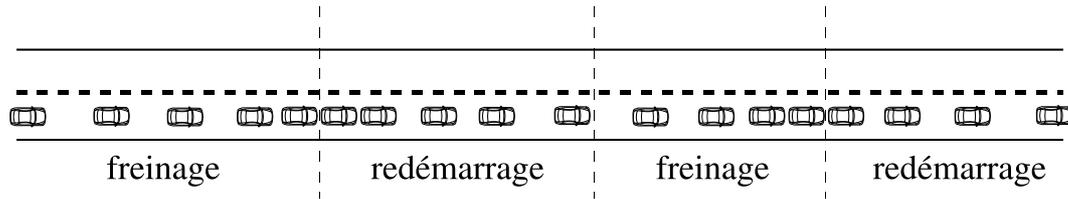


FIGURE 1.3 – Effet accordéon

Dans les approches mathématiques, un tel phénomène est reproduit à l'aide de lois de poursuite. Un véhicule d'indice n est repéré à un instant t par : sa position sur le réseau $x_n(t)$, sa vitesse $x'_n(t) = \frac{dx_n(t)}{dt}$ et son accélération $x''_n(t) = \frac{dx'_n(t)}{dt}$. Une loi de poursuite exprime l'accélération du véhicule n à l'instant $t + T$ en fonction de la vitesse instantanée de n , de la vitesse relative entre n et $n - 1$ et de la distance inter-véhiculaire. La forme générale d'une loi de poursuite est : $x''_n(t + T) = f(x'_n(t), x'_{n-1}(t) - x'_n(t), x_n(t) - x_{n-1}(t))$. Dans la littérature, on rencontre généralement deux types d'expression de lois de poursuites. Le premier exprime l'accélération du véhicule n comme une pondération de deux variables cinématiques [Leglise, 1971] :

$$x''_n(t + T) = \lambda (x'_{n-1}(t) - x'_n(t)) + \mu (x_{n-1}(t) - x_n(t))$$

Le second type de loi n'est pas une forme linéaire et exprime l'accélération x''_n en fonction la vitesse relative de n et de l'espace qui le sépare de $n - 1$ [Gazis, 1974] :

$$x''_n(t + T) = \frac{f(x'_n(t + T))}{g(x_{n-1}(t) - x_n(t))} (x'_{n-1}(t) - x'_n(t))$$

Dans ces équations, l'élasticité du peloton est obtenue à l'aide de différents coefficients (λ , μ , f et g dans les deux exemples ci-dessus) représentant une sensibilité du conducteur n aux différentes variables cinématiques du conducteur $n - 1$.

Dans une simulation comportementale, cet effet d'élasticité du peloton de véhicules n'est pas explicitement décrit. La contraction et le relâchement du peloton sont obtenus par émergence suite à la prise de décision de chaque conducteur simulé qui tente de maintenir une distance de sécurité avec les autres en fonction des caractéristiques physiques de son véhicule. La complexité d'une situation de trafic simulée ne dépend pas de la complexité d'une équation la décrivant mais résulte des multiples interactions existant entre le véhicule simulé et son environnement.

1.1.3.2 Origines de l'approche comportementale

L'approche comportementale que nous venons d'opposer à l'approche mathématique n'est pas exclusive à la simulation de trafic. Son origine est assez diffuse puisqu'elle est apparue conjointement dans d'autres domaines de recherche : écologie, vie artificielle, robotique, intelligence artificielle distribuée.

A la fin des années quatre-vingt, l'écologie s'intéresse à la modélisation de l'individu. Dans [Hutson *et al.*, 1988], les auteurs justifient la nécessité de prendre en compte l'individu dans les modèles d'éthologie. Dans [Hogeweg et Hesper, 1990], les auteurs établissent les bases de la vie artificielle. Aujourd'hui, la simulation multi-agent est largement utilisée dans ce domaine : [Ramat *et al.*, 1998], [Bousquet *et al.*, 1998].

Plus près de notre problématique, la robotique a contribué à la mise en évidence des faiblesses des simulations fondées sur une approche mathématique. Différents projets en robotique visant à mettre en place des entités automatisées se déplaçant de manière autonome (exemple : robots d'exploration) ont dû avoir recours à la simulation tant au niveau de la mise au point des algorithmes de déplacements que de leur validation ([Reece et Shafer, 1993],[Suksthanekar *et al.*, 1996]).

En ce qui concerne la simulation de trafic, les projets de routes automatisées et de véhicules intelligents ont poussé à repenser la manière de concevoir les outils de simulation de trafic. De manière à pouvoir tester un véhicule autonome en simulation, il devient nécessaire que l'ensemble du trafic interagisse avec lui, ce que ne pouvaient offrir les modèles mathématiques. La prise en compte de l'individu a été également suggérée par l'apparition des simulateurs de conduite dont le but est d'immerger un être humain au sein de situations de conduite virtuelles. Enfin, les besoins sans cesse croissants en terme de réalisme ont conduit à introduire autour du conducteur immergé un trafic ambiant réaliste [Bonakdarian *et al.*, 1998].

1.1.4 Modélisation de la tâche de conduite

L'approche comportementale en simulation de trafic revient à essayer de reproduire de manière individuelle la tâche de conduite de chaque conducteur simulé. La modélisation de la tâche de conduite est donc le cœur du problème dans les approches comportementales. Plus la tâche de conduite est modélisée fidèlement par rapport à la réalité, plus les interactions entre les entités simulées sont justes et plus les phénomènes de trafic émergents sont réalistes.

Les premières études et tentatives de modélisation de la tâche de conduite datent des années 70. Aujourd'hui, la prise en compte des facteurs humains (influence du risque, du lieu de vie, etc) fait partie intégrante du corpus d'études sur la théorie du trafic [Koppa, 1997]. Une modélisation hiérarchique basée sur l'identification de trois niveaux de prise de décision fait généralement référence dans la littérature [Michon, 1985] :

- le niveau stratégique vise à planifier le déplacement de l'individu : itinéraire à suivre et temps de trajet estimé constituent par exemple un choix de buts de haut niveau.
- le niveau tactique concerne la prise de décision relative aux changements de trajectoire, aux traversées d'intersection, etc. Il ne s'agit plus ici de buts de haut niveau mais plutôt d'objectifs à court terme.

- le niveau opérationnel est celui des tâches de bas niveau de la conduite comme l'utilisation du volant et des pédales. Il correspond à l'application effective des décisions prises dans les niveaux supérieurs.

Plusieurs implémentations de modèles de trafic fondés sur l'approche comportementale ont été proposées dans la littérature et seront détaillées dans le chapitre 2.

De par sa philosophie privilégiant la prise en compte de l'individu, l'approche comportementale est à rapprocher d'une branche de l'intelligence artificielle distribuée : les systèmes multi-agents dont le but est l'étude des interactions entre entités informatiques autonomes.

1.2 Approche multi-agent et simulation

Les concepts d'agent et de système multi-agent peuvent différer suivant le contexte applicatif. De nombreuses définitions ont été introduites dans la littérature, chacune exprimant une vision propre à l'auteur. La définition la plus générale qu'il est possible de donner d'un agent est : "*entité virtuelle ou physique agissant dans un environnement*".

Dans cette seconde section, nous introduisons les éléments théoriques qui nous permettront de présenter et de situer le modèle de trafic ARCHISIM dans un cadre multi-agent en dernière partie du chapitre.

1.2.1 Les systèmes multi-agents : motivations, concepts et propriétés

1.2.1.1 Historique et motivations

Bien que le terme soit apparu plus tard, il est possible de resituer la notion d'agent dès le début de l'intelligence artificielle. En effet, cette branche de l'informatique apparue au milieu des années 50, visait la modélisation informatique de capacités de raisonnements. Autrement dit, elle avait pour objectif de construire une entité informatique dotée d'une certaine intelligence : un agent rationnel.

A la fin des années 70, la communauté scientifique s'est intéressée à la résolution distribuée de problèmes en essayant de tirer partie des recherches sur la parallélisation des algorithmes dans l'optique de réduire les temps de calculs. Les travaux se sont principalement focalisés sur la manière de diviser un problème donné en sous-problèmes, chacun alloué à un résolveur et sur la synthèse des résultats permettant d'atteindre la solution globale du problème initial. Cette coordination d'agents rationnels marque les débuts des systèmes multi-agents.

Le concept de système multi-agent (SMA) tire sa justification du fait que les capacités cognitives d'un être humain résultent :

- d'un point de vue social, des interactions entretenues avec les autres (rôle de la discussion coopérative dans le développement de l'intelligence [Piaget, 1999])
- d'un point de vue neurologique, des multiples interactions des différentes zones du cerveau, chacune étant dotée de capacité bien particulière comme le traitement de la parole ou de la vision (cette idée d'intelligence résultant d'un "collectif" est décrite par Minsky dans "La société de l'esprit" [Minsky, 1988])

1.2.1.2 Définitions et propriétés

Pour définir de manière un peu plus formelle un système multi-agent, considérons l'approche VOYELLES de Demazeau [Demazeau, 1995] qui modélise un SMA autour de quatre dimensions : l'Agent, l'Environnement, l'Interaction et l'Organisation. Un système multi-agent peut se définir comme un ensemble A d'entités autonomes plongées dans un environnement E et interagissant entre elles dans le cadre d'une organisation O structurant l'ensemble A et définissant les fonctions de chacune.

La littérature associe généralement aux SMA plusieurs caractéristiques. Dans [Ferber, 1995], l'auteur précise qu'un agent dispose en général d'un ensemble de buts à atteindre en fonction de caractéristiques, de compétences et de ressources qui lui sont propres. L'autonomie peut être complétée par la flexibilité [Jennings, 1998]. Les agents doivent ainsi être capables d'agir dans un temps imparti. Leur comportement doit être proactif et favoriser l'interaction avec leurs congénères de manière à accomplir leurs tâches.

L'absence de contrôle global fait partie des principales propriétés d'un SMA : la prise de décision est décentralisée en chaque agent. Une seconde propriété forte des systèmes multi-agents concerne la notion d'émergence. Elle caractérise les phénomènes qui apparaissent de manière globale au sein d'un système multi-agent à partir des interactions locales entre les agents. En ce sens, l'émergence peut donc être vue comme le passage du niveau microscopique au niveau macroscopique du système (figure 1.4). Ronald et Müller définissent l'émergence dans les systèmes multi-agents comme l'identification par un observateur de régularités associées à un processus qui ne peuvent être prédites à partir de la seule connaissance des propriétés locales des agents ([Ronald *et al.*, 1999],[Müller, 2004]). Le

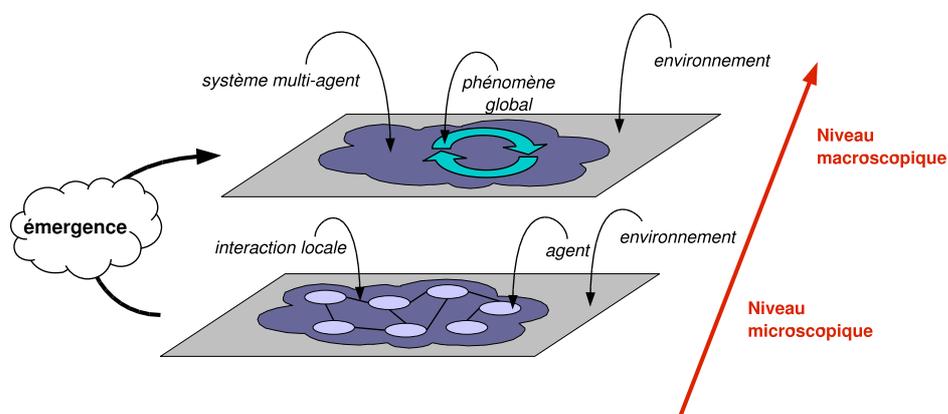


FIGURE 1.4 – Émergence dans les systèmes multi-agents

modèle de Schelling [Schelling, 1978], permettant de modéliser des phénomènes de migration et de répartition de populations sur un territoire, est un exemple parfait de ce concept d'émergence. Des agents positionnés sur un damier interagissent de manière locale avec leur voisinage composé de huit cases. Chaque agent se déplace de manière à avoir au moins 37,5% de voisins semblables. Après quelques itérations, le modèle atteint un état stable en faisant apparaître des configurations globales de ségrégation résidentielle.

Ayant défini les bases et principes d'un système multi-agent, nous allons maintenant détailler individuellement les différents éléments composant un SMA.

1.2.2 Éléments d'un système multi-agent

1.2.2.1 Agent

La prise de décision d'un agent se décompose suivant un cycle de 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. Cette prise d'informations peut s'accompagner pour certains agents de la construction et de la mise à jour d'une représentation symbolique de l'environnement. La phase de cognition correspond au raisonnement de l'agent lui permettant de planifier les actions qu'il devra réaliser. Enfin, la phase d'action est la réalisation effective des décisions préalablement établies.

Le terme *raisonnement* utilisé pour expliquer la phase de cognition est à prendre au sens large. Des agents dits cognitifs peuvent réaliser des raisonnements complexes réutilisant les travaux menés en intelligence artificielle classique. Ils peuvent disposer d'une représentation explicite et symbolique de l'environnement leur permettant de déterminer les actions à exécuter.

A l'inverse, des agents réactifs disposent d'une architecture dans laquelle il existe un couplage direct entre perception et action. Leur phase de cognition est le plus souvent réduite à un déclenchement de règles comportementales simples associant à un stimulus particulier une action spécifique. Les travaux de Brooks ([Brooks, 1986], [Brooks, 1991]) sont une parfaite illustration du concept d'agents réactifs. Selon lui, le comportement intelligent d'un système multi-agent ne requiert pas forcément des raisonnements individuels complexes mais peut résulter de la multitude des interactions simples entre des agents aux capacités limitées.

La frontière entre agents réactifs et cognitifs n'est pas aussi évidente et certaines architectures d'agent tirent parti des deux approches ; on parle alors d'agents hybrides. Il peuvent par exemple reposer sur des mécanismes d'adaptation leur permettant suivant le contexte de passer d'un mode cognitif à un mode réactif et inversement.

1.2.2.2 Environnement

L'environnement joue un rôle essentiel dans la modélisation d'un système multi-agent puisqu'il fournit un espace commun à tous les agents leur permettant d'interagir [Briot et Demazeau, 2001]. La composition de l'environnement diffère suivant la nature de l'application considérée. Par exemple, dans le cadre de la résolution distribuée de problèmes, l'environnement est essentiellement composé des données du problème que les agents pourront modifier ou utiliser suivant le type de couplage adopté entre les agents et l'environnement. Dans les applications de type simulation, l'environnement se définit comme un espace dans lequel est placé un ensemble d'objets pouvant être perçus, manipulés, modifiés par les agents du système.

Plusieurs propriétés peuvent caractériser l'environnement [Russell et Norvig, 1995]. 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*. Dans le cas contraire, on parle d'environnement discret.

Un environnement est dit *dynamique* lorsque son état dépend à la fois 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.

Un environnement est *déterministe* si son état futur est complètement calculable à partir de son état courant et des effets induits par les actions des agents. Si les effets des actions sur l'environnement ne sont pas prédictibles, celui-ci est qualifié de *non-déterministe*.

L'accessibilité constitue également une propriété forte de l'environnement. Elle caractérise le niveau de perception pouvant être atteint par les agents. Lorsqu'ils peuvent obtenir une information parfaite et complète de l'environnement, ce dernier est dit *accessible*.

En résumé, l'environnement est donc :

- la donnée d'entrée de tout processus de raisonnement d'un agent (phase de perception)
- le siège de l'activité des agents permettant la réalisation des actions au niveau individuel et facilitant l'interaction au niveau collectif (phase d'action)

1.2.2.3 Interaction

Une interaction est une relation dynamique instaurée entre plusieurs agents du fait de leurs actions combinées et réciproques. L'interaction au sein d'un SMA peut prendre différentes formes : la coordination, la négociation, la coopération (celles-ci seront détaillées dans le chapitre 2). Une interaction peut avoir lieu avec [Durfee *et al.*, 1987] ou sans communication ([Rosenschein *et al.*, 1986],[Kok *et al.*, 2005]).

La communication dans un système multi-agent peut utiliser différents supports. Historiquement, les communications des premiers SMA utilisaient des structures de données partagées (tableau noir) auxquelles les agents accédaient tour à tour pour y laisser les informations qu'ils souhaitaient diffuser ([Erman *et al.*, 1980],[Corkill, 1991]).

Le modèle de communication actuel dans les SMA est direct et repose sur l'envoi de messages reprenant ainsi les différents modes de communications existants dans les systèmes informatiques : communication point à point, "broadcast", "forecast". De plus, l'introduction de standards (langage et protocoles d'interaction) par les organismes tels que la FIPA (Foundation for Intelligent Physical Agents) a ouvert la porte à une plus grande interopérabilité entre plate-formes multi-agents.

La communication au sein d'un SMA peut également être indirecte et utiliser l'environnement comme support de communication : un agent laisse dans l'environnement une trace qui pourra être perçue par d'autres agents. Ce mode de communication s'applique parfaitement aux agents de type réactif utilisés par exemple en simulation d'animats [Drogoul, 1993].

Les interactions fournissent le moyen de construire des comportements d'agents [Mathieu et Picault, 2006]. Elle peuvent également contribuer à l'apparition d'organisations au sein des SMA.

1.2.2.4 Organisation

Cette quatrième facette d'un SMA est l'une des plus importantes puisqu'elle conditionne l'efficacité globale du système ([Mandiau *et al.*, 2000], [Mandiau, 2000]). L'organisation se définit comme un

agencement de relations entre les agents produisant une unité du système [Ferber, 1995]. Elle fournit un schéma de prise de décision à l'ensemble des agents devant satisfaire leurs buts tout en assurant un comportement global cohérent [Malone, 1987].

L'organisation d'un SMA peut être statique et définie lors de sa création. Les agents peuvent également faire preuve d'adaptation et revoir l'organisation du système en cours d'exécution. Certains systèmes font naître une organisation sans qu'aucune n'ait été explicitement prédéfinie. Ces systèmes se rapprochent des sociétés animales : chaque agent agit sans se préoccuper des autres : l'organisation est émergente et ne sert pas de support à l'activité.

La vision d'une organisation dans un SMA est à plusieurs niveaux. Ainsi une organisation portant sur un petit groupe d'agents peut elle même faire partie d'une organisation plus globale portant sur l'ensemble du système (niveau social).

1.2.3 Simulation multi-agent

Dans cette dernière sous-section, nous nous intéressons à l'un des axes de recherche dans les systèmes multi-agents : la simulation multi-agent.

1.2.3.1 Principes de l'approche

La simulation multi-agent consiste à utiliser un ensemble d'agents en interaction de manière à reproduire la dynamique et l'évolution globale du système que l'on cherche à simuler. Ce type de simulation fait donc l'hypothèse que des processus informatiques peuvent décrire et reproduire le comportement d'entités observées dans la réalité.

La simulation multi-agent est particulièrement adaptée à la reproduction de phénomènes naturellement localisés et distribués [Van Dyke Parunak *et al.*, 1998] fondés sur le collectif. La méthodologie couramment utilisée est une approche descendante consistant à identifier d'une part les entités composant le phénomène réel et d'autre part, les relations entre ces entités. Chaque entité est ensuite modélisée sous la forme d'un agent, généralement en faisant quelques hypothèses simplificatrices du fait de la complexité des phénomènes à simuler. La modélisation des entités et de leurs relations peut être améliorée et enrichie par des évaluations successives et des comparaisons avec les observations du phénomène réel ([Drogoul, 1993],[Espié *et al.*, 2002]).

L'approche multi-agent est une voie prometteuse en simulation. Elle apporte une alternative crédible aux simulations classiques fondées sur une modélisation analytique des phénomènes. Les recherches actuelles dans ce domaine s'attachent à proposer des plates-formes et "frameworks" facilitant la mise au point de telles simulations ([Bonabeau *et al.*, 1999], [Ferber et Michel, 2000]). Des travaux récents s'intéressent également aux problématiques de changement d'échelle : identification de fonctions mathématiques macroscopiques à partir de simulation, couplage entre modèles centré-individu et modèles mathématiques [Duboz *et al.*, 2003].

1.2.3.2 Applications

La simulation multi-agent est utilisée depuis le début des années 90 dans des divers domaines. L'approche a été largement exploitée pour la simulation du "vivant" : éthologie, sociétés animales et

écosystèmes ([Drogoul, 1993], [Ramat *et al.*, 1998], [Bousquet *et al.*, 1998]). Ces travaux tirent parti de différents concepts précédemment décrits comme les agents réactifs, les communications via l'environnement et l'émergence d'organisation. Dans ces domaines d'application, la simulation multi-agent fournit aux chercheurs du vivant (éthologues, biologistes, etc) un outil d'investigation et d'expérimentation. Certaines simulations permettent l'utilisation de protocoles d'observation identiques à ceux utilisés en réel, offrant ainsi un laboratoire virtuel d'études.

Plus proches des phénomènes rencontrés dans les sociétés humaines, nous pouvons citer des travaux en évolution du langage humain [Kaplan, 1998] ou encore en migration inter-urbaines [Vanbergue et Drogoul, 2002]. Ces types de simulation se veulent être des modèles explicatifs du phénomène simulé : ils permettent de comprendre les mécanismes sous-jacents au phénomène global observé.

Une dernière catégorie d'applications concerne la simulation virtuelle de comportements humains. Le cinéma et les jeux vidéos sont de plus en plus demandeurs d'outils permettant de reproduire l'animation comportementale d'un grand nombre d'avatars dans le but de donner plus de réalisme à une situation de jeu ou une scène de film. Inspiré de l'approche des "boids" de Reynolds [Reynolds, 1987], de nombreux outils commerciaux répondant à cette demande ont vu le jour : logiciel *Massive* de Massive Software, *AI implant* d'Engenuity Technologie, etc.

La problématique de simulation multi-agent en trafic routier que nous exposons dans la section suivante au travers de la présentation du modèle ARCHISIM s'inscrit dans les trois catégories précédemment évoquées. En effet, il s'agit de simuler finement le système de trafic en s'attachant à reproduire le comportement des conducteurs réels. Comme dans les jeux vidéo, la difficulté consiste à conserver dans le temps un comportement cohérent du point de vue de l'humain immergé. L'analogie s'arrête pourtant là puisque contrairement à ce que l'on peut rencontrer dans les jeux de type simulation automobile, le trafic ambiant doit être valide d'un point de vue statistique. L'objectif est également de fournir un outil de recherche permettant : d'étudier le comportement du conducteur, d'évaluer l'impact des dispositifs d'aide à la conduite, d'analyser les phénomènes de trafic en isolant leurs causes et leurs effets.

1.3 Le modèle de trafic ARCHISIM

Ayant dressé un état de l'art des simulations de trafic en section 1.1 et après avoir défini les principaux concepts relatifs aux systèmes multi-agents, nous allons nous attacher à positionner et décrire le modèle de simulation de trafic ARCHISIM.

1.3.1 Positionnement du modèle

L'idée centrale du projet ARCHISIM est d'utiliser les résultats des travaux en psychologie de la conduite pour simuler un trafic complexe et réaliste. Les premières réflexions sur ARCHISIM datent de 1988, le projet se concrétise à partir de 1992. L'architecture d'ARCHISIM permettra par la suite de coupler un simulateur de conduite avec une simulation de trafic [Espié, 1995]. La motivation d'une telle approche est de pouvoir étudier le comportement d'un être humain plongé dans un trafic visuellement et statistiquement réaliste. L'approche mathématique, pour les raisons décrites en section 1.1.3, ne peut être utilisée au sein d'une telle architecture. ARCHISIM repose donc sur une approche comportementale [Espié et Saad, 2000] : le modèle de trafic doit être générique afin de pouvoir adapter

les conducteurs simulés aux différents types de situation de conduite. ARCHISIM se positionne donc comme une simulation microscopique, déterministe et à temps discret.

Le modèle de trafic ARCHISIM adopte une approche fondée sur des études réalisées en psychologie de la conduite. De telles études permettent de définir des modèles conceptuels de comportements des conducteurs. La méthodologie utilisée consiste en des expérimentations sur route pour lesquelles des enregistrements vidéos et sonores viennent en complément de mesures réalisées sur le véhicule : vitesse, accélération, pression exercée sur les pédales, mouvements du volant, etc. Un entretien post-expérimentation vient, en général, compléter l'étude en verbalisant les commentaires du sujet face aux comportements qu'il a adoptés durant l'étude [Saad, 1992].

Les données ainsi recueillies sont ensuite regroupées en situations de conduite de manière à pouvoir isoler et identifier les comportements significatifs en rapport avec la situation. Une analyse fine permet ensuite d'explicitier la prise de décision du conducteur en mettant en évidence les motifs qui la sous-tendent.

Les modèles de psychologie du conducteur expriment différentes facettes de son comportement, en particulier :

- la minimisation du risque et la prise en compte de l'aspect sécuritaire [Summala, 2005],
- le maintien des préférences : vitesse désirée, position sur la chaussée, etc,
- la minimisation des efforts physiques et intellectuels sous-jacents à la tâche de conduite,
- et la minimisation des interactions avec l'environnement.

Le modèle du comportement du conducteur utilisé dans ARCHISIM est issue d'une collaboration entre psychologues et informaticiens de l'INRETS dans le but de traduire les résultats obtenus sous la forme de règles génériques et implémentables.

1.3.2 Architecture informatique

1.3.2.1 Vue globale

A l'origine, l'architecture d'ARCHISIM repose sur une distribution de la simulation en un ensemble de processus : chaque agent est considérée comme une tâche et un moniteur temps réel alloue un temps de calcul à chaque agent. ARCHISIM a été conçue de manière à pouvoir "découper" et simuler un réseau routier sur un ensemble de machines reliées entre elles par un réseau de type Ethernet. Une machine héberge toutes les données relatives à l'environnement simulé comme la description du réseau (noeuds d'intersections et liste des routes), des tronçons (type, signalisation, etc) et des mobiles y évoluant. Elle joue ainsi un rôle de serveur de vision : chaque agent qui s'exécute sur une autre machine peut se connecter au serveur de manière à recevoir à chaque pas de simulation les informations lui décrivant son environnement immédiat et lointain (figure 1.5). Chaque agent peut recevoir au maximum une liste de 100 vues (une vue étant un agent ou un élément de l'infrastructure routière) portant sur son environnement local. Ce processus n'intervient pas dans la prise de décision individuelle de chaque agent : il s'agit plus d'un artifice de programmation permettant de simplifier l'interaction environnement/agent. Le fonctionnement d'ARCHISIM est synchrone : sur un pas de temps, le moteur de simulation donne successivement la main à chacun des agents pour leur permettre de récupérer les informations concernant leur environnement, prendre leur décision et mettre à jour leurs données locales vers le serveur.

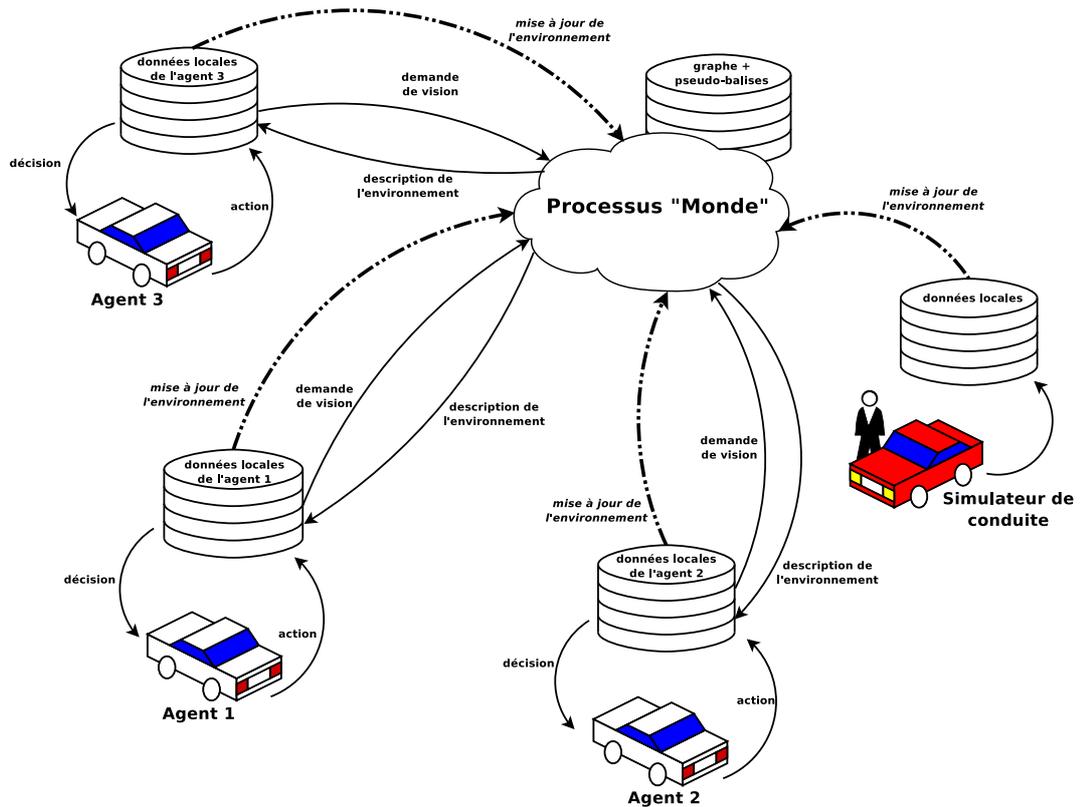


FIGURE 1.5 – Architecture globale d'ARCHISIM

Lorsque la simulation est distribuée sur plusieurs machines, les communications s'effectuent en mode connecté (TCP)[Tanenbaum, 2000] via un mécanisme de socket. Le serveur de vision et les agents peuvent être exécutés sur une seule et même machine lorsque sa puissance de calcul le permet. Les communications s'effectuent dans ce cas sur le principe de mémoire partagée.

Cette architecture ouverte et distribuée offre la possibilité d'accueillir un simulateur de conduite composé de différents modules : une boucle visuelle 3D, un modèle dynamique de véhicule, etc (figure 1.6). L'être humain placé dans la cabine est considéré comme un agent par le système au même titre que les conducteurs simulés du trafic.

1.3.2.2 Conducteur simulé

Chaque acteur mobile du trafic (automobiliste, motard, piéton) est modélisé sous la forme d'un agent disposant :

- de buts à satisfaire : vitesse désirée à maintenir, itinéraire à suivre, ...
- de capacités qui lui sont propres : caractéristiques physiques de son véhicule (capacité de freinage, d'accélération, taille et empattement sur la chaussée), d'un niveau d'expérience de la conduite, style de conduite, niveau de respect du code de la route, ...
- d'une certaine autonomie : l'agent, pour satisfaire ses buts, est libre de modifier et d'adapter sa trajectoire au cours du temps tirant ainsi pleinement parti de l'espace "roulable" disponible. Il

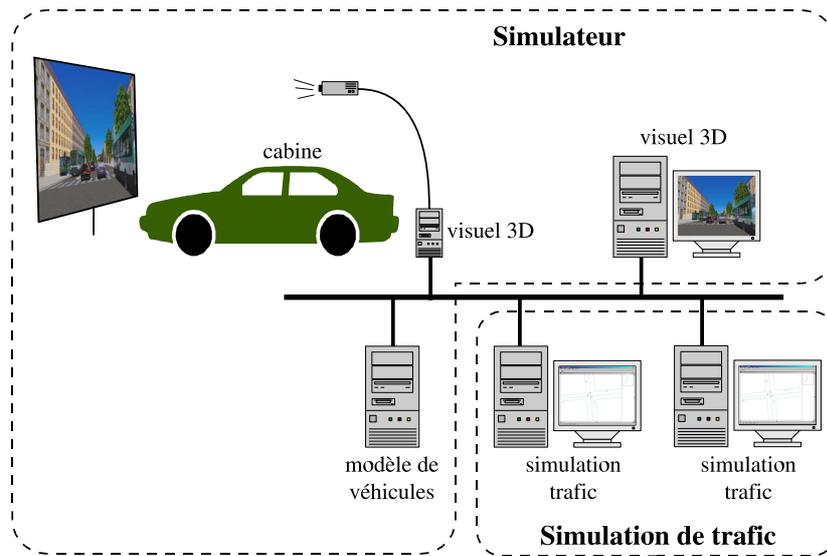


FIGURE 1.6 – Couplage entre simulation de trafic et simulateur de conduite

peut également transgresser les règles du code de la route en fonction du contexte, ce qui dénote d'une certaine autonomie face à la norme ([Dignum, 1997],[Carabelea et Boissier, 2005]).

La tâche de conduite étant par essence fortement cognitive, les agents utilisés dans ARCHISIM seront plutôt qualifiés de cognitif. Les algorithmes mis en jeu prennent en compte un certain nombre d'informations beaucoup plus riches que les simples stimuli utilisés dans les architectures de type réactif.

1.3.2.3 Environnement et perception

La description de l'environnement utilise une structure de graphe dans laquelle les routes sont assimilées aux arcs et les intersections aux noeuds. D'un point de vue topologique, chaque mobile est repéré sur le réseau par un numéro de route, un point kilométrique, une distance latérale par rapport à l'axiale (signée en fonction du sens de parcours) et un cap relatif à l'axe de la route (figure 1.7). Ce choix de repère pour l'environnement permet aux mobiles d'utiliser la totalité de l'espace "roulable" de la route et de changer de voie lorsqu'ils le désirent. Le modèle ARCHISIM s'affranchit ainsi des trajectoires pré-calculées qui sont des limites fréquemment rencontrées dans les autres outils de simulation.

Pour se déplacer sur ce réseau routier, un mobile a besoin de percevoir un certain nombre d'informations relatives à son environnement comme par exemple la présence de signalisation (feu tricolore, stop, etc), le marquage au sol ou la présence de véhicules devant lui. Comme nous l'avons précisé précédemment, la perception de l'environnement des agents repose sur l'utilisation d'un serveur de vision. Au début de son cycle de décision, un agent a la possibilité d'envoyer une requête de vision. Celle-ci est paramétrable : l'agent peut préciser ce qu'il souhaite 'voir' en définissant différents paramètres comme les distances de vision sur le côté, devant et derrière lui.

Les informations récupérées en retour décrivent de manière symbolique et numérique l'environnement du mobile. La vision d'un mobile est organisée par objets perçus. Ceux-ci peuvent être statiques (panneaux de signalisation, feux, marquage) ou mobiles (véhicules, piétons).

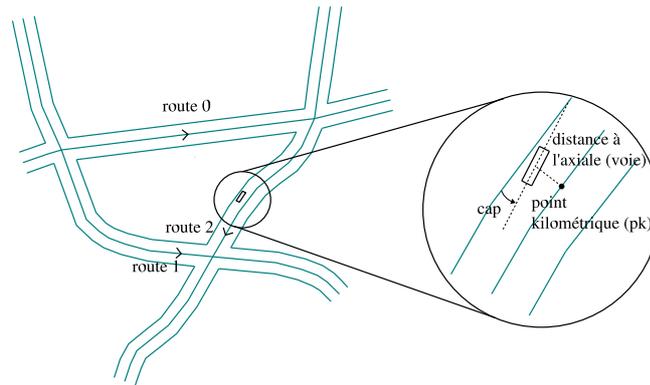


FIGURE 1.7 – Repérage dans l'environnement routier simulé

Pour chacun de ces objets, la vision précise le type, la catégorie (par exemple poids-lourd ou voiture particulière pour un objet de type véhicule), la position sur le réseau. Différentes informations décrivent en supplément l'état de chaque acteur de la simulation d'un point de vue cinématique (ex : accélération (longitudinale et latérale), vitesse instantanée).

Un ensemble de propriétés et de relations qualitatives ([Condotta, 2000], chapitre 3) entre objets fournissent une description spatiale de l'environnement. Un mobile connaîtra par exemple les véhicules situés devant lui, derrière lui, à sa droite et à sa gauche par le biais d'un ensemble de relations directionnelles [Ligozat, 1998]. Cet ensemble est complété par des relations d'orientation [Freska, 1992] propres aux situations de conduite. Celles-ci prennent en compte la position d'un mobile par rapport à l'infrastructure routière et sont définies du point de vue de l'agent voyeur : "*le véhicule X est en travers sur ma voie*", "*le véhicule Y est à côté sur la voie adjacente*", etc.

Un dernier type d'information, de nature déclarative, offre une description symbolique de l'état visuel d'un véhicule perçu : par exemple la présence de feux de détresse ou la mise en marche des clignotants. Ces informations permettent entre autre de connaître les intentions d'un conducteur concernant sa direction au carrefour ou son changement de voie.

L'environnement dans ARCHISIM peut être qualifié d'inaccessible (section 1.2.2). En effet, le processus de vision est limité, d'une part en distance mais également en nombre de véhicules pouvant être perçus. Un agent ne reçoit donc pas du serveur de vision une description complète de l'environnement. Par ailleurs, la vision étant intégrée au moteur de simulation, un agent ne peut recevoir d'informations sur l'environnement qu'une seule fois par pas de temps. L'environnement est donc discret. Enfin, l'environnement dans ARCHISIM est fortement dynamique du fait de l'intervention possible d'un ou plusieurs êtres humains par le biais du couplage avec le simulateur de conduite.

1.3.2.4 Interactions et prise de décision

Chaque mobile évolue dans l'environnement de manière autonome. L'espace "roulable" des axes du réseau apparaît donc comme une ressource commune à tous les agents de la simulation. En particulier, plusieurs mobiles peuvent convoiter, au même moment, un même espace de la route. Ceci crée un conflit de ressources. Les interactions entre les agents sont donc principalement conflictuelles.

Les algorithmes de décision utilisés dans ARCHISIM sont essentiellement des algorithmes de résolution de conflits (ceux-ci seront vus en détail au chapitre 2). Ils sont exécutés à chaque pas de temps et ne tiennent pas compte des connaissances passées. Cette hypothèse forte permet de s'abstenir de toute mémorisation (actions et états antérieurs de l'agent) qui d'une part peut se révéler lourde en terme de coûts calculatoires et d'autre part constitue un sujet de recherche à part entière qui n'a pas encore été étudié dans le cadre d'ARCHISIM.

Les interactions et les résolutions de conflits s'effectuent sans communication : dans ARCHISIM, les agents ne communiquent pas entre eux (de la même manière qu'il n'y a pas véritablement de communication entre conducteurs réels). Les seuls échanges de messages au sein du système s'effectuent entre le serveur de vision et les agents. Une forme de communication indirecte (section 1.2.2) peut être vue dans le fait que les conducteurs simulés adressent à l'ensemble du trafic des informations sur leur intention de tourner ou de freiner en actionnant leurs clignotants et feux de stop.

La tableau suivant résume l'approche multi-agent au sein d'ARCHISIM.

	Réalité	Simulation
Agent	Couple conducteur/véhicule	Agent logiciel autonome
	Vitesse désirée, itinéraire à suivre, ...	Buts de l'agent
	Expérience de conduite, capacité physique du véhicule, ...	Caractéristiques et compétences de l'agent
Environnement	Réseau routier	Ressource commune aux agents (graphe)
Interactions	Évitement des accidents, des blocages, ...	Résolution de conflits entre agents

Ayant présenté les grandes lignes d'ARCHISIM et positionné les modèles sous-jacents par rapport à l'état de l'art des deux premières sections, nous allons maintenant décrire le fonctionnement d'ARCHISIM à travers les différents logiciels qui le composent.

1.3.3 Chaîne de production de trafic sous ARCHISIM

L'équipe MSIS développe pour le projet ARCHISIM, tout un ensemble d'outils facilitant la mise en œuvre d'expérimentation nécessitant la production de trafic. Comme cela a été expliqué en section 1.1.1, toute simulation de trafic prend en entrée une description du réseau à simuler et une demande de trafic.

1.3.3.1 Création de réseaux

ARCHISIM dispose d'un outil d'aide à la création et à la saisie de réseaux : *wr2*. Il permet de générer automatiquement les différents fichiers décrivant les routes et intersections composant le réseau : les points kilométriques des axiales, la structure du réseau (graphe), l'habillage des routes (délimitations et trottoirs), le marquage au sol des différents tronçons, l'affectation des voies au trafic, les signalisations (panneaux statiques et dynamiques, contrôleurs de feux).

1.3.3.2 Génération de la demande de trafic

Une demande de trafic (fichier LOI) consiste en une liste de véhicules à générer durant la simulation en différents points du réseau. ARCHISIM fournit un logiciel de génération automatique de fichiers LOI appelé `distrib`. Le format de ces fichiers est très proche des fichiers de données obtenus lors de l'utilisation de capteurs sur les bords des routes, ce qui facilite les expérimentations et les études trafic à partir de données réelles. Pour chaque véhicule à créer, le fichier LOI indique :

- le pas de temps auquel il doit être créé,
- son emplacement sur le réseau à sa création,
- son accélération et sa vitesse initiales,
- son itinéraire à suivre sur le réseau,²
- les divers paramètres de comportements : expérience, souplesse de conduite, niveau de respect du Code de la Route, etc.

Le logiciel `distrib` permet de spécifier les caractéristiques de la demande de trafic. La répartition de la taille du gabarit des véhicules est obtenue en précisant le pourcentage de poids-lourds, de bus et de motos à générer parmi les automobiles. Pour chaque route, il est possible de spécifier par voie et par sens de circulation, la densité de trafic voulue (exprimée en nombre de véhicules par heure). Enfin, l'utilisateur peut définir la durée totale de la simulation.



FIGURE 1.8 – Logiciels dr2 et Sim²

2. dans le cas où il n'y en a pas, le système se charge d'affecter au véhicule une direction tirée aléatoirement à chaque intersection en fonction d'un pourcentage directionnel affecté au noeud correspondant

1.3.3.3 Simulation

Le coeur de la simulation à proprement parler, ainsi que le modèle de trafic précédemment décrit, sont intégrés dans le logiciel `dr2`. Il prend en entrée les fichiers de description du réseau créés par `wr2` et la demande de trafic générée par `distrib`. `dr2` se présente sous la forme d'une interface graphique offrant une vue en 2D du réseau routier utilisé pour la simulation (figure 1.8). Chaque mobile est symbolisé par un rectangle proportionnel aux dimensions du modèle de véhicule qui lui est associé. La simulation de conduite `Sim2` peut être exécutée en parallèle de `dr2` et les communications entre les deux logiciels permettent d'obtenir une visualisation 3D de la scène de trafic.

1.3.3.4 Recueil et traitement des données

L'utilisateur a la possibilité de placer sur le réseau un ensemble de capteurs permettant, à chaque passage d'un véhicule, d'enregistrer dans un fichier : le temps de passage, sa vitesse, son accélération, etc. Un outil de dépouillement de données (`dess`) permet de traiter les fichiers créés par ces capteurs virtuels de manière à agréger les données contenues suivant une période de temps définie par l'utilisateur. Cela facilite l'analyse des données et l'obtention d'indicateurs statistiques et courbes de trafic.

1.3.3.5 Vue d'ensemble

La figure 1.9 présente une vue d'ensemble des différents outils détaillés et de leur utilisation. Toute la chaîne de production de trafic y est résumée : de la saisie d'un réseau à l'obtention d'un visuel 3D de la scène de trafic simulée.

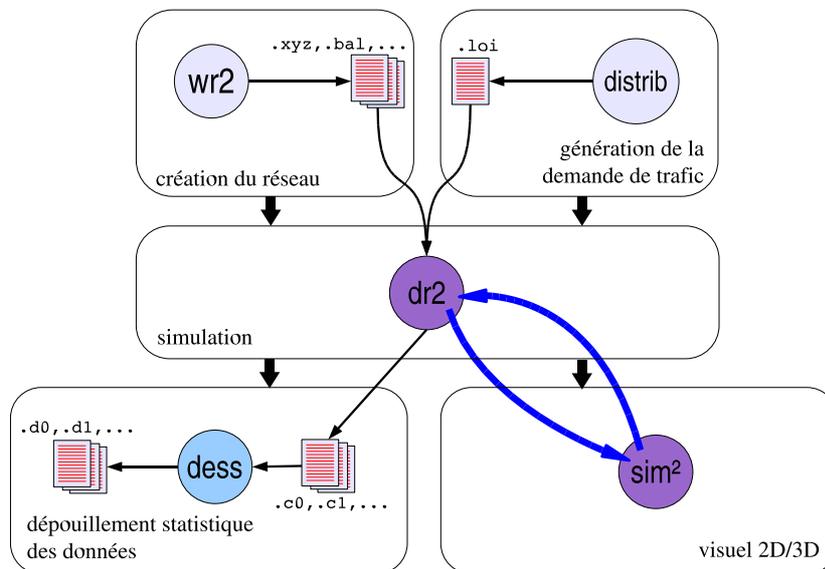


FIGURE 1.9 – Outils de simulation développés autour d'ARCHISIM

Conclusion

Dans ce chapitre, nous avons présenté en première partie le contexte applicatif de nos travaux de recherche : la simulation de trafic routier. Nous nous sommes en particulier attachés à montrer les lacunes que présentent les approches mathématiques couramment utilisées dans le domaine.

L'approche comportementale issue de recherches dans différents domaines comme la robotique, la biologie et les systèmes multi-agents apportent une alternative intéressante aux approches classiques basées sur les lois de poursuite. Les systèmes multi-agents, exposés en seconde partie de ce chapitre, fournissent quant à eux un cadre théorique à cette démarche de conception d'outils de simulation de trafic.

La dernière partie a été consacrée à la présentation du modèle de simulation de trafic ARCHISIM basé sur une approche comportementale et dont la mise en œuvre informatique est inspirée des principes multi-agents. L'architecture d'ARCHISIM a été positionnée par rapport aux concepts multi-agents introduits en deuxième partie et son principe de fonctionnement a été exposé au travers la description des différents outils développés par l'équipe MSIS de l'INRETS.

Le chapitre suivant se focalise sur la simulation de trafic en carrefour. Nous y formulons la problématique de nos travaux et dressons un état de l'art des différentes approches susceptibles d'y répondre.

Chapitre 2

Simulation de trafic : gestion des conflits et coordination multi-agent

Sommaire

2.1	Problématique de simulation de trafic en carrefour	28
2.1.1	Conduite en file / franchissement d'intersection	28
2.1.2	Les approches existantes	31
2.1.3	Synthèse	38
2.2	Coordination en environnement dynamique	38
2.2.1	Coordination : généralités	38
2.2.2	Mécanismes de coordination d'agents cognitifs	40
2.2.3	Mécanismes de coordination d'agents réactifs	42
2.2.4	Synthèse	44
2.3	Mécanisme de coordination pour la simulation de trafic en carrefour	45
2.3.1	Fondement théorique	45
2.3.2	Théorie des jeux et systèmes multi-agents	48
2.3.3	Modélisation	50
2.3.4	Résolution des conflits	53
2.3.5	Analyse et critique	56

Nous avons présenté au chapitre 1 la simulation comportementale de trafic ARCHISIM dans laquelle s'inscrivent nos travaux. Le modèle de trafic sous-jacent a fait l'objet de validations pour le cas de la conduite en file. Dans un premier temps, des évaluations basées sur différents scénarii tests ont permis de valider d'un point de vue local et visuel les comportements des conducteurs simulés pour des situations de conduite sur autoroute ([El Hadouaj *et al.*, 2000],[El Hadouaj, 2004]). Ces évaluations menées sur le plan individuel (niveau microscopique) ont été complétées par des validations sur le plan collectif (niveau macroscopique). Celles-ci ont consisté à simuler différentes tranches horaires (principalement en heures de pointe) sur une portion autoroutière au sud de Paris (autoroute A6) et à comparer les débits et vitesses moyennes obtenus en simulation avec les données mesurées en réel [Champion *et al.*, 2002].

Dans ce chapitre, nous nous focalisons sur la simulation de trafic en milieu urbain. La première partie est consacrée à la présentation de notre problématique. Nous y dressons un état de l'art des

approches proposées dans la littérature et utilisées dans divers outils de simulation. En seconde partie, nous exposons comment la simulation de trafic en carrefour peut se formaliser dans un cadre multi-agent comme un problème de coordination. Nous y présentons quelques travaux récents dans les systèmes multi-agents qui nous permettront de situer les travaux existant au sein d'ARCHISIM. Ceux-ci sont détaillés et analysés en dernière partie.

2.1 Problématique de simulation de trafic en carrefour

2.1.1 Conduite en file / franchissement d'intersection

Vu à un niveau macroscopique, le trafic est un écoulement de véhicules à l'intérieur d'un réseau routier (section 1.1.3). Les phénomènes de trafic peuvent être vus comme des perturbations dans la circulation des flots de véhicules. Celles-ci peuvent être par exemple liées à la composition du réseau et aux caractéristiques de ces différents tronçons (largeur de la chaussée, nombre de voies, ...) ou à la présence de véhicules lents dans un flot plus rapide. Ces phénomènes résultent d'une combinaison entre deux opérations de manipulation de flux : la *fusion* et le *croisement*.

2.1.1.1 Fusion et croisement de flux

Les fusions de flux de véhicules se rencontrent dans diverses situations de conduite : rétrécissement de la chaussée (travaux, fin de voie,...), insertion dans une file de véhicules (bretelles d'entrée sur route et autoroute, échangeurs), entrée dans un giratoire. Les croisements de flux quant à eux sont observés dans tous les cas d'intersections (figure 2.1).

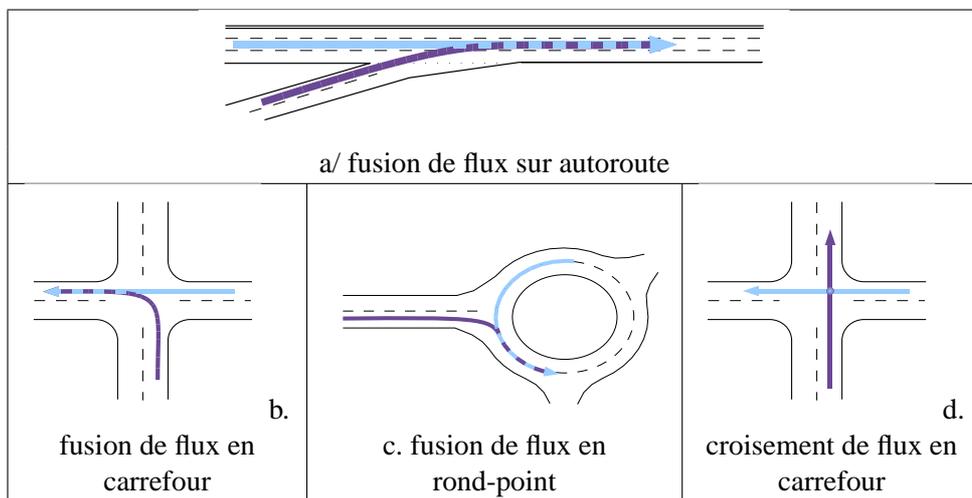


FIGURE 2.1 – Différentes situations de fusions et croisements de flux

Resitués à un niveau microscopique, la fusion et le croisement de flux s'illustrent au travers de deux tâches de conduite bien distinctes : la conduite en file et le franchissement d'intersection.

2.1.1.2 Conduite en file

La conduite en file consiste pour un automobiliste à choisir la voie la plus adaptée à ses contraintes de déplacements et aux différentes interactions qu'il partage avec d'autres conducteurs. Par exemple, un conducteur souhaite maintenir une vitesse désirée à 130km/h sur une portion d'autoroute. Si le véhicule le précédent ne roule qu'à 110km/h, il existe une interaction entre les deux véhicules puisque le plus lent peut contraindre le plus rapide à freiner. Cette interaction peut être de durée variable : le véhicule roulant à 110km/h peut par exemple se rabattre sur la droite (dans le cas où une voie existe) et auquel cas l'interaction sera de courte durée. Si le véhicule ne change pas de voie, l'interaction est de longue durée et nécessite une adaptation : un dépassement ou un freinage.

Le choix de la voie s'effectue en fonction d'un certain nombre de paramètres caractérisant la voie visée. La vitesse instantanée du véhicule, la vitesse moyenne perçue du flux de véhicules sur la voie adjacente, la distance avec le véhicule le précédent sont autant d'indicateurs qui permettent au conducteur d'évaluer l'intérêt du changement de voie. Cette notion de "gain" correspondant à une hausse de la satisfaction des contraintes du conducteur est à mettre en parallèle avec le "coût" requis pour effectuer la manœuvre de changement de voie : efforts pour s'insérer dans un flux de véhicules, stabilité pré-supposée du flux sur la voie visée, etc.

En cas de forte densité et de débit élevé, l'insertion dans le flux de véhicules peut être critique. Il existe alors une zone de conflit entre le véhicule voulant s'insérer et ceux déjà présents sur la voie (figure 2.2).

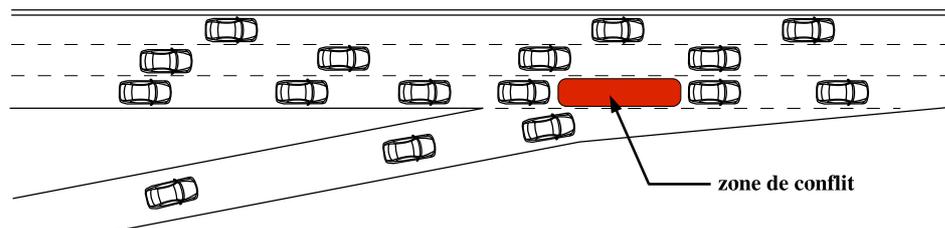


FIGURE 2.2 – Situation conflictuelle en conduite en file (insertion sur autoroute)

Il s'agit alors de trouver un créneau, c'est-à-dire une fenêtre de temps permettant au conducteur d'effectuer la manœuvre en fonction d'un certain nombre de paramètres de sécurité. Le mécanisme de résolution de conflit utilisé dans ARCHISIM pour ce type de situation ([El Hadouaj *et al.*, 2000],[El Hadouaj, 2004]) prend en compte les notions économiques de gains et de coûts relatifs à un changement de voie tout en considérant les différentes contraintes liées à l'approche de l'insertion. La résolution de conflit est difficile puisque dans le cadre d'une insertion sur autoroute, la fin de la voie d'insertion contraint la manœuvre dans le temps.

Les situations de conflits pour la conduite en file impliquent généralement deux ou trois véhicules. En cas de non-résolution du conflit, il s'en suit généralement le blocage d'une partie du trafic (par exemple ralentissement sur une bretelle d'entrée d'autoroute), voire un accident. Les situations de conflits observées dans le cas d'un franchissement d'intersection se révèlent encore plus complexes que celles observées pour la conduite en file.

2.1.1.3 Franchissement d'intersections

Un carrefour est défini par un ensemble de routes dont l'intersection constitue un espace de circulation (centre du carrefour) permettant aux véhicules : soit de changer de direction en s'insérant dans un nouveau flux de circulation (fusion de flux), soit de poursuivre son trajet sur le même axe en traversant un ou plusieurs flux antagonistes (croisement de flux).

Les changements de voie et les multiples combinaisons des différents mouvements tournants contribuent à l'apparition de nombreux points de conflits (figure 2.3). Un véhicule traversant une intersection peut donc avoir plusieurs points de conflits sur sa trajectoire.

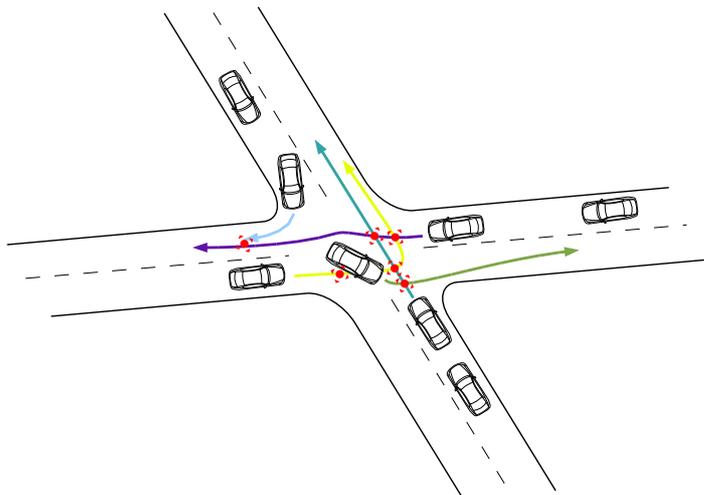


FIGURE 2.3 – Conflits en carrefour

Le franchissement d'intersection est de manière intrinsèque une tâche complexe et difficile. Les objectifs actuels en terme d'aménagements routiers visent d'ailleurs à réaménager les intersections de manière à simplifier la tâche de conduite des automobilistes et limiter les accidents en carrefour. Ainsi, lorsque cela est possible, une intersection est équipée d'un système de contrôleur à feux permettant de coordonner les flux de trafic par des signaux lumineux. Une autre méthode consiste à réagencer l'architecture du carrefour en favorisant les fusions de flux jugées moins dangereuses que les croisements. Ceci se traduit depuis quelques années par une recrudescence du nombre de ronds-points sur nos routes. Malgré cela, les carrefours non-signalisés ou simplement équipés de panneaux statiques (Stop ou Cédez-le-passage) restent encore les plus répandus.

Un conducteur en situation de franchissement d'intersection doit continuellement se coordonner avec le trafic l'environnant (autres véhicules, piétons, ...). Pour ce faire, il doit prendre en compte ses propres objectifs (positionnement sur la voie, direction à suivre) mais également les intentions et comportements des autres usagers de la route. Cette tâche s'avérerait très ardue sans l'aide du Code de la Route.

Mais les usagers de la route (automobilistes, motards, cyclistes, piétons) ne respectent pas toujours les règles du Code de la Route. Les travaux de Björklund et Åberg [Björklund et Åberg, 2005] montrent de plus que celles-ci peuvent être parfois ambiguës et interprétées de manières différentes suivant les personnes. Dans certains cas, elles s'avèrent mal adaptées aux caractéristiques de l'infras-

structure routière, voire totalement en opposition avec un comportement humain naturel. Pour toutes ces raisons, un conducteur développe généralement un ensemble de règles informelles venant compléter ou remplacer celles du Code de la Route.

2.1.1.4 Enjeux et difficultés

Simuler de manière correcte et réaliste le franchissement d'intersection est un problème difficile puisqu'il s'agit de reproduire les comportements individuels des conducteurs humains permettant l'émergence d'une coordination entre les différents flux entrants.

Cette problématique resituée dans un contexte multi-agent est extrêmement intéressante puisqu'elle fait ressortir différents types de difficultés :

- La coordination des flux de trafic conflictuels et antagonistes doit se faire dans un environnement fortement dynamique.
- Il faut généralement composer avec une géométrie non simple de carrefour : angle non-droit entre les axiales, branches multiples, présence d'îlots, etc.
- Les mobiles autonomes simulés doivent être capables de prendre une certaine distance par rapport au Code de la Route. Comme celui-ci n'est pas toujours respecté par les conducteurs, il faut pouvoir rendre compte des normes sociales qui régissent les situations de trafic au centre des carrefours [Björklund et Åberg, 2005].
- Il est nécessaire de prendre en compte les particularités de l'espace au centre de carrefour : il s'agit généralement d'un espace non-structuré (absence de marquage) pouvant servir de zone de stockage pour les véhicules en attente de passer.

Toutes ces difficultés font que la simulation de trafic en carrefour est généralement traitée de manière plutôt simpliste par les différents outils de simulation existants. Dans cette thèse, nous nous focalisons plus particulièrement sur les aspects de coordination en environnement dynamique, de respect du Code de la Route et d'anticipation. La gestion de l'espace au sol fait l'objet de travaux actuellement en cours [Bonte *et al.*, 2006].

2.1.2 Les approches existantes

2.1.2.1 Analyse informelle des outils existants

Comme le souligne [Leclercq, 2002] dans sa thèse, il existe peu de publications détaillant les différents modèles de gestion des conflits à l'intérieur des carrefours comparativement aux lois de poursuite pour lesquelles il existe une littérature complète et abondante. Dans les différents logiciels de simulation microscopique diffusés (outils commerciaux ou de recherche), la gestion des intersections apparaît souvent comme une boîte noire dont le fonctionnement et l'implémentation informatique restent assez obscures.

Ces outils de simulation sont fondés sur des approches mathématiques et peuvent être classés suivant deux critères : le nombre de carrefours pouvant être gérés (carrefour isolé ou réseau urbain) et le type d'intersection simulée. On distingue ainsi les intersections gérées par un contrôleur de feux, les intersections signalisées (ex : rond-point) et les intersections non signalisées (ex : carrefour soumis à la règle de la priorité à droite).

Parmi les simulations de carrefours isolés, nous pouvons citer des outils spécifiques développés dans le but de fournir des indicateurs bien précis sur des intersections gérées par le biais d'un contrôleur de feux. Les logiciels CASIMIR [Cohen et Semaan, 1990] ou TEXAS [Gibson, 1981] permettent d'évaluer et de comparer différentes stratégies de régulation d'un carrefour à feux suivant différents critères tels que : le temps d'attente des véhicules, le nombre de véhicules ayant franchi l'intersection, la consommation de carburant, etc.

D'autres types de logiciels s'intéressent aux intersections signalisées mais non gérées par des feux. Le logiciel KNOSIMO³, par exemple, place les véhicules en approche du carrefour dans des files d'attente virtuelles (une par voie sur chaque branche). Les têtes de chaque file sont déplacées par un mécanisme centralisé de recherche de créneaux d'insertion ou de traversée. De nombreux outils procèdent de cette manière. Cette recherche de créneaux est généralement basée sur le principe de "gap acceptance" qui correspond, pour un conducteur, au temps inter-véhiculaire entre deux véhicules du flux conflictuel qu'il juge acceptable pour pouvoir s'insérer ou traverser.

Des outils de simulation comme SimTraffic⁴, AIMSUN NG⁵ et PARAMICS⁶ généralisent les principes simples de gestion des carrefours précédemment décrits pour des réseaux urbains. Pour ce faire, ils introduisent en plus des règles simples de gestion des "effets de bord" entre carrefours successifs.

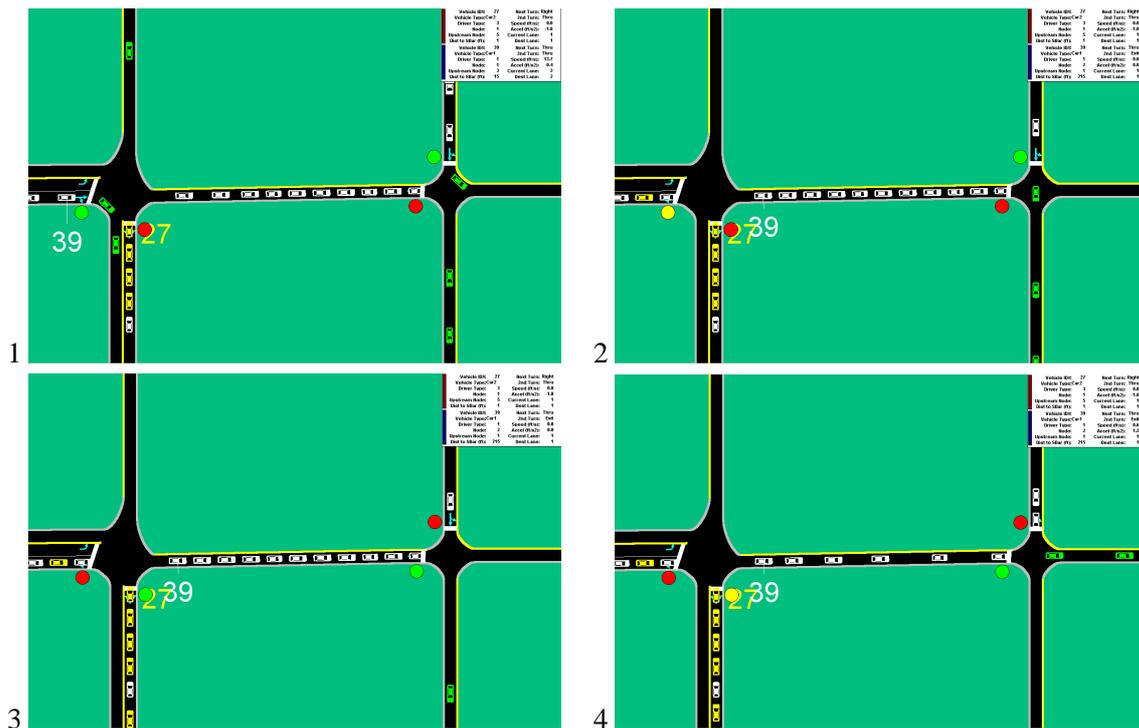


FIGURE 2.4 – Gestion de deux carrefours à feux proches dans le logiciel SimTraffic

3. <http://www.brilonps.de/html/knosimo.html>
4. <http://www.trafficware.com/simtraffic.htm>
5. <http://www.aimsun.com>
6. <http://www.paramics-online.com>

Dans SimTraffic par exemple, lorsque deux carrefours à feux sont relativement proches et que la remontée de queue du carrefour en amont atteint les limites du second carrefour, les véhicules présents n'avancent pas lorsque leur feu passe au vert. Ils attendent que la remontée de queue se réduise pour pouvoir avancer (figure 2.4).

De simples règles comme celles-ci permettent d'éviter les blocages partiels de certaines voies conduisant à des situations conflictuelles complexes. Cependant, dans la réalité il y a fort à parier qu'un conducteur entre dans le carrefour pour diverses raisons. D'une part parce que sa perception locale de la situation ne lui permet pas de savoir ce qu'il se passe dans le carrefour en amont et, d'autre part, parce que le comportement d'un conducteur est souvent individualiste voire égoïste.

Les données de trafic obtenues par les différents outils précédemment cités peuvent être satisfaisantes d'un point de vue statistique, mais les comportements individuels sont trop simplistes et manquent de réalisme. Chaque véhicule se déplace en effet sur des axes filaires prédéfinis lors de l'initialisation de la simulation. Durant la simulation, les véhicules sont donc fortement contraints dans leur déplacement, puisqu'ils ne peuvent se déplacer que sur des rails.

Le second reproche qu'il est possible d'adresser à ces outils concerne la manière peu réaliste dont les conflits sont gérés ou plutôt évités. En effet, le nombre de véhicules présents au centre du carrefour est toujours relativement faible (par rapport à une situation de trafic réel) et les conflits entre ces véhicules sont toujours quasi inexistant.

Il semble que la plupart de ces outils se ramène à un problème d'ordonnancement entre les flots de véhicules présents sur les différentes branches du carrefour. Un processus central prend la décision en lieu et place des véhicules, son rôle est assimilable à un agent de police en charge de la circulation. Ce processus ne laisse rentrer dans le carrefour que les véhicules dont les trajectoires ne sont pas en conflit (figure 2.5).

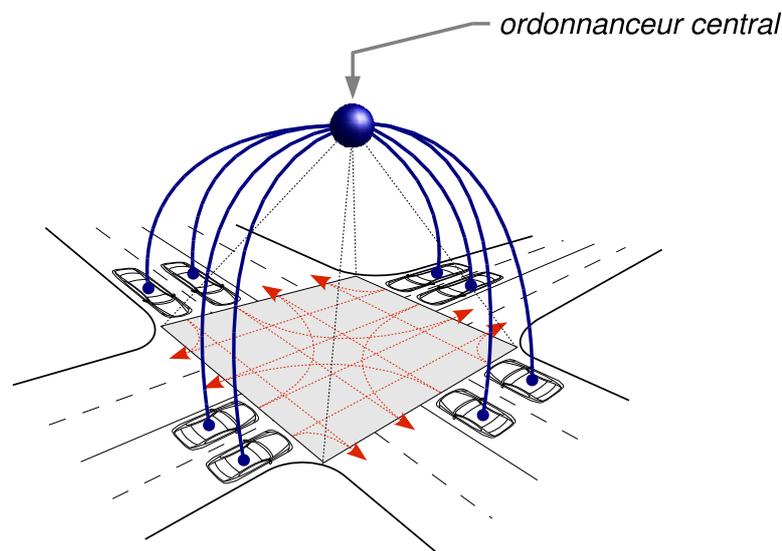


FIGURE 2.5 – Simulation au centre des carrefours vue comme un problème d'ordonnancement

La centralisation de la prise de décision par un seul et même processus contribue également à fausser les comportements simulés puisque les décisions pour chaque véhicule sont prises à partir

d'une connaissance globale de l'état du réseau, ce qui est loin d'être le cas dans la réalité pour des conducteurs humains.

De telles approches peuvent s'avérer suffisantes lorsqu'il s'agit d'étudier des phénomènes trafic en amont ou en aval du carrefour. Auquel cas, il suffit que le processus de gestion du carrefour respecte les densités de trafic en entrée et en sortie ainsi que la répartition en pourcentage des différents mouvements tournants. En revanche, lorsque l'étude porte sur le carrefour lui-même, ces outils montrent leurs limites puisqu'ils ignorent bon nombre de phénomènes observables dans la réalité au centre des carrefours tels que le stockage de véhicules, le blocage partiel de certaines voies, les remontées de queue à l'intérieur de l'intersection, etc.

2.1.2.2 Simulation par automate cellulaire

Des modèles utilisant des automates cellulaires ont également été introduits pour répondre à la problématique de simulation de trafic en carrefour. A mi-chemin entre une approche mathématique et une approche purement comportementale, ces modèles consistent en une discrétisation du carrefour et de l'espace roulant sous forme d'une grille dont la taille des cellules peut varier suivant le niveau de détail désiré.

Dans [Ruskin et Wang, 2002], les auteurs utilisent un automate cellulaire déterministe. L'état d'une cellule est binaire : elle peut être soit vide, soit occupée par un véhicule⁷. Le déplacement des véhicules est géré par les fonctions de transition de l'automate. Chaque véhicule peut se déplacer de v cellules par pas de temps (égal à 1 seconde dans le modèle présenté). La vitesse d'un véhicule est discrétisée en deux ou trois niveaux et varie de 0 à v_{max} ($v_{max} = 1$ ou 2). Les vitesses possibles du véhicule sont fonction de la taille de la cellule. Par exemple, pour une taille de cellule égale à 7 mètres, les vitesses possibles d'un véhicule seront : $0km/h$, $25km/h$ et $50km/h$. A chaque itération, une cellule occupée par un véhicule inspecte donc son voisinage et compte le nombre E de cellules vides la précédant (ou sur son chemin si le véhicule est censé tourner à l'intersection). Si $v < E$ et $v < v_{max}$, le véhicule peut se déplacer.

La gestion des insertions et traversées de flux est ici une variante du principe de *gap acceptance* précédemment évoqué. Il ne s'agit plus "de temps nécessaire" mais "d'espace nécessaire" exprimé en nombre de cellules vides. Pour faciliter la reconnaissance des espaces vides et l'identification des mouvements tournants, chaque cellule comporte un marquage : 0 si elle est vide, a si elle est vide ou occupée par un véhicule tournant à gauche, b si elle n'est pas occupée par un véhicule tournant à droite et c dans le cas contraire (figure 2.6).

Bien que des travaux récents [Wang et Ruskin, 2006] tentent d'améliorer la finesse et le réalisme des modèles, l'approche par automate cellulaire est beaucoup trop simpliste. La discrétisation de l'espace roulant en cellule rend difficile la simulation de carrefours réels à la géométrie non simple (angle des routes non droit, voies multiples) et pose le problème de l'obtention d'un pavage optimal du carrefour. Par ailleurs, elle interdit également toute utilisation combinée avec un simulateur de conduite. Il serait en effet peu réaliste d'insérer dans une scène de conduite des véhicules, à la cinématique limitée, ne disposant que de trois vitesses de déplacement.

7. ou une partie de véhicule puisque le modèle permet de considérer des véhicules longs comme par exemple un bus (figure 2.6)

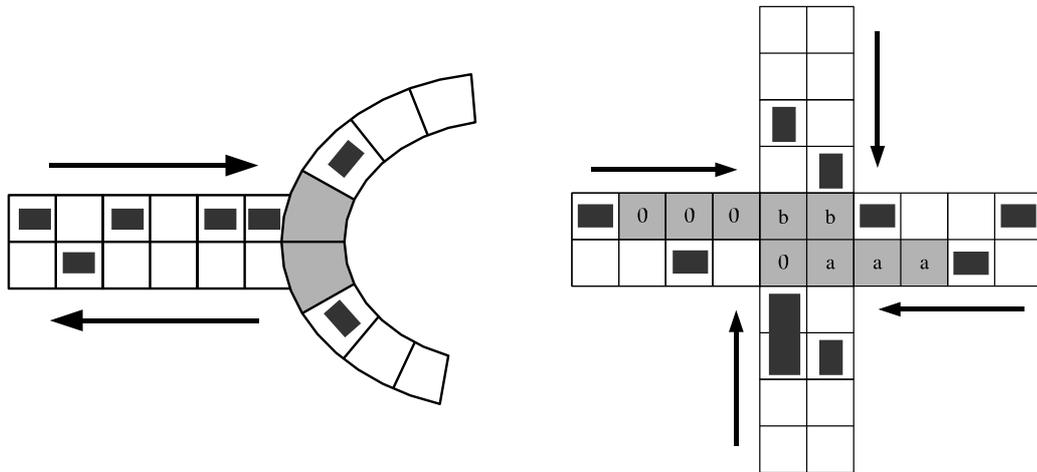


FIGURE 2.6 – Simulation en carrefour (conduite à gauche) à l'aide d'automate cellulaire [Ruskin et Wang, 2002]

2.1.2.3 Approches comportementales issues de la robotique

L'approche comportementale introduite au chapitre 1 se prête mieux à la gestion des interactions en carrefour de par ses principes privilégiant une prise de décision distribuée en chaque entité simulée.

Une approche possible consiste à essayer de distribuer les lois de recherche de créneaux pour les insertions utilisées dans les approches mathématiques. Dans [Wang *et al.*, 2004], les auteurs introduisent un modèle comportemental simple permettant à chaque véhicule simulé de déterminer si sa trajectoire sera coupée par un mobile sur un intervalle de temps $[t_1, t_2]$, t_1 étant le temps d'entrée dans le carrefour et t_2 le temps de sortie estimé. En cas de conflit détecté, la règle proposée est de ne pas entrer dans le carrefour. Dans cette approche, la notion de Code de la Route et de règles informelles de conduite (développées par un conducteur humain) ne sont pas prises en compte. Ces travaux sont donc plus adaptés à la navigation autonome de robot en "milieu urbain simplifié" qu'à une problématique de simulation de trafic comme la nôtre.

Cette démarche roboticienne se retrouve également dans d'autres travaux plus aboutis : [Reece et Shafer, 1993], [Suksthanakar *et al.*, 1996]. Les travaux de Reece se veulent ainsi plus complets en proposant un modèle intégrant les trois niveaux de la tâche de conduite : stratégique, tactique et opérationnel. Destiné à couvrir l'ensemble des situations de conduite, son modèle propose un algorithme pour la prise de décision en carrefour. Celle-ci est décomposée en différents cas suivant la distance à laquelle se trouvent le véhicule conflictuel et le type de signalisation (feux, panneaux) présent à l'intersection.

La figure 2.7 présente la gestion d'un véhicule V arrivant par la droite pour un véhicule autonome R devant franchir une intersection. Cette approche basée sur des modèles à "automate" semble difficilement extensible à une simulation de situation de trafic où plusieurs véhicules peuvent être en conflit en même temps. L'algorithme de franchissement d'intersection est pertinent pour un véhicule autonome (robot) qui par définition respecte le Code de la Route. Dans le cas d'une simulation de trafic, il devient trop simpliste puisque là encore le comportement obtenu est trop normatif par rapport à celui observé chez un conducteur humain.

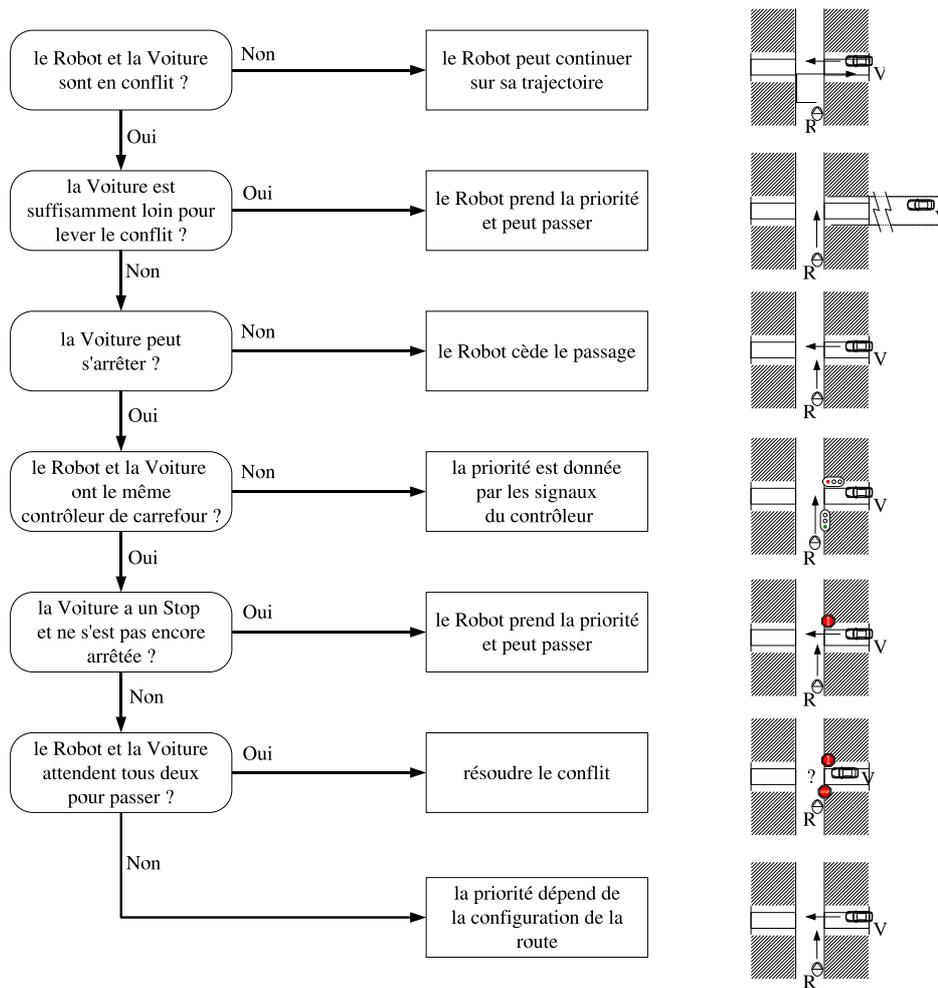


FIGURE 2.7 – Algorithme de décision pour des mobiles en situation de carrefours [Reece et Shafer, 1993]

2.1.2.4 Approche comportementale multi-agent

Les systèmes multi-agents ont été également appliqués à diverses problématiques liées à la gestion, l'exploitation et la simulation de trafic routier en situation de carrefour.

Dans [Desner et Stone, 2004], les auteurs présentent un mécanisme multi-agent de gestion de trafic en intersection. Le principe repose sur un système de réservation : chaque agent récupère de l'intersection un créneau durant lequel il peut traverser en toute sécurité. Ce mécanisme dont l'efficacité est évaluée par simulation offre des temps d'attente et des débits meilleurs qu'un système classique de feux tricolores. Une telle approche semble difficile à transposer dans le cadre de notre problématique puisqu'elle repose sur des communications entre les agents et l'infrastructure.

Dans [Paruchuri *et al.*, 2002], les auteurs proposent un modèle multi-agent relativement complet possédant un grand nombre de paramètres et permettant de rendre compte de comportements de conducteurs relativement réalistes. Les aspects psychologiques de la conduite sont pris en compte au niveau des calculs cinématiques par le biais de caractéristiques individuelles propres à chaque

agent. La prise de décision est individuelle mais reste supervisée par des processus centralisés destinés à prendre le contrôle dans certaines situations comme par exemple des embouteillages ou des interblocages en carrefour (figure 2.8).

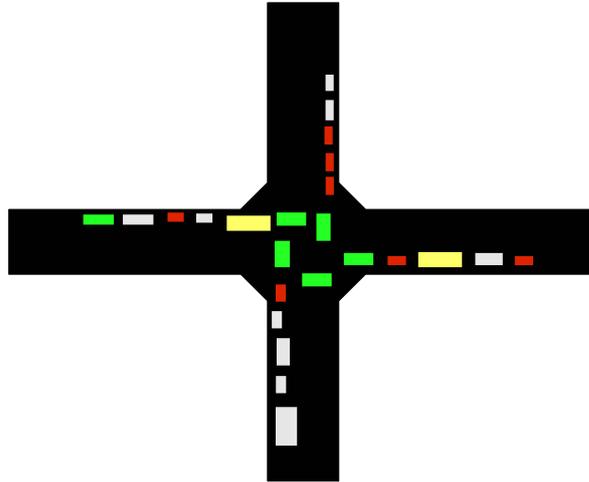


FIGURE 2.8 – Situation d'interblocage nécessitant l'intervention d'un processus centralisé [Paruchuri *et al.*, 2002]

L'approche multi-agent la plus pertinente pour notre problématique consiste en une adaptation du principe de blackboard évoqué au chapitre 1. Dans [Tranois *et al.*, 1998], les auteurs proposent un mécanisme individuel de planification d'actions pour des agents évoluant dans un environnement urbain. L'originalité de leur approche réside dans la modélisation de l'environnement et la répartition des connaissances relatives à la tâche de conduite. Celles-ci ne sont pas intégrées en chaque agent mais réparties entre les différentes entités constituant l'environnement routier.

Ainsi, un feu tricolore fournira aux agents qui s'en approchent les connaissances nécessaires au franchissement d'une intersection gérée par un contrôleur de ce type. La planification des actions d'un agent suit le triptyque "Perception-Cognition-Action" présenté en section 1.2.2. L'infrastructure est considérée comme un agent capable de délivrer un certain nombre d'informations aux mobiles simulés. Dans une première phase, un agent "véhicule" reçoit de l'agent "carrefour" l'ensemble des éléments présents dans l'environnement. En fonction de cette perception, l'agent met à jour sa base de connaissances exprimée sous la forme de règles et de faits. Ces connaissances sont ensuite ordonnées suivant leur pertinence de manière à pouvoir déclencher le raisonnement de l'agent. Lorsque plusieurs actions sont envisageables aux vues des connaissances de l'agent, l'action mise en œuvre est celle suggérant la vitesse la plus faible.

L'avantage de cette approche est que chaque agent prend sa décision suivant un contexte bien précis, ce qui permet de rendre compte de la grande variété et de la complexité des situations de trafic observables en carrefour. Par ailleurs, la distribution des connaissances à travers l'environnement permet aux agents de ne raisonner que sur un sous-ensemble de connaissances pertinent pour la situation de trafic dans laquelle il est impliqué. Cela garantit des temps de calculs acceptables permettant d'envisager des simulations en temps réel.

Les approches multi-agents présentées dans cette section ont été conçues pour répondre à des problématiques bien précises liées au trafic routier. Nous poursuivons notre état de l'art en étudiant la

possibilité d'adapter à notre problématique un des nombreux mécanismes de coordination multi-agent présents dans la littérature.

2.1.3 Synthèse

Le tableau 2.1 résume les différentes approches dédiées au trafic routier et leurs principaux inconvénients.

	Avantages	Inconvénients
Ordonnancement	Mise en œuvre aisée	Vision centralisée du trafic, peu de véhicules présents en même temps au centre du carrefour
Automate cellulaire	Mise en œuvre aisée	Modèle cinématique limité
Approches roboticiennes ([Reece et Shafer, 1993], [Suksthanakar <i>et al.</i> , 1996])	Vision distribuée du trafic	Comportement trop normatif par rapport à celui d'un conducteur réel
Travaux de Desner et Stone	Approche décentralisée du trafic	Approche dédiée à l'optimisation de flux de trafic et non à la simulation
Modèle décentralisé de gap acceptance [Paruchuri <i>et al.</i> , 2002]	Modèle riche, nombreux paramètres	Gestion centralisée des interblocages
Ardoise contextuelle [Tranois <i>et al.</i> , 1998]	Prise de décision individuelle des conducteurs simulés, richesse du modèle	Supervision centralisée de certaines situations de trafic

TABLE 2.1 – Synthèse des approches spécifiques au trafic routier

2.2 Coordination en environnement dynamique

Au sens général, la coordination se définit comme l'ensemble des actions permettant une production et/ou une organisation en vue d'atteindre un objectif déterminé. La coordination constitue l'une des facettes de l'interaction existant entre les agents d'un SMA. Elle fait référence à tous les processus utilisés dans la prise de décision et au comportement global d'un ensemble d'agents.

2.2.1 Coordination : généralités

Les objectifs de la coordination peuvent varier suivant l'application. D'une manière générale, la coordination a pour but d'atteindre un état global respectant un ensemble de contraintes en fonction des états locaux des différents agents impliqués. Par exemple, l'état global souhaité peut être une solution à atteindre dans le cadre d'une résolution distribuée de problème.

2.2.1.1 Différentes facettes de la coordination

Les multiples définitions données dans la littérature ([Malone, 1998], [Gasser, 1992], [Sycara *et al.*, 1991]) montrent que les motivations de la coordination sont diverses et nombreuses.

La coordination peut ainsi avoir pour perspective le raisonnement d'agents ayant des points de vue antagonistes. Il s'agit pour un bon nombre de problèmes de confronter et concilier des décisions conflictuelles dont les répercussions au sein du système peuvent nuire à l'obtention de l'état global recherché.

La coordination peut encore se définir comme un ensemble de méta-actions que les agents exécutent en complément de leurs tâches courantes de manière à interagir avec le groupe. Malone présente la coordination d'actions comme *l'ensemble des activités supplémentaires qu'il est nécessaire d'accomplir dans un environnement multi-agent et qu'un agent seul ne peut s'acquitter* [Malone, 1998].

Pour Jennings, *la coordination est le processus par lequel un agent raisonne sur ses propres actions et sur celles de ses congénères de manière à s'assurer que le système se comporte de manière correcte* [Jennings, 1996]. La coordination est donc le moyen de satisfaire et garantir différents critères de cohérence du système sans l'intervention d'un processus global. La coordination est alors vue comme un problème de contrôle distribué.

La coordination est également un moyen de résoudre différents types de conflits entre des agents. L'accès à une même ressource (service, outil, autres agents, ...) peut, par exemple, provoquer des conflits se traduisant par des phénomènes de blocages, famines, etc. Qu'ils aient des buts communs ou antagonistes, les agents doivent développer et appliquer des stratégies d'accès et de partage des ressources communes : négociation, définition d'une structure organisationnelle, etc.

Enfin, la coordination peut être vue comme les différents processus mis en œuvre par un ensemble d'agents pour la résolution distribuée d'un problème ([Yokoo, 2001], [Doniec *et al.*, 2005c]). La coordination englobe alors les raisonnements internes aux agents, les mécanismes de résolution entre agents.

2.2.1.2 Coopération vs compétition

Les premiers travaux en coordination se sont intéressés aux aspects de coopération en faisant l'hypothèse que les agents sont motivés collectivement à l'accomplissement d'un but commun. La coordination a donc d'abord été abordée comme une résolution distribuée de problèmes.

Dans [Lesser et Corkill, 1981], le modèle de coordination est basé sur l'acquisition d'informations par chaque agent et sur l'utilisation de stratégies reposant sur des échanges de messages décrivant les buts, les priorités et les préférences des agents. Ce type d'approche a permis d'exprimer le concept de coordination de manière plus formelle comme un problème de satisfaction de contraintes distribuées [Piechowiak et Hamadi, 2002] dans lequel chaque agent a en charge la résolution d'un sous-problème. La solution globale est obtenue par concertation entre les agents et échanges de solutions partielles ([Yokoo, 2001], [Doniec *et al.*, 2005c]).

D'autres travaux ont abordé la coordination sous l'angle du partage de tâches ou de la planification distribuée que nous détaillerons en section 2.2.2.

Plus récemment, il a été émis l'idée que la coordination n'est pas forcément coopérative. Il est possible de considérer que les agents soient motivés individuellement à l'accomplissement de buts

qui leur sont propres tout en essayant de conserver certaines propriétés au niveau du groupe. Il s'agit alors d'une coordination qualifiée de compétitive.

Dans [Shoham et Tennenholtz, 1995], les auteurs introduisent le concept de lois sociales qui permettent de restreindre les actions d'agents motivés par des buts antagonistes (déplacement de robots dans un espace restreint) et ainsi minimiser le nombre de résolutions de conflits. La coordination d'agents compétitifs peut être également perçue comme un problème de formation de coalition qui consiste à trouver une solution globale la plus satisfaisante pour la majorité des agents. Dans [Aknine *et al.*, 2000], les auteurs proposent un mécanisme de coordination basé sur des agrégations multi-critères réutilisant des techniques d'aide à la décision et d'optimisation.

Dans les sections qui suivent, nous présentons différents mécanismes de coordination permettant la résolution de conflits entre agents. Nous les séparons en deux parties : les mécanismes de coordination cognitifs et les mécanismes de coordination réactifs (section 1.2.2).

2.2.2 Mécanismes de coordination d'agents cognitifs

Nous détaillons ici quelques mécanismes de coordination de la littérature appliqués ou applicables à la résolution de conflits. Pour chaque approche, nous analysons son adéquation avec notre problématique de simulation de flux de trafic conflictuels en carrefour.

2.2.2.1 Négociation

La négociation est souvent présentée comme un raisonnement de haut niveau permettant de concilier entre eux des agents dont les intérêts sont divergents voire opposés. Les nombreux mécanismes existants s'inspirent de la richesse des négociations entre humains : métaphore du contrat établi entre un gestionnaire et des sous-traitants (réseau contractuel [Smith, 1980]), utilisation d'un agent médiateur [Sycara, 1989], etc.

Une négociation porte sur un sujet particulier, source de conflit entre agents. Le processus de négociation en lui-même consiste en une série d'échanges d'informations (communication) entre les agents, entrecoupés de phases d'évaluation des propositions reçues et d'élaboration de réponses. Les communications sont régies par un formalisme précis comprenant un langage commun à tous les agents et un ensemble de règles et de conventions définissant le protocole et les séquences d'échanges de messages. Les formulations de proposition et leur évaluation par un tiers sont fonction de l'application. Il peut, par exemple, s'agir de déterminer un compromis entre les agents du système. Chaque proposition émise peut être évaluée par un agent en fonction d'un ensemble de contraintes qu'il cherche à satisfaire. L'obtention du compromis est obtenue à la suite de relâchement successif de contraintes de la part des agents impliqués dans la négociation.

La négociation est traitée de manière très riche dans la littérature pour des applications aussi diverses que variées. La proposition de modèles génériques de négociation facilite l'implémentation et la réutilisation des différentes formes de négociation [Verrons, 2004]. Mais, bien qu'elle soit couramment utilisée pour la résolution de conflits entre agents, la négociation semble conceptuellement très éloignée de la tâche de conduite.

Une négociation réussie (résolution du ou des conflits) est généralement le résultat d'une succession d'échanges verbaux entre agents concernant leurs buts, intentions, etc. Les processus mis en jeu peuvent donc être relativement longs en terme de temps d'exécution.

Par ailleurs, une négociation requiert une communication évoluée (échanges de messages, utilisation d'un langage commun, actes de langage). Dans notre contexte applicatif, les agents (conducteurs virtuels) ne peuvent ni communiquer entre eux ni verbaliser de manière concrète leurs buts. L'utilisation d'un mécanisme de négociation nécessiterait donc une certaine abstraction par rapport à la réalité. Il serait possible d'imaginer la mise en place d'un "agent médiateur de carrefour" chargé de recueillir les objectifs de chaque agent en approche de carrefour (voie visée, direction à suivre, etc) et de proposer en fonction des demandes reçues différentes fenêtres de temps. Une argumentation entre agents non satisfaits du créneau obtenu pourrait alors s'engager avec comme support certains paramètres tels que le temps d'attente. Mais une telle démarche trouverait rapidement ses limites dans la mesure où parmi les mobiles en approche du carrefour se trouverait un être humain immergé dans la simulation.

2.2.2.2 Planification

La planification fait également partie des raisonnements de haut niveau permettant à des agents de se coordonner. Un agent peut planifier ses objectifs futurs sous la forme de plans contenant les actions et les interactions dont il devra s'acquitter. La coordination multi-agent par planification consiste à obtenir à partir des plans individuels de chaque agent un plan multi-agent décrivant le comportement global du système. Il s'agit d'identifier les enchaînements séquentiels possibles entre toutes les actions, les exécutions en parallèle mais aussi les conflits potentiels qui peuvent survenir.

La planification peut être réalisée par un seul et même agent. Dans [Georgeff, 1983], l'auteur introduit un agent "synchroniseur" qui garantit l'exclusion mutuelle des actions et sous-butts conflictuels. Dans d'autres cas, la synthèse des plans individuels peut nécessiter plusieurs agents. Ceux-ci interagissent tous ensemble à la construction d'un plan global suivant des mécanismes d'échanges de sous-plans. Une autre approche consiste à envisager la construction de plans partagés liés à une activité conjointe de plusieurs agents ([Cohen et Levesque, 1990], [Mandiau, 1993]).

Plus récemment, des travaux appliquent la planification à des problématiques pour lesquelles les agents sont situés dans un environnement fortement dynamique. L'exécution d'un plan par un agent spatialement situé pose de nombreuses difficultés liées au fait que les actions planifiées ne tiennent pas compte des positions des éléments impliqués dans l'exécution de la tâche. Dans [Devigne *et al.*, 2004], les auteurs séparent le *plan abstrait dans lequel les positions relatives et absolues ne sont pas prises en compte* du *plan concret qui lui dépend du contexte d'exécution*. Pour cela, ils considèrent les dépendances spatiales entre les éléments (agents ou objets) impliqués dans la planification. Les auteurs introduisent ainsi une méthodologie permettant d'obtenir un plan concret en intégrant les déplacements des agents entre chaque action atomique du plan abstrait.

Un environnement dynamique pose également d'autres types de contraintes que la spatialisation des actions. La planification doit ainsi être capable de s'adapter aux perturbations extérieures subies par l'environnement. Non seulement les plans individuels de chaque agent doivent être recalculés mais la dynamique collective du groupe doit aussi être redéfinie. De manière à prendre en compte ses contraintes, Frédéric Marc propose dans sa thèse une modélisation des plans des agents sous la forme d'automate hybride [Marc, 2005]. L'ensemble des automates donne un réseau qui représente des plans multi-agents. Une telle approche permet d'obtenir une flexibilité du comportement collectif des agents et facilite la validation du plan exécuté [El Fallah-Seghrouchni *et al.*, 2004].

Même si elles ont été appliquées avec succès à la simulation de missions aériennes [Marc, 2005], les approches par planification restent relativement lourdes à mettre en place et requièrent encore

des temps de calcul ne permettant pas le couplage d'une simulation de trafic impliquant plusieurs centaines de véhicules avec un simulateur de conduite.

2.2.3 Mécanismes de coordination d'agents réactifs

La coordination réactive s'intéresse aux mécanismes simples mis en jeu par des agents réactifs en leur permettant d'agir ensemble à l'accomplissement d'un but commun (coopération) ou de résoudre leurs conflits d'intérêts (utilisation d'une ressource commune, accès à un même espace).

2.2.3.1 Mécanismes d'attraction et de répulsion

Les mécanismes de coordination réactifs sont issus pour certains d'entre eux de l'observation de phénomènes naturels. Pour créer ses "boids", Reynolds s'est par exemple inspiré des vols d'oiseaux [Reynolds, 1987]. Chaque "boid" coordonne sa vitesse, son déplacement et sa position par rapport aux autres en appliquant trois règles simples définissant des comportements de séparation, d'alignement et de cohésion.

De nombreux mécanismes de coordination réactive s'inspirent donc de ce principe de répulsion et d'attraction. Les champs de forces utilisés par Zeghal en simulation aérienne en sont une illustration [Zeghal, 1993]. Chaque agent associe aux autres agents qu'il perçoit comme conflictuels une force dont la valeur dépend du risque de collision et dont la direction indique le point de fuite permettant d'éviter la collision.

2.2.3.2 Auto-organisation coopérative issue de la théorie des AMAS

La théorie des systèmes multi-agents adaptatifs (AMAS) [Glize, 2000] considère *la coopération comme un mécanisme auto-organisateur permettant l'obtention de comportements globaux émergents pour des systèmes plongés en environnement dynamique* [Picard et Gleizes, 2006]. Pour ce faire, les agents agissent et se réorganisent au niveau local pour éviter les situations non-coopératives.

Dans [Picard et Gleizes, 2005], les auteurs proposent un modèle d'agents coopératifs issu de la théorie des AMAS. L'application consiste en un système multi-robots dans lequel les robots doivent effectuer des tâches de transports. Chaque robot est implémenté comme un ensemble de modules décrivant ses capacités de perceptions, de représentations et d'actions.

Le fonctionnement interne d'un robot suit le triptique : Perception-Décision-Action. Ici, la phase de décision est réalisée par l'exécution en parallèle de deux modules : l'un fournissant un comportement nominal, l'autre préconisant un comportement d'adaptation en cas de détection d'une situation non-coopérative. Le module en charge du comportement coopératif est construit autour d'un ensemble de règles de type Condition-Action. Cette approche réactive est complétée par un module de mémorisation permettant aux robots d'éviter de répéter des erreurs de coopération passées.

Les expérimentations menées par les auteurs exhibent des propriétés intéressantes comme l'émergence d'une circulation "intelligente" répondant à la problématique de transport collectif, l'utilisation de connaissances de bas niveau (pas de vision globale), la robustesse de l'approche, etc.

2.2.3.3 Modèle de satisfaction-altruisme

Dans [Simonin, 2001], l'auteur introduit un mécanisme de coordination réactif basé sur le passage des agents d'un état normal à un état d'altruisme. Chaque agent pourvoit à la satisfaction de ses propres besoins jusqu'à la réception de signaux indiquant l'insatisfaction des autres agents du système. La règle de passage à l'état d'altruisme pour un agent est fonction du niveau d'insatisfaction perçue de son entourage. L'émission et la réception des signaux reposent sur un système de communication relativement simple et applicable à la robotique mobile.

Simonin applique notamment son modèle à la résolution de conflits spatiaux ([Simonin, 2001], chapitre 7). Le principe de résolution s'appuie sur la propagation et la perception de signaux locaux entre les agents. A chaque déplacement, les agents émettent des signaux de satisfaction interactive indiquant à leur entourage une gêne ou un blocage quantifiés sous la forme d'une valeur numérique. Les agents les moins contraints recevant les signaux d'insatisfaction des agents bloqués basculent dans un comportement altruiste et se mettent à reculer pour libérer le passage. Les blocages en chaîne sont résolus par propagation et amplification des signaux d'insatisfaction.

La figure 2.9 présente l'algorithme utilisé pour la résolution de conflits spatiaux. Il s'appuie sur quatre règles de comportements :

- Règle 1 : Elle définit le passage de l'état normal à l'état d'altruisme d'un agent : tout signal reçu est prioritaire sur ses propres émissions et sa satisfaction personnelle provoquant son changement d'état vers l'altruisme.
- Règle 2 : Elle conditionne l'émission des signaux répulsifs lorsque le taux d'insatisfaction personnelle d'un agent agissant de manière non-altruiste dépasse un certain seuil.
- Règle 3 : Un agent bloqué et en train d'émettre des signaux d'insatisfaction continue tant que des agents restent présents sur son chemin. La valeur émise à chaque signal augmente tant que l'insatisfaction de l'agent augmente également.
- Règle 4 : Elle permet la propagation des signaux répulsifs lorsque il y a présence d'une chaîne de blocage composée de plusieurs agents.

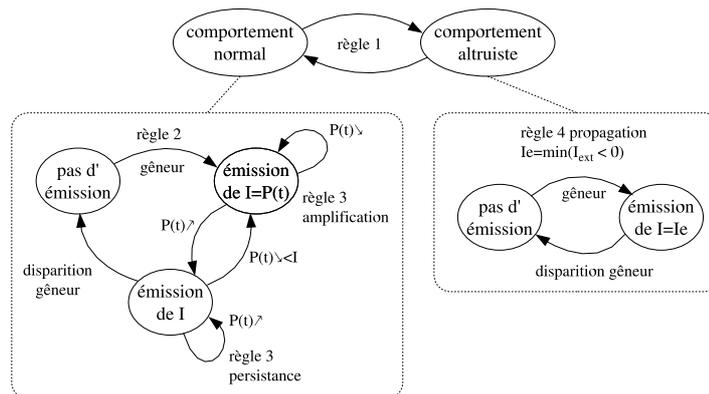


FIGURE 2.9 – Algorithme de résolution de conflits spatiaux utilisant le modèle de satisfaction altruisme [Simonin, 2001]

2.2.3.4 Adéquation des approches réactives avec notre contexte

D'une manière générale, l'utilisation d'une méthode de coordination réactive pour la résolution des conflits n'est pas applicable dans notre contexte de simulation de trafic. En effet, les véhicules dont nous simulons le déplacement ont des contraintes en terme de braquage et ne disposent donc pas des mêmes degrés de liberté que les agents situés de [Simonin, 2001]. Ainsi, en cas de blocage entre plusieurs agents, ceux-ci peuvent changer de direction de manière diamétralement opposée. Dans le cas de véhicules, il serait nécessaire d'effectuer une marche arrière. Or il s'agit là d'une limitation d'ARCHISIM qui pour l'instant interdit toute marche arrière. Cette limitation est justifiée par le couplage entre ARCHISIM et un simulateur de conduite. Pour pouvoir effectuer une marche arrière dans de bonnes conditions sur un simulateur, il est nécessaire que le conducteur immergé puisse bénéficier d'un visuel 3D de la situation de conduite sur 360 degrés, ce qui nécessite la mise en place d'équipements de projection coûteux. Le choix a donc été fait de ne pas gérer la marche arrière dans le modèle de trafic.

Un second obstacle à l'application de cette approche provient du fait qu'en situation de conduite, les conducteurs réels résolvent leurs conflits sans forcément laisser apparaître de blocage. En d'autres termes les décisions d'un conducteur réel sont anticipées et très peu réactives⁸.

2.2.4 Synthèse

Le tableau 2.2 résume les différentes approches de coordination multi-agents et leurs inadéquations avec la problématique de simulation de trafic routier.

	Avantages	Inconvénients
Négociation	Bien adaptée à la résolution de conflits de ressources	Conceptuellement éloignée de la tâche de conduite, nécessite des actes de communication évolués
Planification	Adaptée aux environnements dynamiques ([Marc, 2005], [Devigne <i>et al.</i> , 2004])	Temps de calcul prohibitifs
Coordination réactive	Bien adaptée à la résolution de conflits spaciaux (robotique [Simonin, 2001], transport aérien [Zeghal, 1993])	La résolution du conflit dépend du degré de liberté de l'agent, pas d'anticipation des conflits

TABLE 2.2 – Synthèse des approches de coordination multi-agent

8. sauf cas extrêmes où un conducteur peut avoir des réactions réflexes lui permettant de faire face à une situation de danger imprévue

2.3 Mécanisme de coordination pour la simulation de trafic en carrefour

La difficulté à réutiliser un mécanisme de coordination existant a conduit l'équipe MSIS (S. Espié) en collaboration avec le LAMIH (R. Mandiau et C. Kolski) à développer un mécanisme spécifique répondant aux diverses contraintes de l'application [Champion, 2003]. Celui-ci consiste en une extrapolation de la notion de jeux appliquée aux situations conflictuelles en carrefour. La première partie s'attache à introduire de manière succincte les différents concepts relatifs à la théorie des jeux. La seconde présente le mécanisme de coordination résultant d'une modélisation des conflits en carrefour à l'aide de la théorie des jeux.

2.3.1 Fondement théorique

La théorie des jeux fournit un cadre formel pour l'étude des situations conflictuelles entre individus partageant un même intérêt. Les premiers concepts de cette théorie ont été introduits au début des années vingt par le mathématicien français Emile Borel. La formalisation mathématique qu'on lui connaît aujourd'hui date de 1928 et est due à l'américain von Neumann.

Les véritables applications de la théorie des jeux n'ont vu le jour que dans la deuxième moitié des années quarante après la publication de "*Theory of Games and Economic Behaviour*". Ecrit par von Neumann et l'économiste autrichien Morgenstern, cet ouvrage bouleversa les sciences économiques et reste encore de nos jours une référence.

Outre l'économie, de nombreuses disciplines se sont penchées sur les concepts novateurs qu'apporte la théorie des jeux. Les jeux évolutionnaires sont, par exemple, une application des concepts mathématiques de la théorie des jeux dans un contexte biologique. Les espèces animales et même les gènes sont considérés comme des joueurs et leur évolution peut ainsi être étudiée d'un point de vue mathématique. Des études en sociologie et en psychologie tirent également parti de la théorie des jeux pour modéliser les comportements humains dans certains contextes. Dans [Bjørnskau et Elvik, 1992], les auteurs utilisent la notion de jeux pour évaluer l'impact de politiques sécuritaires en trafic routier notamment en terme de réduction du nombre d'accidents. D'une manière générale, comme le souligne Gintis dans la préface de son livre [Gintis, 2000], *la théorie des jeux est un langage universel pour l'unification des sciences traitant du comportement*.

En informatique, la première application de la théorie des jeux et plus particulièrement du théorème du Minimax a été la mise au point de programmes capables de jouer aux échecs contre un être humain. L'intelligence artificielle distribuée et l'étude d'agents logiciels autonomes interagissant entre eux offrent aujourd'hui un nouveau champ d'application pour la théorie des jeux.

2.3.1.1 Définitions

De manière informelle, un jeu regroupe classiquement un ensemble de joueurs (j_1 à j_n) respectant un certain nombre de règles (ordre entre les joueurs, calcul des gains, ...) et appliquant une ou plusieurs stratégies dans le but de gagner. Les nombreux travaux de formalisation autour de la théorie des jeux ont conduit à l'introduction de deux représentations mathématiques des jeux.

Définition 2.1 Sous sa forme stratégique, un jeu est défini par un ensemble de joueurs $j_i : \{j_1, j_2, \dots, j_n\}$ disposant d'un ensemble de stratégies (actions) $S_{j_i} = \{a_{j_i}^1, a_{j_i}^2, \dots, a_{j_i}^m\}$ et d'une matrice $S_{j_1} \times S_{j_2} \times \dots \times S_{j_n} \mapsto \mathbb{N}$ ou \mathbb{R} définissant les gains espérés pour chaque stratégie.

La forme extensive d'un jeu privilégie une vision de son déroulement sous une représentation arborescente où chaque noeud de décision exprime une situation de jeu et chaque feuille correspond à une issue du jeu. Par exemple, pour un jeu à 2 joueurs j_1 et j_2 , la racine correspond aux coups joués par j_1 . Le premier niveau de l'arbre énumère toutes les situations de jeu possibles pour j_2 en fonction du coup précédent joué par j_1 . Le nombre total de feuilles de l'arbre correspond au nombre de parties possibles du jeu.

Chaque feuille peut être étiquetée par une valeur numérique correspondant aux gains attendus pour la réalisation de l'issue de jeu correspondante. On trouve généralement dans la littérature le terme de *fonction d'utilité*. Celle-ci est définie de l'ensemble I des issues d'un jeu à n joueurs vers \mathbb{N}^n ou \mathbb{R}^n . Pour les jeux dits à un seul tour (où chaque joueur ne joue qu'une seule fois), cette fonction d'utilité correspond à la matrice des gains utilisée dans la définition stratégique d'un jeu.

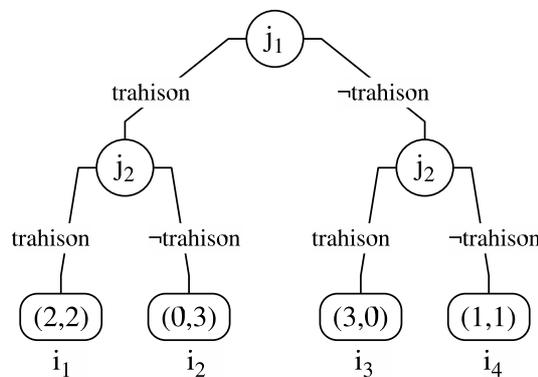
2.3.1.2 Exemple du dilemme du prisonnier

Deux joueurs j_1 et j_2 sont faits prisonniers suite à un délit qu'ils ont commis ensemble. Les preuves n'étant pas suffisantes pour les condamner, la police leur propose séparément un marché. Si l'un des joueurs trahit son partenaire en le dénonçant, il est libre tandis que l'autre est puni de 3 ans de prison. Si les deux se trahissent mutuellement, leur peine est de 2 ans. Si aucun des deux ne collabore, leur peine est réduite à 1 an.

Cet énoncé peut être synthétisé de la manière suivante, sous sa forme stratégique :

$$\begin{aligned}
 t(x, y) &\equiv x \text{ trahit } y \\
 S_{j_1} &= \{t(j_1, j_2), \neg t(j_1, j_2)\} & S_{j_2} &= \{t(j_2, j_1), \neg t(j_2, j_1)\} \\
 G_{j_1} : S_{j_1} \times S_{j_2} &\mapsto \mathbb{N} & G_{j_2} : S_{j_2} \times S_{j_1} &\mapsto \mathbb{N} \\
 G_{j_1} &= \begin{pmatrix} 2 & 0 \\ 3 & 1 \end{pmatrix} & G_{j_2} &= \begin{pmatrix} 2 & 3 \\ 0 & 1 \end{pmatrix}
 \end{aligned}$$

et sous sa forme extensive :



Sous cette représentation arborescente, nous pouvons voir que ce jeu est à un seul tour : chaque joueur joue une seule fois et en séquence. Le nombre d'issues de ce jeu est égal à 4 : $I = \{i_1, i_2, i_3, i_4\}$ avec :

$$\begin{aligned} i_1 &\equiv j_1 \text{ trahit } j_2 \text{ qui le trahit aussi} \\ i_2 &\equiv j_1 \text{ trahit } j_2 \text{ qui ne le trahit pas} \\ i_3 &\equiv j_1 \text{ ne trahit pas } j_2 \text{ qui lui le trahit} \\ i_4 &\equiv j_1 \text{ ne trahit pas } j_2 \text{ qui ne le trahit pas non plus} \end{aligned}$$

Chaque feuille est associée à un vecteur décrivant les gains attendus de chacun des deux joueurs. Les fonctions d'utilité u_{j_1} et u_{j_2} pour ce jeu se définissent comme suit :

$$\begin{aligned} u_{j_1} : I &\mapsto \mathbb{N} & u_{j_2} : I &\mapsto \mathbb{N} \\ \left\{ \begin{array}{l} u_{j_1}(i_1) = 2 \\ u_{j_1}(i_2) = 0 \\ u_{j_1}(i_3) = 3 \\ u_{j_1}(i_4) = 1 \end{array} \right. & & \left\{ \begin{array}{l} u_{j_2}(i_1) = 2 \\ u_{j_2}(i_2) = 3 \\ u_{j_2}(i_3) = 0 \\ u_{j_2}(i_4) = 1 \end{array} \right. \end{aligned}$$

2.3.1.3 Éléments et propriétés

Lorsque les joueurs connaissent les actions des autres joueurs participant ainsi que les issues possibles, le jeu est dit à *information complète*. A l'inverse lorsque les joueurs ne disposent d'aucune information concernant les stratégies adverses, le jeu est dit à *information incomplète*. En le considérant sous sa forme extensive, un jeu peut être qualifié de jeu à *information parfaite* lorsque pour chaque étape les joueurs peuvent connaître précisément le noeud de l'arbre atteint.

Définition 2.2 *Un jeu est dit strictement compétitif si et seulement si pour tout couple (i, j) d'issues possibles, il existe deux joueurs j_1 et j_2 tel que :*

$$(G_{j_1}(i) - G_{j_1}(j))(G_{j_2}(i) - G_{j_2}(j)) < 0$$

Concrètement, un jeu strictement compétitif permet de rendre compte d'une situation dans laquelle les intérêts des joueurs sont divergents et progressent dans des directions opposées. Lorsque pour toute issue la somme des gains est égale à zéro, le jeu est dit à *somme nulle*.

2.3.1.4 Résolution

L'une des contraintes fortes de la théorie des jeux est de considérer que tous les joueurs sont rationnels : entre deux issues de jeux possibles, un joueur préférera celle ayant le gain le plus élevé. Cette hypothèse de rationalité des joueurs permet de définir la solution d'un jeu comme une combinaison de stratégies permettant d'aboutir à une issue optimale en terme de gain.

En théorie des jeux, la rationalité des joueurs peut s'exprimer à deux niveaux : le niveau individuel et le niveau collectif. La rationalité appliquée à un ensemble de joueurs se traduit par l'optimalité selon Pareto.

Définition 2.3 Soit un jeu à n joueurs $j_k, k \in \{1..n\}$ possédant un ensemble d'issues possibles I . Soit $i \in I$. i est dit Pareto-optimale si et seulement si :

$$\forall i' \in I \setminus \{i\} \quad \exists j_k \in \{j_1, \dots, j_n\} \quad u_{j_k}(i') < u_{j_k}(i)$$

Littéralement cela signifie qu'une issue ne peut être Pareto-optimale dès qu'il existe une autre issue préférée par une majorité de joueurs. Concrètement, la Pareto optimalité traduit donc une satisfaction collective des joueurs.

La rationalité du point de vue d'un joueur individuel est souvent assimilée dans la littérature à l'équilibre de Nash.

Définition 2.4 Soit un jeu à n joueurs possédant un ensemble d'issues possibles I . La meilleure réponse mr possible donnée par le joueur j_k à la stratégie $a \in A_{j_t}$ d'un joueur j_t est définie par :

$$mr_{j_k}(a) = (\bar{a}^* \in A_{j_k} / u_{j_k}(\bar{a}^*, a) \geq u_{j_k}(\bar{a}, a) \forall \bar{a} \in A_{j_t})$$

Définition 2.5 Soit $i \in I$. i est un équilibre de Nash si et seulement si chaque joueur $j_k \in \{j_1, \dots, j_n\}$ joue sa meilleure réponse face aux stratégies de tous les autres joueurs.

De manière moins formelle, l'équilibre de Nash décrit pour un joueur donné une situation de non-regret vis-à-vis de la stratégie choisie et au vu de celle de ces opposants.

Reprenons l'exemple du dilemme du prisonnier en supposant que nos deux joueurs soient profondément individualistes : seule la durée de leur propre peine leur importe. Le joueur j_1 ne dispose d'aucune information sur les éventuelles stratégies de son adversaire (jeu à information incomplète) ; il va donc chercher à obtenir la peine minimale en se basant sur sa matrice de gains $G_{j_1} = \begin{pmatrix} 2 & 0 \\ 3 & 1 \end{pmatrix}$. Au vu de cette matrice, j_1 choisira de trahir j_2 . De son côté, le joueur j_2 se trouve dans une situation analogue et va donc suivre la même démarche en trahissant j_1 . Au final, les deux joueurs vont donc devoir purger chacun deux ans de prison. Ce premier raisonnement fait état d'un comportement individuel et égoïste de la part des deux joueurs et exhibe ainsi les deux équilibres de Nash du jeu.

Supposons maintenant que nos deux joueurs fassent passer l'intérêt collectif avant leur propre intérêt. Les joueurs vont donc rechercher une solution en utilisant la matrice $G_{j_1} \oplus G_{j_2} = \begin{pmatrix} 4 & 3 \\ 3 & 2 \end{pmatrix}$. Le gain optimal est cette fois-ci obtenu en choisissant pour chacun des joueurs de ne pas trahir son partenaire. Il est intéressant de constater qu'ici la peine de chaque joueur ne sera que d'un an, ce qui constitue une issue Pareto-optimale. Cela montre que privilégier un comportement individualiste ne conduit pas forcément à une satisfaction individuelle optimale, autrement dit tout équilibre de Nash n'est pas forcément Pareto-optimal.

2.3.2 Théorie des jeux et systèmes multi-agents

La question posée à la théorie des jeux dans un contexte multi-agent est : *quelle est la meilleure chose qu'un agent puisse faire ?* [Parsons et Wooldridge, 2000]. En d'autre terme, l'objectif est de rationaliser la prise de décision d'un agent.

2.3.2.1 Application de la théorie des jeux à la coordination

Dans sa thèse, Rosenschein utilise la théorie des jeux pour éviter les communications entre agents lors d'une coordination [Rosenschein, 1985]. Chaque agent utilise cette théorie pour modéliser les décisions prises par les autres agents. Cela permet de prédire ce que les autres vont faire et rend de ce fait les actes communicatifs inutiles. Cette prédiction n'est possible que si l'agent peut s'appuyer sur l'hypothèse selon laquelle les autres agents du système vont agir de manière rationnelle. La rationalité des agents devient donc une condition nécessaire et suffisante pour l'émergence d'une coordination au sein du groupe.

Cette hypothèse forte de rationalité nécessaire pour chaque participant du jeu constitue l'une des grandes limitations de la théorie des jeux en ce qui concerne son application aux phénomènes humains. Cela impose qu'entre deux options, le sujet dont on cherche à modéliser le comportement choisisse celle lui rapportant le gain le plus élevé. Pour diverses raisons, un être humain ne cherchera pas forcément à atteindre un optimum dans sa prise de décision. Dans bon nombre de cas il se contente d'une "bonne" solution qui ne sera pas forcément optimale. Dans la littérature sur les jeux, le terme "satisfaisant" est parfois proposé en remplacement de la notion d'optimum ([Aubin, 1984], chapitre 7).

Une autre limitation à l'utilisation de la théorie des jeux dans le cadre de la coordination multi-agent concerne les connaissances requises et communes à tous les agents. Pour qu'un agent puisse prédire le comportement d'autres agents, il doit être capable d'une part de pouvoir construire une matrice des gains reflétant la situation de jeu dans laquelle les agents se trouvent et d'autre part de connaître leurs préférences vis-à-vis des différentes issues possibles [Nwana *et al.*, 1996].

D'autres limitations à l'utilisation de la théorie des jeux portent sur des aspects pratiques de mise en œuvre et de coûts calculatoires. La taille des matrices qui est fonction du nombre de stratégies possibles, peut constituer un frein pour les applications nécessitant des temps de calculs courts. Enfin, la plupart des travaux dans la littérature utilise des matrices à deux joueurs. Peu de travaux se sont penchés sur le problème de la généralisation à n joueurs.

2.3.2.2 Adéquation de la théorie des jeux avec notre problématique

Pour justifier l'adéquation de la théorie des jeux à la problématique de coordination en carrefour, il nous semble important de resituer et de discuter les différentes limitations énoncées précédemment par rapport à notre contexte.

Les questions que nous nous posons sont les suivantes :

- Est-ce qu'un conducteur fait constamment preuve d'un comportement rationnel durant sa tâche de conduite ?
- Cette rationalité des conducteurs peut-elle se ramener à un problème de maximisation des gains ?

Il semble que l'on puisse répondre par l'affirmative à la première question. Même si les objectifs et les contraintes des conducteurs peuvent être différents suivant la culture, le pays, la psychologie de l'individu, ces conducteurs agissent bien de manière à satisfaire leurs intérêts (qui peuvent être individuels ou collectifs). Nous pouvons ainsi identifier différents types d'objectifs pour lesquels les conducteurs essaient de satisfaire des critères relatifs au temps de parcours le plus court, à la sécurité

dans la circulation et l'exécution de manœuvre, à l'aisance dans le flot de circulation et à la qualité de l'espace occupé, à la facilité de contrôle du véhicule dans le flot de véhicules, etc.

La satisfaction de ces critères peut être exprimée à chaque fois comme une maximisation d'utilités. L'intégration de tous les critères au sein d'un jeu n'est pas possible puisque cela complexifie exagérément la prise de décision. De plus, certains critères sont incompatibles entre eux. Par exemple, il est très difficile de concilier la vitesse (permettant de minimiser le temps de parcours) et la sécurité. Il convient donc de faire un choix sur le ou les critères qui seront pris en considération pour une modélisation à base de jeux.

L'adéquation de la théorie des jeux à notre problématique se justifie également par quelques travaux s'intéressant à la modélisation de situations de trafic sous formes de jeux. Dans [Kita *et al.*, 2001], les auteurs étudient les phénomènes d'insertion. Dans [Bazzan, 2005], l'auteur utilise les techniques d'apprentissage et de jeux évolutionnaires pour proposer un système décentralisé de contrôle de trafic à un niveau macroscopique avec pour objectif l'optimisation des débits des flots de véhicules.

2.3.3 Modélisation

La modélisation présentée ici et le mécanisme de coordination associé décrit en section 2.3.4 sont un condensé des travaux de thèse d'Alexis Champion. Le lecteur désireux d'approfondir cette partie pourra se référer à [Champion, 2003].

2.3.3.1 Les joueurs et les actions

Le mécanisme de coordination, ici résumé, prend en charge le calcul de l'accélération longitudinale dans ARCHISIM. L'accélération latérale est quant à elle déterminée par les algorithmes de changement de voie existants (utilisés pour la conduite en file). Pour proposer une modélisation s'appuyant sur la théorie des jeux, il est nécessaire d'identifier les joueurs et leurs actions. Le choix le plus naturel est de considérer qu'un joueur est un conducteur (agent ou être humain utilisant le simulateur) en approche de l'intersection. La prise de décision repose sur le choix rationnel entre les différentes alternatives, il convient de discrétiser l'accélération longitudinale que l'on cherche à calculer. Le choix fait dans [Champion, 2003] est de considérer les deux actions : *accélérer* et *freiner*.

2.3.3.2 Caractéristiques des jeux

Compte tenu du contexte de la conduite sur route, les jeux utilisés pour modéliser les situations de trafic en carrefour présentent les caractéristiques suivantes :

- Ils sont non coopératifs. Comme nous l'avons évoqué au chapitre 1, la tâche de conduite est plutôt compétitive : chaque conducteur tente de satisfaire ses propres contraintes et objectifs sans forcément tenir compte de ceux des autres automobilistes. Pour les situations critiques en carrefour, le comportement d'un conducteur peut devenir plus coopératif notamment à cause des risques d'accidents ou de blocage obligeant à raisonner de manière plus collective.
- Les jeux sont à information imparfaite et incomplète. En effet, la prise de décision des conducteurs se fait généralement simultanément : pour un joueur il est impossible de savoir quelles

actions les autres joueurs vont choisir. Par ailleurs, chaque agent (conducteur virtuel) de la simulation de trafic ne perçoit que de manière locale la simulation (section 1.3.2.1).

- Les matrices de gains utilisées doivent décrire des jeux à somme non nulle. Même si les jeux sont non coopératifs, les intérêts de deux conducteurs en situation de carrefour ne sont pas forcément diamétralement opposés. Les jeux ne peuvent pas être strictement compétitifs puisque dans un carrefour le déplacement d'un véhicule n'implique pas obligatoirement l'arrêt de tous les autres.
- Enfin, le choix a été fait de ne considérer que des jeux à un seul tour. En effet, les phénomènes de trafic en carrefour sont extrêmement dynamiques, ce qui signifie qu'au cours de la simulation une situation de jeu peut évoluer très vite sur quelques pas de temps. De plus, rappelons que les conducteurs virtuels simulés dans ARCHISIM n'ont absolument aucune mémoire du passé (section 1.3.2.4).

2.3.3.3 Situations de trafic élémentaires

Toute intersection complexe peut se décomposer en une succession de carrefours "élémentaires" en X ou en T [Saad *et al.*, 1989]. Un rond-point se décompose ainsi en autant de carrefours en T qu'il y a de branches. Une situation de trafic en carrefour peut être décrite par l'ensemble des véhicules présents dans ou proche de l'intersection, par leurs positions mais aussi et surtout par les relations de priorités qui les unissent.

Cette notion de priorité est très importante du point de vue de la modélisation puisque c'est sur elle que repose la coordination de flux de trafic en carrefour. Comme nous l'avons vu au chapitre 1, la priorité en carrefour est un concept beaucoup plus large que les règles définies par le Code de la Route. Suivant les conducteurs, leurs cultures, le pays où ils conduisent, elle est le résultat d'une agrégation entre les règles du Code de la Route et d'un ensemble de règles informelles relatives à :

- la priorité par rapport à la vitesse : un conducteur arrivant comme prioritaire à une intersection aura tendance à perdre sa priorité au profit d'un véhicule arrivant au carrefour avec une vitesse élevée,
- la priorité par rapport à l'impatience : un conducteur à l'arrêt depuis un certain temps aura tendance à se considérer comme prioritaire par rapport aux autres,
- etc.

Le prédicat *prio* introduit dans la figure 2.10 doit ainsi être pris au sens large et sa sémantique ne doit pas uniquement se restreindre à la notion de priorité du Code de la Route. Les combinaisons possibles entre deux mobiles x et y partageant une relation de priorités sont au nombre de quatre et correspondent à une situation de trafic élémentaire à deux conducteurs (figure 2.10) :

- x et y traversent le carrefour sans être en conflit : $\neg prio(x, y) \wedge \neg prio(y, x)$,
- x arrive sur la droite de y et est par conséquent prioritaire : $prio(x, y) \wedge \neg prio(y, x)$,
- y arrive sur la droite de x et est donc prioritaire : $\neg prio(x, y) \wedge prio(y, x)$,
- x et y sont de front et veulent tous deux tourner sur leur gauche, ils sont tous deux prioritaires : $prio(x, y) \wedge prio(y, x)$.

2.3.3.4 Matrices de gains

Chacune des quatre situations précédentes fournit une configuration de jeu à laquelle il est possible d'associer une matrice de gains exprimant les vecteurs d'utilité attendue en fonction des actions Go

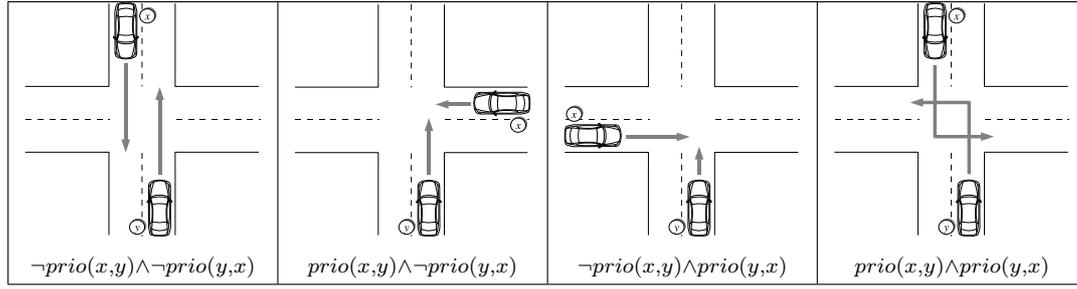


FIGURE 2.10 – Situations élémentaires en carrefour

et *Stop*⁹ des deux joueurs. Ces matrices sont donc de tailles 2×2 et de la forme suivante :

$$\begin{pmatrix} (u_1 = u_x(Go), u_2 = u_y(Go)) & (u_3 = u_x(Go), u_4 = u_y(Stop)) \\ (u_5 = u_x(Stop), u_6 = u_y(Go)) & (u_7 = u_x(Stop), u_8 = u_y(Stop)) \end{pmatrix}$$

Suivant l'une des quatre situations élémentaires considérées, plusieurs simplifications peuvent s'opérer. Par exemple, pour la première situation les paiements de x et y pour une même stratégie sont identiques du fait de la symétrie dans la relation de priorité : $\neg prio(x,y) \wedge \neg prio(y,x)$. Cela donne donc : $u_1 = u_2, u_7 = u_8, u_5 = u_4$ et $u_8 = u_7$. Le même type de raisonnement peut également s'appliquer pour la dernière situation où là encore il y a symétrie dans la relation de priorité : $prio(x,y) \wedge prio(y,x)$.

L'affectation des gains dans les matrices suit les règles suivantes :

- Lorsqu'un joueur choisit la stratégie *Go* sans résoudre le conflit, il va contre son intérêt : la fonction d'utilité doit donc renvoyer un paiement négatif.
- Si à l'inverse le joueur résout son conflit en choisissant l'action *Go* : le paiement associé devient positif.
- Lorsque le joueur freine pour s'arrêter, le conflit est résolu mais cela va à l'encontre de ses objectifs personnels : le paiement est donc nul.

Compte tenu de ces trois règles et des différentes simplifications applicables, les matrices de gains représentant les quatre situations élémentaires sont :

$$\begin{pmatrix} (a_1, a_1) & (a_3, 0) \\ (0, 0) & (0, 0) \end{pmatrix} \quad \begin{pmatrix} (-b_1, -b_2) & (b_3, 0) \\ (0, b_6) & (0, 0) \end{pmatrix} \\ \neg prio(A, B) \wedge \neg prio(B, A) \quad prio(A, B) \wedge \neg prio(B, A)$$

$$\begin{pmatrix} (-b_2, -b_1) & (b_6, 0) \\ (0, b_3) & (0, 0) \end{pmatrix} \quad \begin{pmatrix} (-c_1, -c_1) & (c_3, 0) \\ (0, c_3) & (0, 0) \end{pmatrix} \\ prio(A, B) \wedge prio(B, A) \quad \neg prio(A, B) \wedge prio(B, A)$$

$$a_1, a_3, b_1, b_2, b_3, b_6, c_1, c_3 \in \mathbb{N}^{+*}$$

9. nous considérons cette notation correspondant à l'accélération et au freinage de notre véhicule autonome

2.3.4 Résolution des conflits

2.3.4.1 Situations à n agents

La modélisation précédemment introduite rend compte de situations impliquant deux conducteurs. Dans le cas général, pour un agent x impliqué dans une situation à n agents, il existe potentiellement $\frac{n(n-1)}{2}$ matrices de taille 2×2 chacune correspondant à une relation de priorité entre 2 des n agents.

Pour se coordonner avec les n autres agents, l'agent x ne peut pas considérer toutes les relations de priorités existantes. Plaçons-nous dans le cas d'un conducteur x arrivant à une intersection à quatre branches. Trois autres véhicules y , z et t sont également en approche (figure 2.11).

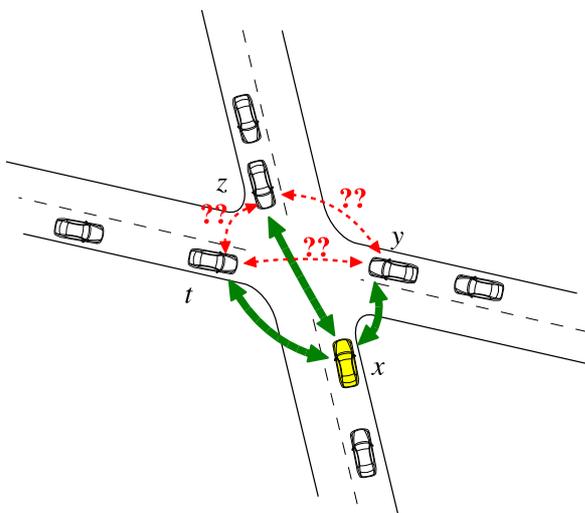


FIGURE 2.11 – Relations entre agents considérées dans le cadre de la coordination en carrefour

En fonction des informations dont il dispose, x peut déterminer sans aucun problème les relations de priorité qui le lient avec y , z et t . En revanche, il est plus difficile pour lui d'évaluer la relation de priorité entre z et t . En effet, si du point de vue du Code de la Route, il peut déduire que z est prioritaire sur t , il n'est pas évident que z et t respectent scrupuleusement le Code pour se coordonner. En fonction de leurs vitesses respectives et de leurs contraintes propres, la relation de priorité utilisée pouvant être : $\neg prio(z, t) \wedge prio(t, z)$. Parmi les $\frac{n(n-1)}{2}$ matrices existantes, seules les $n - 1$ matrices impliquant l'agent x sont donc établies suivant son point de vue.

2.3.4.2 Agrégation des matrices et résolution

Lorsqu'un agent doit se coordonner avec un autre agent par le biais d'une seule matrice de taille 2×2 , la maximisation des gains est relativement simple à opérer. Si ce même agent doit se coordonner avec $n - 1$ autres agents, la maximisation porte alors sur $n - 1$ matrices.

Champion résout cette difficulté en proposant un mécanisme d'agrégation permettant de passer de $n - 1$ matrices 2 joueurs (de taille 2×2) à une seule matrice n joueurs de taille 2^n :

$$G'_{jk} : \begin{array}{l} S_{j_1} \times S_{j_2} \times \dots \times S_{j_n} \mapsto \mathbb{N} \\ (a_{j_1}, a_{j_2}, \dots, a_{j_n}) \longrightarrow \sum_{a_i \in \{a_{j_1}, a_{j_2}, \dots, a_{j_n}\} / a_{j_k}} G_{jk}(a_{j_k}, a_{j_i}) \end{array}$$

Pour la résolution proposée dans le cas de n joueurs, *Champion* se distingue par rapport à ce que l'on peut trouver généralement dans la littérature. La maximisation des gains introduite est purement individualiste. La justification de cette démarche tient en deux points. D'une part pour "coller" à la réalité, il semble plus naturel que les joueurs tentent de maximiser leurs propres gains sans tenir compte des éventuels préjudices qu'ils pourraient infliger aux autres. D'autre part, l'information disponible ($n - 1$ matrices sur $\frac{n(n-1)}{2}$) ne permet pas la mise en place d'une maximisation plutôt collective. La stratégie a^* d'un agent x est basée sur la maximisation suivante :

$$\left(a^* \in S_x / \exists \bar{a} \in S_x \ G'_x(a^*, \bar{a}) = \max_{b \in \{Go, Stop\}} (G_x(b, Go) + G_x(b, Stop)) \right)$$

2.3.4.3 Dynamique du mécanisme de coordination

La modélisation précédemment présentée a permis d'aboutir à un mécanisme de coordination multi-agent dédié à la résolution des conflits en carrefour dans la simulation de trafic ARCHISIM. Il est exécuté à chaque pas de temps par tout véhicule en approche ou à l'intérieur d'un carrefour et calcule une accélération longitudinale qui sera appliquée au prochain pas de temps.

L'algorithme proposé dans [*Champion, 2003*] se décompose en cinq étapes successives (figure 2.12). Pour le détailler, plaçons-nous du point de vue d'un agent x situé à l'entrée d'un carrefour et désirant le traverser pour poursuivre son itinéraire. L'algorithme débute par une recherche des joueurs parmi les différents agents perçus par x (section 1.3.2.1). Ces joueurs sont ceux impliqués dans un conflit direct avec l'agent x .

Pour chacun de ces joueurs identifiés, l'agent x doit essayer d'évaluer la relation de priorité qui les unit. Cette évaluation doit bien entendu tenir compte de la signalisation et de l'infrastructure du carrefour. Ici encore, il s'agit d'une recherche d'éléments de signalisation (feux, panneaux) indiquant la réglementation à laquelle est soumis le carrefour parmi les objets vus par x . A cette première relation de priorité, l'agent x va agréger d'autres types de priorité comme celles relatives à la vitesse ou à l'impatience. La relation finale ainsi obtenue permet de déterminer la situation de jeu dans laquelle l'agent x se trouve avec chaque joueur considéré.

Ces deux premières étapes de l'algorithme définissent le comportement du conducteur. C'est dans la manière de reconnaître sa situation, qu'il est donc possible de donner un comportement cohérent et réaliste à l'agent. Le reste de l'algorithme est de nature calculatoire : il consiste en une suite d'opérations permettant d'agréger ensemble les différentes matrices 2×2 et ainsi rechercher l'action maximisant les gains de l'agent.

La dernière étape de l'algorithme permet de transformer la décision symbolique (*Go* ou *Stop*) en une valeur numérique exprimant l'accélération γ longitudinale future du véhicule. Notons qu'il n'y a pas forcément de correspondance entre le choix de l'action *Go* ou *Stop* et le signe de γ . Autrement dit, le fait de choisir l'action *Go* n'implique pas obligatoirement que $\gamma > 0$. En effet, supposons une situation dans laquelle un conducteur souhaite tourner à gauche dans un carrefour. Compte tenu de

sa vitesse et de l'éloignement des autres véhicules arrivant à sa droite il peut se considérer comme prioritaire. En l'absence d'autres véhicules contraignant arrivant en face de lui, le conducteur choisit l'action *Go*. Dans ce cas précis, l'accélération longitudinale calculée pourra tout de même être négative. En effet, il faut tenir compte de la manœuvre de changement de route du véhicule qui est contraint par une vitesse de passage en courbe. Si la vitesse courante du véhicule est supérieure à celle-ci, γ devra être négative.

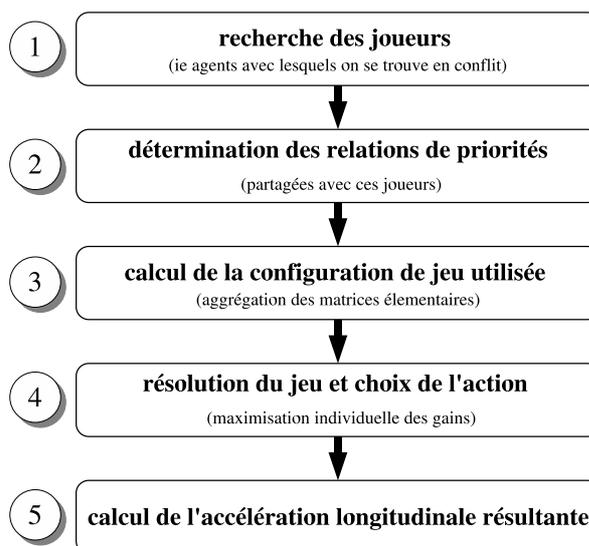


FIGURE 2.12 – Décomposition du mécanisme de coordination utilisé dans ARCHISIM pour la résolution des conflits

La résolution d'un conflit en carrefour se traduit par une succession d'exécutions de l'algorithme présenté ci-dessus. Le nombre d'exécutions est fonction du pas de simulation choisi. Dans ARCHISIM, le pas de simulation se définit par un paramètre exprimé en centième de seconde qui définit la durée d'un pas de temps. Ainsi, plus la discrétisation du temps est petite, plus le nombre de pas de temps nécessaire à la traversée d'un carrefour est grand.

2.3.4.4 Efficacité et complexité du mécanisme

Comme le montre le graphique de la figure 2.13, le mécanisme de coordination offre d'assez bons résultats pour peu que le nombre de joueurs impliqués dans la résolution du conflit reste inférieur à cinq. En effet, passé cette limite, plus de la moitié des relations de priorités n'est plus considérée dans la construction des matrices de jeux. Cette limitation est cependant acceptable puisqu'elle reste en adéquation avec les capacités cognitives d'un être humain qui, en situation de conduite, ne peut gérer les interactions avec plus de quatre autres véhicules en même temps.

De la même manière, la complexité théorique de l'algorithme proposé est exponentielle en nombre d'agents considérés. Là aussi, il convient de relativiser ce résultat théorique puisque pour une résolution de conflits impliquant quatre à cinq agents, les temps de calcul restent tout à fait acceptables.

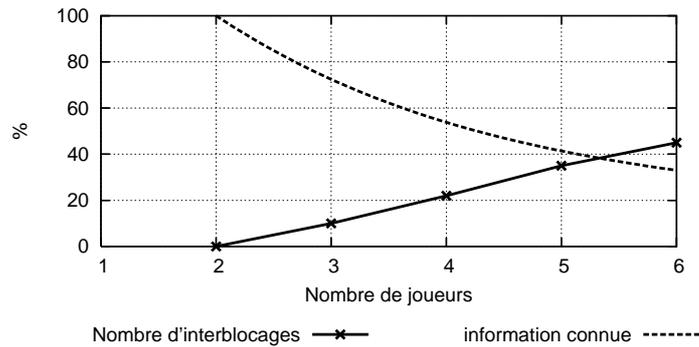


FIGURE 2.13 – Pourcentage de conflits non résolus et d'information connue en fonction du nombre de joueurs [Champion, 2003]

2.3.5 Analyse et critique

Nous terminons cette section et ce chapitre par une analyse du mécanisme de coordination précédemment présenté. Nous dressons en particulier un panorama de ses faiblesses qu'il nous semble important de combler.

2.3.5.1 L'importance du choix des joueurs

Comme nous l'avons souligné précédemment, l'aspect comportemental de la tâche de conduite se situe au niveau du choix des joueurs et de la perception des priorités. L'obtention d'un trafic réaliste passe donc par une reconnaissance réaliste de la situation.

La validation présentée dans [Champion, 2003] montre que les insertions de véhicules dans un flux prioritaire ne sont pas correctes. En particulier les débits simulés sont plus faibles que ceux observés dans la réalité. Ceci illustre qu'un choix erroné dans les joueurs ou une mauvaise perception des relations de priorités peut entraîner des comportements non réalistes, voire une absence de résolution des conflits se traduisant par des accidents et des interblocages.

L'algorithme de sélection des joueurs et de reconnaissance de leurs priorités utilisés dans le mécanisme de coordination nous semble améliorable en plusieurs points. Celui-ci est essentiellement fondé sur l'utilisation d'une relation d'ordre permettant d'ordonner les véhicules perçus par un agent et de faire ainsi ressortir les véhicules les plus contraignants. Cette relation ordonne d'abord les véhicules par branche de carrefour et compare ensuite les temps d'arrivée au conflit. Lorsque les vitesses sont nulles, la comparaison porte sur les distances au conflit. Le choix entre deux véhicules devient difficile dans certaines situations comme lorsque l'un est à l'arrêt et proche du point de conflit et l'autre en mouvement avec une vitesse élevée.

Pour certaines situations, il est difficile d'avoir une reconnaissance qui rend compte avec précision du dynamisme de la situation. Cela peut se traduire par des discontinuités relatives aux choix des joueurs et aux priorités associées. Pour une même situation séparée d'un ou deux pas de temps, les données en entrée de l'algorithme de coordination peuvent être différentes et conduire à l'utilisation de matrices de jeux différentes. Par effet de cause, la décision résultant peut fondamentalement changer. Concrètement, au niveau de la simulation, les accélérations et décélérations successives provoquent une oscillation de la vitesse du véhicule concerné autour d'une valeur médiane. De tels phénomènes

de discontinuité dans la décision et d'oscillations de la vitesse empêchent la convergence vers une résolution du conflit dans lequel l'agent est impliqué.

Il nous semble donc important d'apporter un soin particulier à la manière dont les agents vont reconnaître une situation de trafic. D'une part le réalisme des comportements peut s'en trouver amélioré (aussi bien visuellement que statistiquement). D'autre part, l'instauration d'une certaine stabilité dans cette reconnaissance peut limiter les états d'indécision évoqués ci-dessus.

2.3.5.2 Dynamique de la coordination

Analysons le mécanisme de coordination présenté en première partie en se plaçant du point de vue d'un agent x arrivant à l'approche d'une intersection et désirant la traverser. A chaque pas de temps, l'agent tente de maximiser des gains relatifs à une accélération longitudinale discrétisée sous la forme de deux actions : *Go* et *Stop*. Comme nous l'avons vu, cette maximisation a pour objectif de minimiser le temps de traversée du carrefour.

La dynamique de la coordination, telle que nous l'avons détaillée ci-dessus, fait que cette minimisation du temps de traversée revient pour chaque agent à essayer de maximiser la distance parcourue à chaque pas de temps. Dans [Champion, 2003], l'auteur pose l'hypothèse que maximiser une distance parcourue sur un certain intervalle de temps revient à maximiser la distance parcourue sur chaque pas de temps de cet intervalle. Ceci n'est applicable que pour des situations de trafic indépendantes les unes des autres entre deux pas de temps, ce qui n'est pas forcément toujours le cas. Une solution optimale à un instant t peut être sous-optimale à l'instant $t + 1$. Autrement dit, un agent peut rallonger son temps de présence dans le carrefour en essayant de maximiser sa distance de déplacement dans l'instant.

2.3.5.3 Situation d'interblocages

Nous avons vu que lorsque le nombre de joueurs augmente de manière linéaire, la perte d'informations (liée au fait que certaines relations ne peuvent pas être prises en compte par souci de réalisme) augmente de manière quadratique. Cette perte d'informations conduit inévitablement à l'apparition d'interblocages. Une grande partie de ces interblocages ne perdure pas dans le temps. En effet, si un interblocage apparaît à la fin d'un cycle de jeu, il est très probable qu'il disparaisse au jeu suivant vu que les relations inter-agents ont évolué.

Dans certains cas, les interblocages peuvent s'enchaîner et conduire à un blocage complet du carrefour. Dans d'autres cas, lorsque les véhicules sont longs ou l'espace des carrefours réduit, un seul interblocage peut provoquer le blocage complet du carrefour.

Pour illustrer ces phénomènes, considérons la situation de trafic de la figure 2.14. L'intersection est soumise à la règle de la priorité à droite. Quatorze véhicules sont présents dans le carrefour. Seuls deux d'entre eux (les mobiles 8 et 13) peuvent encore avancer et permettre au carrefour de se vider. La situation de conflit entre les deux véhicules s'exprime par $\neg prio(8, 13) \wedge prio(13, 8)$ du point de vue du mobile 8 et $prio(13, 8) \wedge \neg prio(8, 13)$ du point de vue du mobile 13.

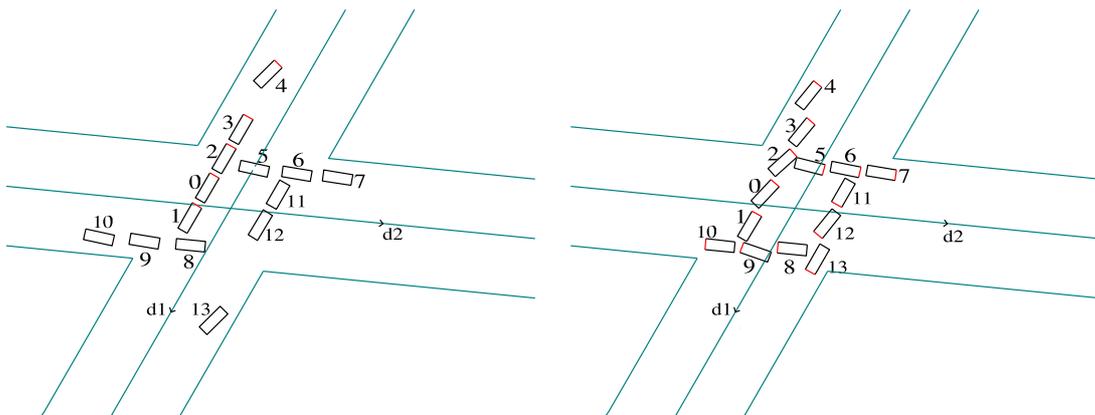


FIGURE 2.14 – Exemple d'interblocage

L'agent 8 construit la matrice correspondant à sa perception de la situation : $G_8 = \begin{pmatrix} -18 & 0 \\ 1 & 0 \end{pmatrix}$, de même pour l'agent 13 : $G_{13} = \begin{pmatrix} -1 & 2 \\ 0 & 0 \end{pmatrix}$ ¹⁰. La maximisation individualiste des gains sur ces deux matrices donne respectivement *Go* et *Stop* comme actions pour les mobiles 13 et 8. Ceci est itéré jusqu'à arriver à la situation de la figure 2.14 où tous les véhicules s'interbloquent mutuellement et ne peuvent plus avancer.

D'autres situations de blocages peuvent se présenter (figure 2.15) lors de stockages au centre du carrefour dûs à un mouvement de double tourne-à-gauche. Il peut s'agir de deux files de véhicules créant une remontée de queue vers les extérieurs du carrefour ou tout simplement deux véhicules un peu longs de type bus ou poids-lourd.

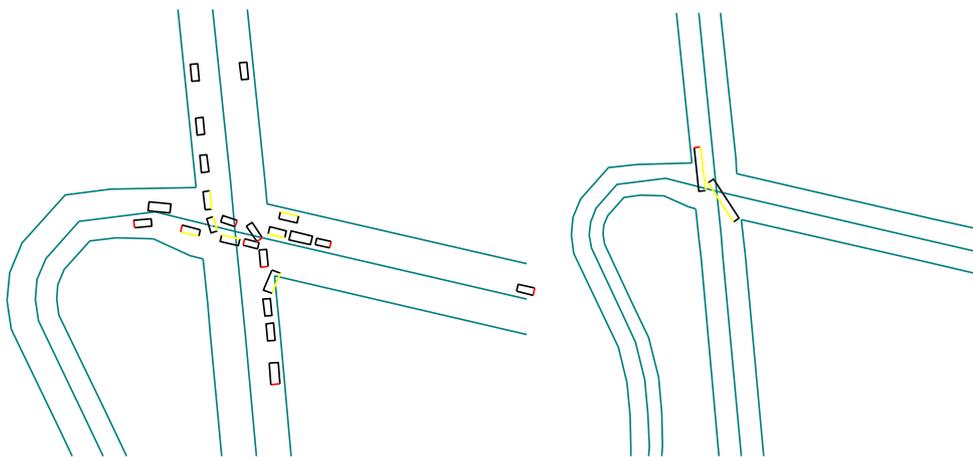


FIGURE 2.15 – Autres exemples d'interblocage

Les situations d'interblocages complets en carrefour¹¹ sont courantes dans la vie de tous les jours notamment dans les grands centres urbains. En règle générale, elles se résolvent par l'intervention d'un agent de la circulation ou grâce à l'intervention d'un ou deux usagers clairvoyants opérant des

10. les valeurs dans les matrices sont tirées des applications numériques présentées dans [Champion, 2003] et sont uniquement données à titre d'exemple

11. toutes les voies de circulation sont mutuellement bloquées et plus aucun véhicule ne peut avancer

manœuvres de marche arrière de manière à amorcer le déblocage des voies de circulation. En simulation et en particulier dans le cadre d'ARCHISIM, notre objectif est d'éliminer de la simulation ces situations de blocage complet de carrefour tout en faisant entrer un nombre de véhicules réaliste. En cela, nous prenons une certaine distance par rapport à la réalité. Ce choix est motivé notamment par l'absence de gestion de la marche arrière dans le modèle de trafic (section 2.2.3.4).

Conclusion

Dans ce chapitre, nous avons présenté les enjeux et les difficultés que représente la simulation de trafic en carrefour et de manière plus générale en milieu urbain. Un état de l'art en ingénierie du trafic a montré que peu de travaux existent et qu'aucune modélisation satisfaisante de la traversée d'intersection n'a été proposée.

Le projet ARCHISIM étant développé dans un cadre multi-agent, nous avons montré que les phénomènes de trafic en carrefour peuvent être vus comme une coordination multi-agent entre véhicules simulés. La seconde partie de ce chapitre a donc été consacrée à une étude des différents mécanismes de coordination en environnement dynamique. Nous nous sommes en particulier intéressés aux mécanismes pouvant être appliqués à la résolution de conflits.

Les différentes contraintes relatives à l'application nous ont permis de justifier en troisième partie l'utilisation d'un mécanisme de coordination spécifique développé pour ARCHISIM. Nous nous sommes attachés à en présenter les principaux éléments. ARCHISIM exploite une modélisation des situations de trafic en carrefour sous formes de jeux. Son analyse nous a permis de pointer ses différentes faiblesses. D'une manière générale, elles se ramènent presque toutes à un manque d'anticipation : soit dans la reconnaissance de la situation, soit dans la prise de décision des agents. Cela se traduit par des comportements parfois peu réalistes (hésitation, oscillation de vitesse) et des interblocages.

L'anticipation est une composante essentielle de la tâche de conduite. Comme le démontrent les travaux de Farida Saad [Saad, 1992], "conduire c'est anticiper". Il semble donc assez naturel d'essayer de fournir des capacités d'anticipation aux conducteurs simulés dans ARCHISIM. Le chapitre 3 est donc consacré à la proposition d'un modèle multi-agent intégrant à la fois coordination et anticipation.

Chapitre 3

Modèles de comportements opportunistes et anticipatifs

Sommaire

3.1 Comportements opportunistes	62
3.1.1 Norme dans les systèmes multi-agents	62
3.1.2 Sélection des agents en amont de la coordination	65
3.1.3 Reconnaissance et perception des priorités	66
3.2 L'anticipation	69
3.2.1 Anticipation et comportement	70
3.2.2 Formalisation de l'anticipation	71
3.2.3 L'anticipation en intelligence artificielle	71
3.2.4 Conclusion	76
3.3 Un modèle d'anticipation fondé sur les réseaux de contraintes	76
3.3.1 Réseaux de contraintes	76
3.3.2 Formalisation de l'anticipation	79
3.3.3 Algorithmes d'anticipation	83
3.3.4 Instanciation du modèle d'anticipation dans le cadre de la simulation comportementale de trafic	86

Nous avons présenté en dernière partie du chapitre 2 un algorithme de coordination permettant de répondre en partie à la problématique de simulation de trafic en carrefour. La modélisation dont cet algorithme est issu et sa dynamique le rendent extrêmement dépendant de la reconnaissance, par les agents, de la situation de trafic à laquelle ils doivent faire face. Les comportements des agents dépendent donc essentiellement de la manière dont ils vont rechercher les autres véhicules avec lesquels ils sont en conflit et la relation de priorité qu'ils vont estimer partager avec ceux-ci.

L'analyse de ce mécanisme de coordination a également montré que de nombreux interblocages pouvaient survenir en cours de simulation. Pour certaines situations de trafic (densité élevée de véhicules, stockage au centre des carrefours, etc), ces interblocages conduisent à un blocage complet du carrefour. Ces phénomènes, que l'on souhaite éviter en simulation, résultent du manque d'anticipation dont les agents font preuve au moment de la coordination.

Dans ce chapitre 3, nous nous focalisons donc sur deux problématiques en amont de la coordination :

- comment reconnaître une situation de trafic afin que les agents aient un comportement réaliste ?
- comment introduire des capacités d’anticipation permettant aux agents d’éliminer de leur plage de décision les actions potentiellement interbloquantes ?

Dans la première section, nous abordons la perception des relations de priorité par les agents en approche d’un carrefour. Un conducteur humain ne respectant pas toujours les règles du Code de la Route, il nous semble pertinent que les agents puissent faire de même. Dans un contexte multi-agent, cela peut se formuler comme un problème de gestion de la norme du comportement dans le système. Nous détaillons donc une modélisation de la perception des priorités permettant aux agents de faire preuve d’un comportement que nous définissons comme opportuniste.

Les deux autres sections de ce chapitre sont consacrées à l’anticipation. Bien que cette composante essentielle du raisonnement cognitif humain soit souvent citée dans la littérature multi-agent, peu de travaux significatifs ont été publiés dans ce domaine. Nous proposons donc un modèle formel de l’anticipation inspiré du principe des réseaux de contraintes. Cette approche originale est présentée de manière générique, puis instanciée dans le cadre de la simulation de trafic.

3.1 Comportements opportunistes

Un de nos objectifs est la reproduction réaliste de comportements humains au sein du système de trafic. Or ce système, comme beaucoup d’autres, est soumis à un ensemble de règles et de normes communément appelés Code de la Route. Dans le cadre des systèmes multi-agents, la gestion de la norme est une problématique à part entière.

3.1.1 Norme dans les systèmes multi-agents

3.1.1.1 Généralités

Au sens commun du terme, une norme désigne un ensemble de règles et principes appliqués à une entité (un objet, une personne, ...) et décrivant comment cette entité doit être et doit se comporter. Le principe de norme est donc souvent utilisé pour décrire et obtenir l’état idéal d’un système : système judiciaire, système de trafic, etc.

Les normes sont utilisées dans des domaines variés. En sociologie par exemple, l’étude des sociétés ou des groupes de personnes soumis à une autorité montre que les relations entre les individus sont régies par des normes pouvant être explicites ou implicites. Les systèmes judiciaires dans les sociétés modernes établissent ainsi des lois permettant de préserver les droits des citoyens. Dans le champ des applications industrielles, les normes regroupent différentes données (mesures physiques, critères de qualité, etc) et processus de fabrication permettant d’améliorer la production.

Une norme n’a aucun sens sans une autorité. Le rôle de celle-ci est double. D’une part, l’autorité doit formuler, en accord avec ses objectifs, l’ensemble des règles définissant la norme. D’autre part, l’autorité doit veiller à la bonne application de la norme. Ceci implique le contrôle et l’application de pénalités en cas de violation.

Pour résumer, une norme est utilisée pour garantir certaines propriétés d'un système à un niveau collectif. Le principe de norme vient donc en opposition avec les buts, les motivations et les désirs exprimés au niveau individuel. Cette opposition est la principale origine des phénomènes de violation de normes.

3.1.1.2 Normes et agents

Dans les systèmes multi-agents, les normes sont généralement utilisées pour spécifier ce qu'un agent doit faire. En d'autres termes, les normes fournissent la manière de coordonner un ensemble d'agents par une forme de régulation. Dans [Shoham et Tennenholtz, 1995], les auteurs utilisent le concept de lois sociales de manière à contraindre les actions motivées par des buts antagonistes. En restreignant les moyens de satisfaire les objectifs individuels des agents, ces lois sociales permettent de garantir des propriétés de coopération au niveau collectif. L'application présentée par les auteurs met en jeu un ensemble de robots se déplaçant dans un environnement inconnu. Dans ce contexte, les lois sociales s'apparentent à une sorte de Code de la Route guidant le déplacement des robots.

Dans [Minsky et Ungureanu, 1997], les auteurs introduisent un système multi-agent basé sur la notion de règlement. Sous ce terme "règlement" les auteurs identifient trois éléments : les actions guidées par le règlement, l'ensemble des agents participant à cette réglementation et enfin les principes destinés à guider les actions des agents. Sur la base de ce triptyque, les auteurs formalisent la coordination par réglementation. Les lois relatives à une réglementation sont exprimées sous la forme de règles situations / actions. Ces lois portent sur le traitement des messages échangés entre les agents et permettent de réguler les deux types d'évènements que sont l'émission et la réception.

D'autres travaux étendent des approches classiques en systèmes multi-agents. Par exemple, dans [Dignum *et al.*, 2000], les auteurs complètent le modèle BDI¹² de manière à ce que les agents puissent considérer la norme comme l'une de leur motivation. Le modèle ainsi proposé introduit un nouveau type d'évènement : le résultat de l'application d'une norme.

Quelques recherches se focalisent sur la manière de concevoir les normes d'un système multi-agent. Les travaux de Boella par exemple, étudient la formalisation d'un système normatif. Dans [Boella et Lesmo, 2002], les auteurs montrent que dans l'optique d'un système normatif modélisé comme un système multi-agent, les relations que les agents entretiennent avec la norme peuvent être modélisées comme des jeux à deux joueurs.

Tous ces travaux ainsi que ceux de [Aldewereld *et al.*, 2005] et [Barbuceanu *et al.*, 1998] utilisent donc la norme pour définir un comportement standard des agents face à une situation spécifique. Le terme d'agent normatif correspond à un agent agissant en accord avec les normes spécifiées pour le système dans lequel il évolue. Par abus de langage, nous pouvons dire qu'un agent normatif est un agent dont le comportement est codé avec les normes du système.

3.1.1.3 Norme vs autonomie

L'autonomie est l'une des facettes les plus intéressantes d'un agent. Elle exprime la capacité à agir sans contrôle externe et semble donc être en opposition avec la notion de norme précédemment détaillée.

12. Belief/Desire/Intention

La violation et le non-respect des normes dans un système multi-agent ont fait l'objet de quelques travaux. Dans [Boella et van der Torre, 2002], les auteurs dressent une liste de situations décrivant un contexte favorable à la violation de normes mais sans présenter d'application réelle. D'une manière générale, le concept de violation de normes dans les systèmes multi-agents est rarement mis en œuvre dans les travaux issus de la littérature.

La violation de la norme présente pourtant de nombreux intérêts. D'un point de vue comportemental, la violation de la norme sous certaines conditions et dans certaines situations semble plus réaliste que le strict respect de l'ensemble des règles qui la composent. Cela est d'autant plus vrai lorsque l'objectif est de reproduire des comportements humains.

La violation semble être une manière de concilier norme et autonomie au sein d'un même système. Dans [Carabelea et Boissier, 2005], les auteurs notent qu'un des aspects de l'autonomie concerne le degré d'influence de la norme du système sur la prise de décision individuelle de l'agent.

3.1.1.4 Norme dans le contexte de la simulation de trafic

Le déplacement de conducteurs humains sur les routes est régi par un ensemble de règles et de relations de priorités : le Code de la Route. Le système de trafic pourrait donc être considéré comme un système normatif. L'observation des comportements réels montre pourtant qu'un conducteur réel ne suit pas toujours les règles du Code de la Route (section 2.1.1.3) suivant le contexte, les conditions de circulation, etc.

Dans [Bou *et al.*, 2006], les auteurs étudient la norme dans le contexte des intersections au sein d'une institution électronique autonome (IEA). Une institution électronique [Esteva *et al.*, 2001] peut se définir comme un environnement virtuel permettant la régulation des interactions entre les agents participant à l'institution. Les auteurs présentent une extension de ce modèle permettant à l'institution d'être autonome et d'adapter les normes qui la régissent en fonction des circonstances. Leur application montre qu'une IEA en charge du contrôle de trafic à un carrefour est capable d'adapter sa norme (c'est-à-dire les règles régissant le trafic au sein du carrefour) de manière à diminuer le nombre d'accidents. Ces travaux montrent donc que la gestion de la norme est essentielle dans un contexte de simulation de trafic.

Dans la section 2.1.1 consacrée à la modélisation de la tâche de conduite, nous avons présenté différents résultats en psychologie de la conduite qui montrent qu'un conducteur humain va jusqu'à développer ses propres règles informelles du Code de la Route [Björklund et Åberg, 2005]. Ces règles sont souvent contextuelles et particulières à une situation donnée. Si l'on met ces résultats en parallèle à ceux de Saad [Saad, 1992] ou [Summala, 2005] montrant que la minimisation du risque est une composante forte des motivations des conducteurs, on peut penser que la violation des règles du Code de la Route ne se fait pas sans une hausse de l'attention et de l'activité intellectuelle permettant de compenser la prise de risque.

Il nous semble donc plus approprié de parler de comportement opportuniste plutôt que de comportement réellement non normatif. Prenons l'exemple d'un véhicule arrêté à un Stop. Le conducteur peut détecter, dès son arrivée au panneau, un créneau lui permettant de s'insérer dans le flux de trafic. Suivant la densité de la circulation, il peut estimer un temps d'attente relativement long avant le prochain créneau et décider de redémarrer sans avoir attendu les trois secondes réglementaires dictées par le Code de la Route. Le non-respect de la norme est dans ce cas la conséquence d'un comporte-

ment que l'on peut qualifier d'opportuniste. Nous définissons donc un comportement opportuniste¹³ comme un comportement circonstancié subordonnant la norme d'un système à l'intérêt momentané de l'agent.

Compte tenu de la prise de décision dans ARCHISIM et de la dynamique du mécanisme de coordination présentées au chapitre 2, une manière simple d'introduire de tels comportements dans la simulation est d'agir sur la perception des agents.

3.1.2 Sélection des agents en amont de la coordination

Le choix des agents avec lesquels se coordonner conditionne la prise de décision d'un conducteur simulé. Ce choix doit être pertinent suivant le contexte de la situation de manière à garantir des comportements réalistes.

Nous présentons ici un critère de sélection des agents fondé à la fois sur leur position par rapport aux points de conflit détectés et sur leur perception des priorités [Doniec *et al.*, 2006a].

3.1.2.1 Points de conflit

Pour définir la notion de points de conflit, considérons deux agents x et y placés en situation de franchissement d'intersection. Pour ce couple de mobiles (x, y) il est possible d'associer un point de conflit correspondant à l'intersection de leurs trajectoires.

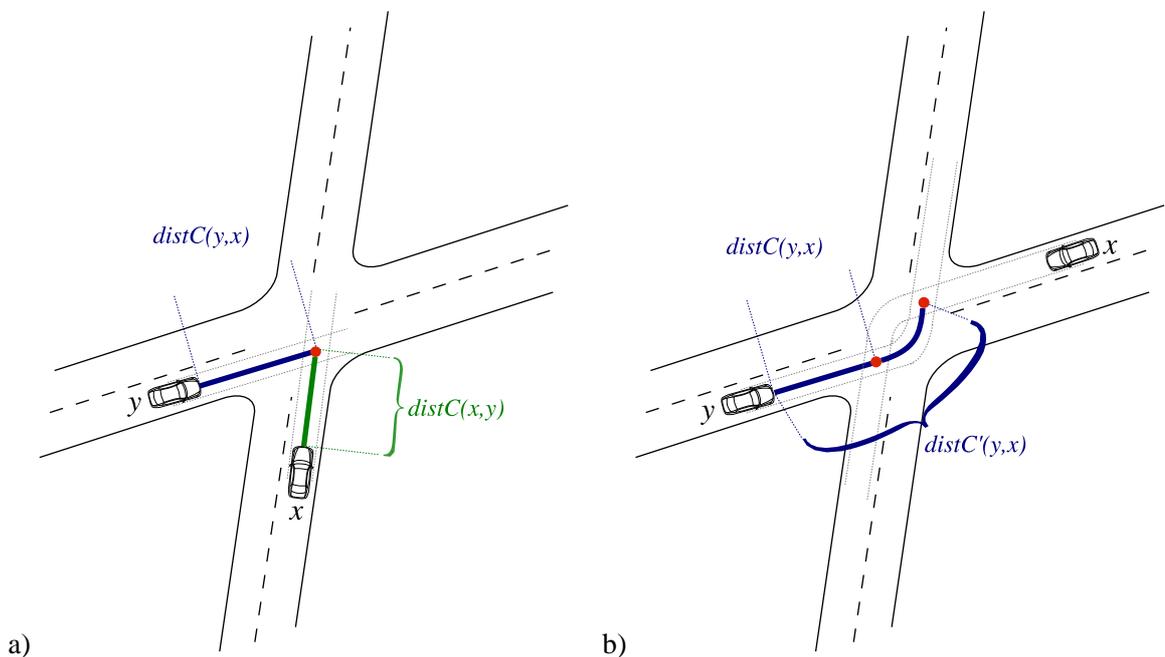


FIGURE 3.1 – Distances aux conflits en carrefour

13. le terme d'opportuniste ne fait ici en aucun cas référence aux travaux de Barbara Hayes-Roth associant deux modes de contrôle de l'action : modèles par planification et modèles réactifs [Hayes-Roth, 1993]

Pour chaque point de conflit entre deux mobiles x et y , nous pouvons considérer deux distances (figure 3.1.a) :

- $distC(x, y)$ qui exprime la distance entre le mobile x et le point de conflit partagé avec le mobile y
- $distC(y, x)$ la distance duale entre le mobile y et le point de conflit partagé avec x

Le point de conflit entre deux mobiles x et y n'existe pas toujours. C'est par exemple le cas pour deux mobiles situés sur une même route, de part et d'autres de l'axiale et ne changeant pas de direction au carrefour. Lorsque le point de conflit existe, son unicité n'est pas garantie. Dans le cas d'un double tourne-à-gauche, il existe potentiellement deux points de conflit (figure 3.1.b). La levée de cette ambiguïté peut se faire par un calcul de temps au conflit qui consiste à diviser les distances au conflit par la vitesse courante du véhicule concerné. L'unicité du point de conflit peut alors être obtenue en recherchant le point accessible dans la même fenêtre de temps pour les deux mobiles.

3.1.2.2 Algorithme de recherche des agents

La sélection des agents se fait par une recherche en deux étapes. La première étape pour chacun des points de conflit p_i consiste à ordonner les véhicules y_{p_i} du plus contraignant au moins contraignant du point de vue de la relation de priorité entre x et les y_{p_i} . La seconde étape consiste à ordonner tous les points de conflit p_i se trouvant sur la trajectoire d'un conducteur simulé x du plus proche au plus éloigné.

Pour deux mobiles y_{p_i} et y'_{p_i} ayant une distance au conflit avec x égale : $distC(x, y) = distC(x, y')$, le mobile choisi est le plus proche du point de conflit : $min(distC(y, x), distC(y', x))$.

Pour résumer, la sélection des agents repose sur l'application de la relation d'ordre \prec suivante :

$$(y \prec y') \equiv \begin{cases} prio(x, y') \wedge \neg prio(x, y) \\ distC(x, y) \leq distC(x, y') & \text{si } prio(x, y) \wedge prio(x, y') \\ distC(y, x) \leq distC(y', x) & \text{si } distC(x, y) = distC(x, y') \end{cases}$$

La recherche des agents dépend donc de la perception des priorités. Nous détaillons cet aspect dans la suite.

3.1.3 Reconnaissance et perception des priorités

Dans [Champion, 2003], différents types de priorité sont considérés. L'agrégation entre ces priorités permet d'obtenir la relation de priorité finale $prio$ utilisée en entrée de l'algorithme de coordination (chapitre 2). Cette agrégation conditionne en grande partie les comportements locaux obtenus et donc le réalisme de la situation de trafic émergente. Dans la partie validation de [Champion, 2003], il apparaît que le modèle de comportement des conducteurs ne permet pas de simuler correctement la traversée et l'insertion en carrefour d'un flux non prioritaire. Nous proposons donc dans cette section un ensemble de règles d'agrégation permettant d'améliorer les travaux existants.

3.1.3.1 Différents types de priorité

Nous exprimons la priorité relative au Code de la Route par le prédicat *prioCode*. Un comportement normatif de conducteur simulé en carrefour peut ainsi être défini simplement par la règle :

$$prioCode(\alpha, \beta) \rightarrow prio(\alpha, \beta)$$

L'approche que nous utilisons considère la relation de priorité entre deux agents comme résultant à la fois de la priorité relative au Code mais aussi de règles de conduite informelles et de pratiques individuelles couramment observées dans la réalité [Björklund et Åberg, 2005] :

- la priorité par rapport à la vitesse : un conducteur arrivant comme prioritaire à une intersection aura tendance à perdre sa priorité au profit d'un véhicule arrivant au carrefour avec une vitesse élevée,
- la priorité par rapport à l'impatience : un conducteur à l'arrêt depuis un certain temps aura tendance à se considérer comme prioritaire par rapport aux autres,
- la priorité par rapport à l'engagement : un véhicule déjà engagé sur la zone de conflit sera perçu comme prioritaire.

Nous détaillons dans la suite la prise en compte de ces différentes priorités dans le modèle de trafic d'ARCHISIM.

3.1.3.2 Priorité par rapport à l'impatience

De manière à prendre en compte la notion d'impatience dans notre définition d'un comportement opportuniste, nous introduisons les prédicats suivants :

- *impatience* qui indique l'état d'impatience de l'agent,
- *redemarrage* qui précise si l'agent est en phase d'accélération après une période d'arrêt.

L'état d'impatience d'un agent est activé lorsque celui-ci est arrêté pour une durée supérieure à une certaine limite de temps faisant partie de ses caractéristiques propres.

Pour illustrer nos propos, plaçons-nous dans la situation d'un mobile x arrêté à un stop (figure 3.2) et devant déterminer ses relations de priorités avec les mobiles y et y' de l'axe principal. Trois règles du Code de la Route peuvent s'appliquer à cette situation :

- x doit marquer un arrêt d'au moins trois secondes,
- x doit laisser la priorité aux flux arrivant à sa droite et à sa gauche,
- x ne doit pas entrer dans le carrefour s'il n'est pas sûr d'en sortir.

Lorsque le trafic à l'intérieur du carrefour est relativement dense, le respect de ces trois règles conduit à des temps d'attente au Stop extrêmement longs par rapport à ce qui est observable dans la réalité. Nous modulons donc la perception des priorités autour de l'impatience de l'agent. Lorsque le mobile y est également à l'arrêt dans le carrefour (ou avec une vitesse très faible), l'agent x se considère comme prioritaire s'il est impatient et plus près que y du point de conflit :

$$\neg prioCode(x, y) \wedge prioCode(y, x) \wedge impatience(x) \\ \wedge (distC(x, y) < distC(y, x)) \rightarrow prio(x, y) \wedge \neg prio(y, x)$$

Si le mobile y est en mouvement, la distance au conflit n'a plus vraiment de sens et il est préférable de comparer des temps au conflit. Le véhicule x étant à l'arrêt, son temps au conflit doit être calculé

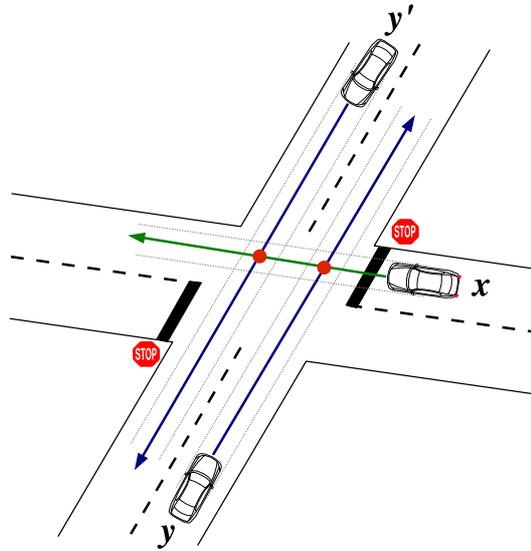


FIGURE 3.2 – Franchissement de Stop et choix des agents pour la coordination

avec une accélération γ_x^* estimée dans le cas où x prendrait la décision d'avancer. Ce temps est donné par la formule cinématique : $\sqrt{\frac{2 * distC(x,y)}{\gamma_x^*}}$. Le temps au conflit pour y peut être obtenu simplement à partir de sa vitesse instantanée V_y sans tenir compte de son accélération :

$$\neg prioCode(x,y) \wedge prioCode(y,x) \wedge impatience(x) \wedge \left(\sqrt{\frac{2 * distC(x,y)}{\gamma_x^*}} < \frac{distC(y,x)}{V_y} \right) \rightarrow prio(x,y) \wedge \neg prio(y,x)$$

D'autres règles de la même forme peuvent être utilisées en considérant non plus l'impatience des agents mais leur redémarrage après un temps d'arrêt. Le prédicat associé au redémarrage d'un véhicule reste vrai pendant une certaine période de temps, jusqu'à ce que le mobile ait atteint une certaine vitesse seuil :

$$\neg prioCode(x,y) \wedge prioCode(y,x) \wedge redemarrage(x) \wedge \left(\sqrt{\frac{2 * distC(x,y)}{\gamma_x^*}} < \frac{distC(y,x)}{V_y} \right) \rightarrow prio(x,y) \wedge \neg prio(y,x)$$

Comme nous le verrons au chapitre 5, l'utilisation de ce prédicat garantit une certaine stabilité de la reconnaissance et permet d'éviter les oscillations de perception sur plusieurs pas de temps.

3.1.3.3 Priorité relative à la vitesse

La prise en compte de la priorité par rapport à la vitesse se fait sur la base de calculs de temps au conflit et de distances d'arrêt. Prenons le cas d'un véhicule x approchant d'une intersection et prioritaire sur les autres véhicules. Si parmi ceux-ci, un véhicule y est en approche du carrefour avec une vitesse élevée, la règle suivante peut être appliquée :

$$\begin{aligned} & prioCode(x, y) \wedge \neg prioCode(y, x) \wedge \left(\frac{distC(y, x)}{V_y} < \frac{distC(x, y)}{V_x} \right) \\ & \wedge \left(distC(y, x) < -\frac{V_y^2}{2\gamma_y} \right) \rightarrow \neg prio(x, y) \wedge prio(y, x) \end{aligned}$$

Dans cette règle, $-\frac{V_y^2}{2\gamma_y}$ exprime la distance d'arrêt du véhicule y . Dans le cas où le véhicule y arrive au point de conflit avant x , si la distance d'arrêt du véhicule y est supérieure à la distance au conflit alors x se considère comme non prioritaire.

La prise en compte du temps au conflit peut également permettre d'éliminer certaines relations de priorité superflues. Reprenons le cas du véhicule x à l'arrêt à un Stop (figure 3.2). Si par exemple, son temps d'arrivée au conflit est largement supérieur à celui du véhicule y' , il n'a pas de raison de considérer y' comme prioritaire. La règle qui traduit cette agrégation de priorité est la suivante :

$$\begin{aligned} & \neg prioCode(x, y') \wedge prioCode(y', x) \\ & \wedge \left(\frac{distC(x, y')}{V_x} \geq \frac{distC(y', x)}{V_{y'}} + \delta_s \right) \wedge (\gamma_{y'} \geq 0) \rightarrow \neg prio(x, y') \wedge \neg prio(y', x) \end{aligned}$$

δ_s représente ici une marge de sécurité que x s'accorde pour lever le conflit avec y . Le test sur l'accélération de y permet de réduire les oscillations de perception sur plusieurs pas de temps successifs : si y est en train de freiner pour une raison quelconque x doit continuer à le considérer comme un véhicule conflictuel.

De nombreuses autres règles composent cette agrégation de priorité, toutes les détailler ici ne ferait qu'alourdir notre exposé. Nous invitons donc le lecteur intéressé à se référer à l'article [Doniec *et al.*, 2006a] pour une présentation des autres règles.

3.1.3.4 Comportements opportunistes et interblocages

Les apports du comportement opportuniste en terme de réalisme seront illustrés au chapitre 5. Pour le moment, on ne peut que constater que l'introduction de tels comportements augmente encore l'apparition d'interblocages. En effet, elle favorise le stockage des agents et donc le remplissage du centre des carrefours.

Un comportement opportuniste doit donc aller de pair avec un comportement anticipatif permettant aux agents de compenser le non-respect de certaines règles par un raisonnement plus complexe. La suite de ce chapitre est dédiée à la proposition d'un mécanisme d'anticipation permettant aux agents de prendre en compte les conséquences de leurs actions et d'éviter les phénomènes de blocage.

3.2 L'anticipation

La notion d'anticipation telle que nous l'entendons fait référence à une capacité de l'esprit qui vise à émettre un ensemble de prévisions à partir de données du présent [Hachette, 2006]. Le sens

du terme anticipation est parfois étendu à tout mouvement de la pensée permettant de se représenter un évènement futur [Le Petit Robert, 2006] dont la probabilité de réalisation est supérieure à celle du rêve et dont le délai de réalisation est souvent relativement proche.

L'étymologie du mot "anticipation" remonte au XV^{ème} siècle où il définissait le fait d'agir avant, de prendre les devants. Il s'agissait donc de s'adapter face à un futur probable. Depuis le début du XX^{ème} siècle, de nombreuses études menées en sciences du vivant et sciences humaines ont cherché :

- à mettre en évidence l'anticipation dans les comportements
- à formaliser cette capacité cognitive des êtres vivants à travers différents modèles et à expliquer les différents mécanismes qu'elle met en œuvre.

3.2.1 Anticipation et comportement

Le concept d'anticipation couvre de nombreuses disciplines : biologie, psychologie, épistémologie, etc. De nombreux travaux ont donc investigué la capacité d'êtres vivants à prévoir les évènements futurs et leurs conséquences.

Au delà du sixième sens souvent attribué aux animaux capables de sentir les situations de danger, de nombreuses études sur le comportement animal ont permis de mettre en avant de réelles capacités d'anticipation chez certaines espèces. Tolmann fut l'un des premiers à prendre en compte ces capacités en introduisant la notion d'*effets* dans le modèle "stimulus/réponse" utilisé jusque là pour étudier les comportements animaux [Tolman, 1932]. Seward met en évidence l'existence d'une modélisation interne de l'environnement chez les rats se déplaçant dans un labyrinthe [Seward, 1949].

Dans le domaine de la psychologie cognitive, Hoffman propose une théorie du contrôle du comportement s'appuyant sur une boucle d'évaluation permanente des prédictions. Il associe à tout comportement une anticipation des effets attendus à partir d'un historique de situations analogues déjà rencontrées. Chaque prédiction de ces effets est confrontée aux effets réels obtenus de manière à évaluer la pertinence de l'anticipation en fonction de la situation [Hoffmann, 1993].

L'anticipation peut se décliner en plusieurs niveaux de complexité. Dans [Riegler, 2001], l'auteur introduit plusieurs types d'anticipation en se basant sur l'observation des comportements animaux. Il définit l'anticipation :

- *innée* reposant sur des schémas phylogénétiques acquis dès la naissance,
- *émotionnelle* guidée par des instincts propres à l'individu,
- *intelligente* basée sur une capacité à s'imaginer un futur plus ou moins éloigné.

Dans les domaines des systèmes artificiels, il est souvent considéré que l'anticipation est un acte *intentionnel* [Ekdahl, 1999], ce qui tend à éliminer dans le cadre de notre problématique l'anticipation innée et l'anticipation émotionnelle plutôt rencontrées dans les systèmes vivants et naturels.

En dehors de ces différentes considérations philosophiques, différents chercheurs se sont penchés sur une formalisation de l'anticipation mettant en avant les mécanismes et structures sous-jacentes. Comme le souligne [Riegler, 2001], la définition d'un cadre mathématique et/ou informatique approprié pour l'anticipation reste un challenge pour la recherche.

3.2.2 Formalisation de l'anticipation

Dans la littérature, la première formalisation de l'anticipation est généralement attribuée au biologiste Robert Rosen. Dans [Rosen, 1985], il introduit en particulier la définition suivante :

"Un système anticipatif est un système qui contient un modèle prédictif de lui-même et/ou de son environnement lui permettant de changer son état courant en fonction d'un ensemble de prédictions du futur"

Le terme "système" est ici à prendre au sens large et peut ainsi désigner un être humain, un animal, un robot ou un agent logiciel. Cette définition d'un système anticipatif fait aujourd'hui référence dans la littérature et exprime le lien entre connaissances du futur et prise de décision à l'instant présent. Elle décrit une capacité particulière d'un système à adapter son état courant par rapport à une vision du futur. En cela, la définition de Rosen marque clairement la distinction entre anticipation et prédiction : l'anticipation sous-entend des mécanismes bien plus complexes que la simple capacité à se projeter dans le futur.

L'approche de Rosen peut être qualifiée de constructiviste puisqu'elle repose sur la construction d'une représentation mentale. Rosen pose donc le postulat que l'anticipation résulte d'une relation de modélisation permettant de passer d'un système à son modèle prédictif. Certains auteurs comme Ekdahl discutent autour du terme "représentation mentale" en distinguant *description* et *modèle* [Ekdahl, 2000].

Pour notre part, nous considérons une définition de l'anticipation proche de celle de Rosen :

Définition 3.1 *L'anticipation d'une entité artificielle est un processus de raisonnement en deux phases successives : une phase de prédiction de l'état futur du système et une phase d'analyse de ces prédictions permettant à cette entité d'adapter son comportement.*

3.2.3 L'anticipation en intelligence artificielle

Ayant été identifiée par les psychologues et les biologistes comme un élément essentiel du raisonnement humain, l'anticipation a été tout naturellement étudiée en intelligence artificielle. Ces travaux sur l'anticipation sont relativement récents, nous en présentons ici quelques uns.

3.2.3.1 Classification et niveau d'anticipation

Dans le cadre de l'intelligence artificielle, l'anticipation reste sujette à discussion. Il est ainsi difficile de comparer entre eux les travaux abordant ce concept puisque le niveau d'anticipation considéré varie d'un modèle à un autre.

Dans [Butz *et al.*, 2003], les auteurs dressent un état de l'art de l'anticipation dans les systèmes adaptatifs par apprentissage. Ils identifient quatre classes de systèmes anticipatifs en fonction du niveau d'anticipation mis en jeu (figure 3.3).

La première classe est définie par les auteurs comme *anticipation implicite*. Ils considèrent comme une forme d'anticipation le fait qu'un comportement ait été programmé pour répondre à des objectifs bien précis dans le futur. Dans [Picard et Gleizes, 2005] par exemple, les auteurs utilisent des règles d'anticipation spécifiant un comportement bien précis au regard de certaines situations. Cette anticipation qu'ils qualifient de réactive est utilisée dans un contexte de coopération entre robots autonomes et permet d'éviter les situations de blocage.

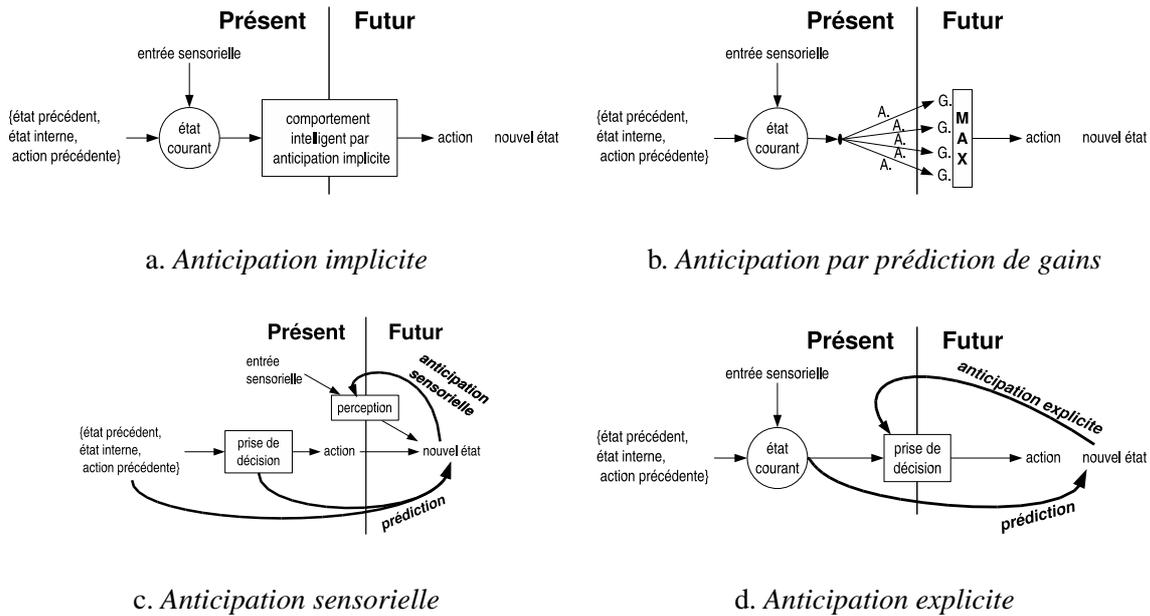


FIGURE 3.3 – Niveau d’anticipation dans les systèmes artificiels [Butz *et al.*, 2003]

La seconde classe regroupe tous les systèmes dont la prise de décision est basée sur une prédiction de gains. Un tel système estime ainsi le bénéfice attendu pour chaque action possible et utilise une fonction de maximisation comme critère de décision. La phase d’analyse de prédiction telle que nous l’avons introduite dans notre définition de l’anticipation est ici réduite à sa plus simple expression puisqu’elle n’influence pas le comportement. L’algorithme de coordination utilisé dans ARCHISIM et présenté au chapitre 2 (section 2.3.4) se positionne clairement dans cette classe.

Dans la troisième classe, qualifiée d’*anticipation sensorielle*, les prédictions ont une influence sur le comportement. Plus précisément, elles peuvent modifier les processus de perception du système. La prise de décision quant à elle reste uniquement guidée par l’état et les actions antérieures du système.

La quatrième et dernière classe est présentée par les auteurs comme le niveau d’anticipation le plus intéressant. Elle regroupe tous les systèmes utilisant une représentation explicite des états futurs (pouvant être partielle ou totale), d’où le qualificatif d’*anticipation explicite*. Ici l’analyse des prédictions influence directement la prise de décision.

Du point de vue de notre définition de l’anticipation, les deux premières classes n’exhibent pas vraiment de capacité anticipative. La proposition que nous formulons dans la suite (section 3.3.2) s’inscrit dans la quatrième classe, celle des systèmes à anticipation explicite.

3.2.3.2 Anticipation et apprentissage

Beaucoup de travaux sont fondés sur des études menées en psychologie démontrant que l’anticipation peut être associée à l’apprentissage. Dans [Marshall *et al.*, 2004], les auteurs utilisent l’anticipation pour générer un signal de renforcement lors de l’apprentissage et de l’entraînement de robots. Dans cette approche la prédiction d’erreurs et l’apprentissage pour diminuer le nombre d’erreurs se font simultanément, ce qui place cette approche dans la classe "anticipation explicite". Dans

[Blank *et al.*, 2005], les auteurs montrent que l'anticipation peut accélérer l'apprentissage pour certains types de problèmes en robotique.

De nombreuses approches sont également proposées dans le cadre des systèmes de classeurs. Un système de classeurs est composé d'un ensemble de règles (classeurs) exprimées sous la forme d'un triplet (*condition, action, force*). Le déclenchement d'une règle se fait par appariement entre sa condition et la situation courante.

Dans [Stolzmann, 1998], l'auteur reprend les travaux d'Hoffmann sur l'anticipation et propose le formalisme ACS (Anticipative Classifier Systems). Il ajoute au triplet de base formant un classeur une partie *effet* destinée à décrire les changements occasionnés par le déclenchement de la règle. Le champ *force* y est exprimé comme un couple représentant la qualité de l'anticipation et le gain espéré. Plus les effets énoncés dans le classeur sont effectivement observés, plus la qualité de l'anticipation sera élevée.

La prédiction de la situation future se fait par application d'un opérateur binaire entre les attributs de la situation courante et ceux de la partie *effet* (figure 3.4). Pour un ensemble de classeurs pouvant être déclenchés, le système choisit celui présentant la plus grande qualité d'anticipation. Dans le cas où la situation prédite par le classeur n'est pas vérifiée, un mécanisme d'apprentissage se charge de diminuer la qualité de l'anticipation associée au classeur utilisé.

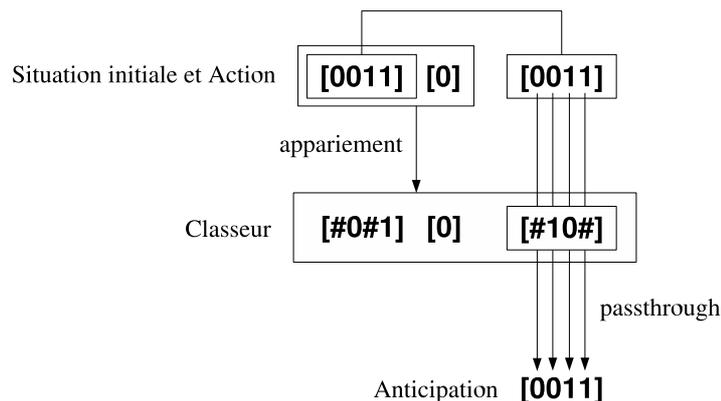


FIGURE 3.4 – Mécanisme d'anticipation d'un classeur dans ACS [Gerard, 2002]

3.2.3.3 Anticipation et comportement d'agents

Quelques travaux explorent l'utilisation de l'anticipation dans la conception d'agents faisant preuve de comportements adaptatifs complexes. Dans [Castelfranchi *et al.*, 2006], les auteurs présentent une architecture permettant de modéliser la prudence des agents. Le comportement résultant peut s'apparenter à une forme d'anticipation. L'approche proposée par les auteurs consiste à changer de comportements en fonction des événements se produisant. La fonction de transition entre les trois comportements : prudence, excitation, curiosité et ennui repose sur une classification des événements en : événements positifs et événements négatifs.

Dans [Laird, 2001], l'auteur détaille la conception d'un agent évoluant dans un environnement fortement dynamique et faisant preuve d'anticipation. Sa proposition est appliquée à la mise au point

d'un bot¹⁴ ayant des capacités d'anticipation sur les stratégies des joueurs et exhibant ainsi une intelligence supérieure aux bots classiques programmés par le biais de scripts.

La méthode décrite se rapproche d'un système expert doté de capacité de prédiction. Le bot "se construit une représentation interne permettant de simuler ce qu'il pense de l'état interne de l'ennemi à partir de ses propres observations de celui-ci". Le bot prédit le comportement de l'ennemi en appliquant au contexte observé (base de faits) ses propres connaissances en terme de stratégies et de tactiques. Les actions externes de l'environnement sont simulées d'un point de vue interne par l'utilisation d'un ensemble de règles (de production) lui permettant de se projeter dans le futur jusqu'à obtention d'une prédiction intéressante (principe de chaînage avant dans les systèmes experts). A partir de cette prédiction, le bot adapte son comportement en optant pour une de ses stratégies disponibles.

Laird analyse et critique les capacités d'anticipation de ses bots en terme de généralisation à d'autres types de jeux, voire d'autres applications. De nombreuses parties de sa formalisation sont en effet très dépendantes de son application. En particulier, la représentation interne de l'état des ennemis dépend fortement de la structure des données utilisée dans le jeu vidéo. Ce manque de généricité nous empêche de considérer cette approche dans le cadre de notre problématique.

3.2.3.4 Anticipation dans les systèmes multi-agents

Dans le cadre des systèmes multi-agents, la notion d'anticipation est parfois évoquée sans toutefois être au cœur de la problématique. Ainsi, dans le cadre particulier des SMA adaptatifs, la prise en compte et la minimisation du préjudice susceptible d'être infligé à d'autres agents s'apparentent à une forme d'anticipation permettant d'éliminer les situations non coopératives [Georgé, 2004].

Parmi les travaux significatifs relatifs à l'intégration de l'anticipation dans la conception de systèmes multi-agents, nous pouvons citer ceux de Davidsson. Dans [Davidsson, 2003], l'auteur reprend le concept d'anticipation préventive introduit dans [Rosen, 1974] et qui consiste à anticiper une situation non désirée et à adapter son comportement de manière à éviter qu'elle ne se produise.

Rosen donne un cadre formel à ce type d'anticipation en considérant un système S et son modèle M . S étant dynamique, son état varie dans le temps : S_n désigne l'état de S à l'instant t_n . De la même manière, l'état du modèle correspondant à S_n est noté M_n . M et S ne sont pas sur la même échelle de temps et évoluent en décalage de sorte qu'il existe un Δt entre M_n et S_n . L'observation de M_n permet donc de prédire S_n (figure 3.5). Rosen introduit, en complément de ce cadre, la notion d'effecteur permettant à M d'agir sur S .

Pour répondre à la problématique consistant à déterminer comment les prédictions issues de M peuvent influencer et modifier le comportement de S à l'instant présent, Rosen propose une partition de l'espace des états de S . Cette partition consiste à diviser S en deux régions correspondant aux états *désirés* et *non désirés*. *Tant que la prise de décision dans M permet de rester dans un état désiré, aucune action n'est prise par M au travers des effecteurs. Dès que la prise de décision conduit M dans un état non désiré [...] les effecteurs sont activés de manière à changer la dynamique de S de telle manière que le système S ne se retrouve pas dans un état non désiré* [Rosen, 1974].

Pour concevoir une architecture agent utilisant l'anticipation préventive, Davidsson pose comme condition préalable d'inclure une représentation de l'environnement dans le modèle M . Il définit ensuite les quatre entités composant son architecture (figure 3.6) [Davidsson, 2003] :

14. adversaire virtuel d'un jeu vidéo doté d'une certaine forme d'intelligence et d'autonomie

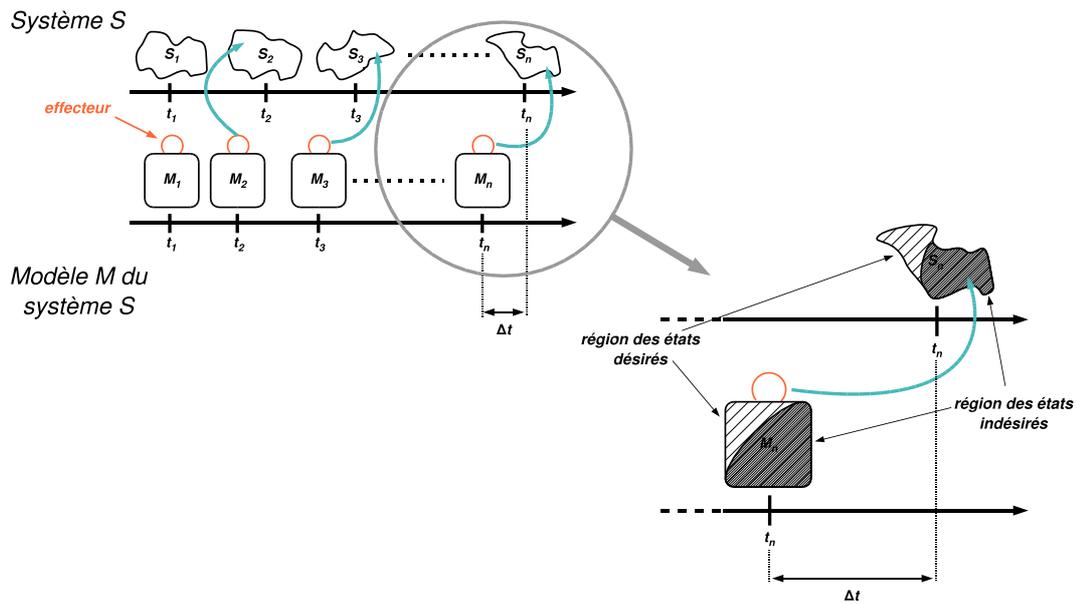


FIGURE 3.5 – L'anticipation selon Rosen

- un module de décision basé sur une approche réactive,
- une couche anticipative composée d'une représentation de l'environnement et d'un méta-composant appelé "anticipateur" capable de simuler en accéléré la représentation de l'environnement et ainsi fournir des prédictions,
- un module appelé "senseur" regroupant les fonctions de perception,
- un module appelé "effecteur" et chargé d'appliquer les décisions prises par l'agent.

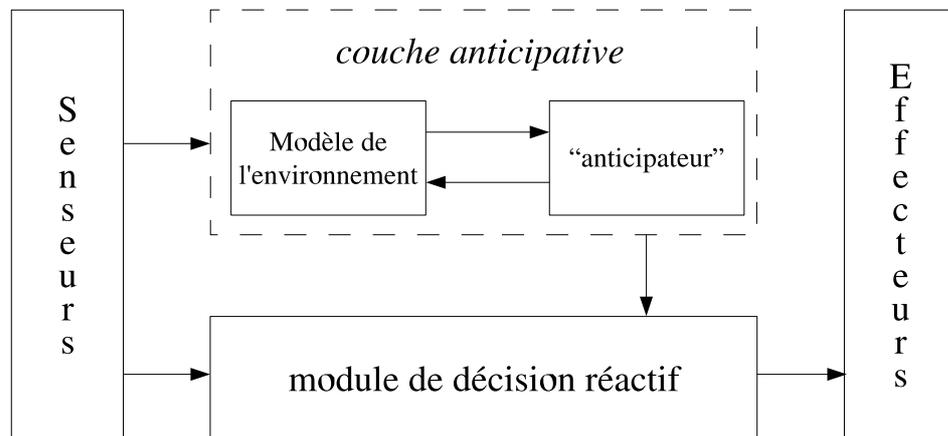


FIGURE 3.6 – Architecture agent pour l'anticipation préventive [Davidsson, 2003]

Comme le suggère la figure 3.6, les couches anticipatives et réactives sont exécutées en parallèle sous la forme de deux processus concurrents. Les données recueillies par les senseurs servent à mettre à jour le modèle de l'environnement et fournissent des stimuli au module de décision réactive. Le composant "anticipateur" utilise le modèle de l'environnement pour générer des prédictions et décider en fonction de celles-ci de modifier les propriétés du module de prise de décision.

3.2.4 Conclusion

Dans cette section, nous avons présenté le concept d'anticipation. Nous nous sommes en particulier attachés à mettre en évidence les différentes implications de l'anticipation sur le comportement (animal ou humain) et les tentatives de formalisation issues de la biologie.

Différentes applications de l'anticipation en intelligence artificielle ont ensuite été exposées à travers différents travaux. Dans le cadre des systèmes multi-agents, très peu de travaux ont investigué le concept d'anticipation. Seule l'approche de Davidsson réutilisant l'anticipation préventive proposée par Rosen semble convenir à notre objectif d'éviter les interblocages en simulation de trafic.

L'architecture proposée par Davidsson reste néanmoins peu précise sur différents points. En particulier, les mécanismes d'inférence et de mise en évidence des états non désirés ne sont pas clairement explicités. La nature même des états non désirés reste indéfinie. Dans ce contexte, nous proposons donc une formalisation du concept d'anticipation préventive s'inspirant du formalisme des réseaux de contraintes.

3.3 Un modèle d'anticipation fondé sur les réseaux de contraintes

Dans cette partie, nous développons une formalisation de l'anticipation préventive précédemment introduite. Le modèle proposé repose sur l'utilisation des réseaux de contraintes également appelés dans la littérature "problèmes de satisfaction de contraintes". Nous commençons donc cette section en introduisant les différents concepts et principes utilisés dans notre modélisation.

3.3.1 Réseaux de contraintes

Les premières études relatives aux problèmes de satisfaction de contraintes (CSP¹⁵) apparaissent dans le courant des années 70, avec comme champ d'application la vision et le traitement d'images [Waltz, 1975]. Quelques années plus tard, les CSP sont utilisés dans de nombreux domaines : raisonnement temporel [Allen, 1984], conception d'interfaces [Sanella *et al.*, 1992], planification [Fox, 1987], diagnostic [Piechowiak, 2001], etc.

3.3.1.1 Définitions

Les CSP fournissent un modèle simple et efficace pour formaliser un grand nombre de problèmes en informatique.

Définition 3.2 *Un CSP est un triplet (X, D, C) où :*

- X est un ensemble fini de n variables : $\{x_1, x_2, \dots, x_n\}$
- chaque variable x_i ($1 \leq i \leq n$) prend ses valeurs sur un domaine fini : $\text{dom}(x_i)$ et $D = \{\text{dom}(x_1), \text{dom}(x_2), \dots, \text{dom}(x_n)\}$
- C est un ensemble fini de p contraintes : $\{c_1, c_2, \dots, c_p\}$, chaque contrainte c_i s'appliquant à q variables ($1 \leq q \leq n$)

15. acronyme anglais pour Constraint Satisfaction Problem

Lorsque $q \leq 2$ pour l'ensemble des contraintes C , le CSP est dit *binnaire* et peut être assimilé à un graphe orienté. Dans la littérature, le terme *réseau de contraintes* est généralement employé pour désigner les CSP binnaires. Les variables sont alors représentées par les sommets du graphe et les contraintes par les arêtes. La figure 3.7 illustre ceci. Le CSP considéré possède trois variables : a , b et c prenant chacune leurs valeurs dans un ensemble $\{1, 2, 3\}$. L'ensemble C comporte trois contraintes exprimant que chaque variable de X est différente l'une de l'autre. Le réseau de contraintes résultant est formé de trois sommets : a , b et c reliés par trois arcs, chacun porteur d'une contrainte d'inégalité.

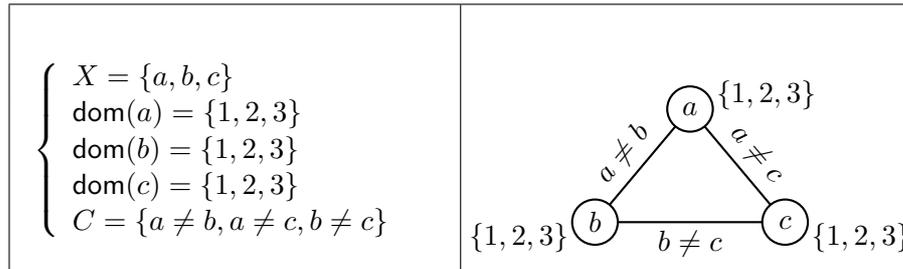


FIGURE 3.7 – Exemple de réseau de contraintes

Définition 3.3 Soit $X = \{x_1, x_2, \dots, x_n\}$, $X_P = \{v_1, v_2, \dots, v_k\}$ ($1 \leq k < |X|$) ($|X|$ désignant la cardinalité de l'ensemble X) est une *instanciation partielle sur X* si et seulement si $\forall i \in [1, k] v_i \in \text{dom}(x_i)$. Lorsque $k = |X|$, X_P est appelé *instanciation totale*.

Définition 3.4 Soit un CSP (X, D, C) et X_P une *instanciation partielle de X* . X_P est *localement consistante pour (X, D, C)* si et seulement si elle satisfait l'ensemble des contraintes C .

3.3.1.2 Raisonnements applicables à un réseau de contraintes

Suivant la nature du problème à traiter, différents raisonnements peuvent être appliqués à un réseau de contraintes. Il peut s'agir par exemple de résoudre le réseau, c'est-à-dire trouver une instanciation totale sur X localement consistante.

La solution d'un CSP n'existe pas nécessairement. Un CSP qui n'admet pas de solution est dit *sur-contraint* ou encore *inconsistant*¹⁶. Pour un tel CSP, il devient alors intéressant de rechercher les causes de cette inconsistance, voire même de rechercher quelles contraintes permettraient, par leur relaxation, d'obtenir une solution.

3.3.1.3 Arc-consistance et algorithmes de filtrage

Deux types d'algorithmes sont applicables sur les réseaux de contraintes : les algorithmes de recherche systématique dédiés à l'instanciation des variables et les algorithmes de filtrage permettant de réduire les domaines des variables et donc accélérer la recherche de solutions¹⁷. Nous ne

16. Par extension, une inconsistance désigne l'impossibilité d'affecter toutes les variables du réseau

17. des algorithmes "hybrides" combinent recherche systématique et filtrage comme par exemple le *forward-checking* [Haralick et Elliot, 1980]

présentons ici que les algorithmes de filtrage (également appelés algorithmes de propagation de contraintes) qui seront utilisés dans notre modèle d'anticipation. Pour une présentation plus complète de l'ensemble des techniques et des algorithmes utilisables sur les réseaux de contraintes, le lecteur peut se référer à [Kumar, 1992], [Tsang, 1993].

Les algorithmes de filtrage reposent sur la notion de k -consistance ($1 \leq k \leq |X|$) :

Définition 3.5 *Un réseau de contraintes est k -consistant si et seulement si pour toute instanciation partielle de $k - 1$ variables localement consistantes, pour toute $k^{\text{ème}}$ variable, il existe une valeur v dans son domaine telle que l'instanciation partielle obtenue en affectant la $k^{\text{ème}}$ variable de la valeur v soit localement consistante.*

Définition 3.6 *Un réseau est fortement k -consistant si et seulement si il est j -consistant pour tout j tel que $1 \leq j \leq k$.*

Dans la pratique, la k -consistance (pour $k > 2$) n'est pas utilisée puisque les algorithmes résultant ont une complexité équivalente à celle des algorithmes de recherche. Seule l'arc-consistance ($k = 2$) est utilisable et permet de simplifier le réseau pour un coût calculatoire raisonnable.

Définition 3.7 *Un arc (X, Y) est dit arc-consistant si pour tout $a \in \text{dom}(X)$ il existe $b \in \text{dom}(Y)$ de telle manière que les instanciations $X \leftarrow a$ et $Y \leftarrow b$ soient permises par la contrainte entre X et Y .*

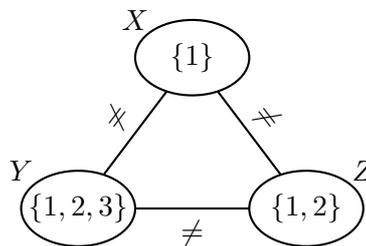


FIGURE 3.8 – Arc-consistance

La propriété d'arc-consistance n'est pas bidirectionnelle : si un arc (X, Y) est arc-consistant cela ne veut pas nécessairement dire que l'arc (Y, X) l'est aussi. Dans l'exemple de la figure 3.8, l'arc (X, Z) est clairement consistant puisque si X prend la valeur 1, il est possible d'affecter 2 à Z et la contrainte $X \neq Z$ est bien satisfaite. En revanche, l'arc (Z, X) n'est pas consistant : lorsque Z prend la valeur 1, aucune valeur dans $\text{dom}(X)$ ne permet de satisfaire la contrainte $Z \neq X$. Pour rendre l'arc (Z, X) consistant, il suffit de supprimer la valeur 1 de $\text{dom}(Z)$. De la même manière, l'arc (Y, X) n'est pas consistant à moins de supprimer la valeur 1 de $\text{dom}(Y)$.

Le premier algorithme de propagation de contraintes (AC-3) a été proposé en 1977 par Mackworth [Mackworth, 1977a]. De nombreuses améliorations ont par la suite été proposées : AC-4 [Mohr et Henderson, 1986], AC-5 [Deville et van Hentenryck, 1991], AC-6 [Bessière, 1994], AC-7 [Bessière *et al.*, 1995]. Tous ces algorithmes reposent sur une fonction permettant de traiter un arc du réseau et de réduire les deux domaines associés, les différentes variantes entre les algorithmes concernent la manière de parcourir le réseau : ajout de marquage, maintien de listes d'arcs déjà parcourues, etc.

3.3.1.4 Réseaux de contraintes dynamiques

Une extension des réseaux de contraintes a été introduite dans la littérature de manière à prendre en compte le caractère dynamique requis pour la modélisation de certains problèmes.

Définition 3.8 ([Bessière, 1992]) *Un réseau de contraintes dynamiques est une suite de réseaux statiques P_0, P_1, \dots, P_n tel que chaque P_i diffère de P_{i+1} par l'ajout (restriction) ou la suppression (relaxation) d'une contrainte.*

La majorité des travaux sur les réseaux de contraintes dynamiques ont porté sur la proposition d'algorithmes de filtrage. Considérons maintenant $\phi(P_i)$ le réseau obtenu après application du filtrage ϕ sur P_i . Une des propriétés du filtrage étant de préserver les espaces de solutions et les relations entre ces espaces, on a : $\phi(P_{i+1}) \subseteq \phi(P_i)$. En ajoutant une contrainte, il n'est pas nécessaire de remettre en cause le travail de filtrage déjà effectué sur un réseau.

La suppression d'une contrainte c dans C_{i+1} est un peu plus problématique. En effet, il faut être capable de replacer dans les différents domaines les valeurs qui avaient été supprimées lors du calcul d'arc-consistance sur la contrainte c . Pour ne pas perdre le bénéfice des calculs déjà effectués, les algorithmes de filtrage des réseaux de contraintes statiques doivent être modifiés de manière à mémoriser certaines informations nécessaires à la réintroduction des valeurs préalablement supprimées : dnAC-4 [Bessière, 1992], AC/DC [Neveu et Berlandier, 1994], dnAC-6 [Debruyne, 1995].

3.3.2 Formalisation de l'anticipation

Nous proposons ici une formalisation de l'anticipation basée sur l'utilisation des principes et techniques relatifs aux réseaux de contraintes. Nous posons quelques hypothèses quant aux caractéristiques de l'environnement requises pour l'application de notre modèle et introduisons les différentes notations nécessaires à sa définition.

3.3.2.1 Propriétés de l'environnement et sa représentation

Pour formaliser l'anticipation, nous nous plaçons dans les conditions suivantes :

Définition 3.9 *Soit un système multi-agent composé d'un ensemble d'agents $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ plongé dans un environnement \mathcal{E} . $I_{\mathcal{E}}$ désigne l'ensemble des informations perçues de l'environnement par un agent de \mathcal{A} . $I_{\mathcal{E}}$ comprend :*

- un sous-ensemble de \mathcal{A} noté $I_{\mathcal{A}}$ représentant les agents perçus
- un ensemble $\mathcal{P} = \{p_{a_1}, p_{a_2}, \dots, p_{a_i}, \dots\}$ de propriétés décrivant chaque $a_i \in I_{\mathcal{A}}$

Dans cette définition, le terme de "propriétés" regroupe toutes les informations relatives à un agent et pouvant être perçues par d'autres. Ces informations peuvent être de natures diverses : position dans l'espace dans le cas d'agents situés, relations topologiques par rapport à d'autres agents ou objets de l'environnement, description contextuelle, caractéristiques internes de l'agent ayant une visibilité extérieure, etc.

Les caractéristiques relatives à l'environnement \mathcal{E} et à sa perception $I_{\mathcal{E}}$ par les agents conditionnent l'anticipation dont ces derniers peuvent faire preuve. Nous posons tout d'abord l'hypothèse que

l'anticipation dans un système multi-agent requiert que \mathcal{E} soit déterministe [Russell et Norvig, 1995]. Cette caractéristique de l'environnement assure que les agents seront capables de générer de bonnes prédictions en calculant les différents états futurs de l'environnement (cf section 1.2.2.2).

Notre seconde condition relative à \mathcal{E} porte sur son accessibilité. Cette propriété de l'environnement qui caractérise la qualité de perception des agents influe directement sur la qualité des prédictions calculées par les agents. Plus les agents percevront l'environnement de manière précise et complète, plus les prédictions émises seront justes. Nous posons donc comme condition que les agents disposent d'informations "parfaites" sur leur environnement local. La connaissance parfaite de l'ensemble de l'environnement n'est pas requise puisque par définition l'anticipation préventive que nous proposons de formaliser est ego-centrée.

Un modèle partiel de son environnement proche oblige l'agent à considérer différentes possibilités. La structure des prédictions successives prend une forme arborescente : pour un instant donné il peut exister plusieurs états futurs possibles. De par les différentes hypothèses que nous avons posées auparavant, nous nous plaçons dans le cadre d'une anticipation linéaire : chaque prédiction pour un instant donné est unique et dépend de la prédiction précédente ou de l'instant présent (figure 3.9).

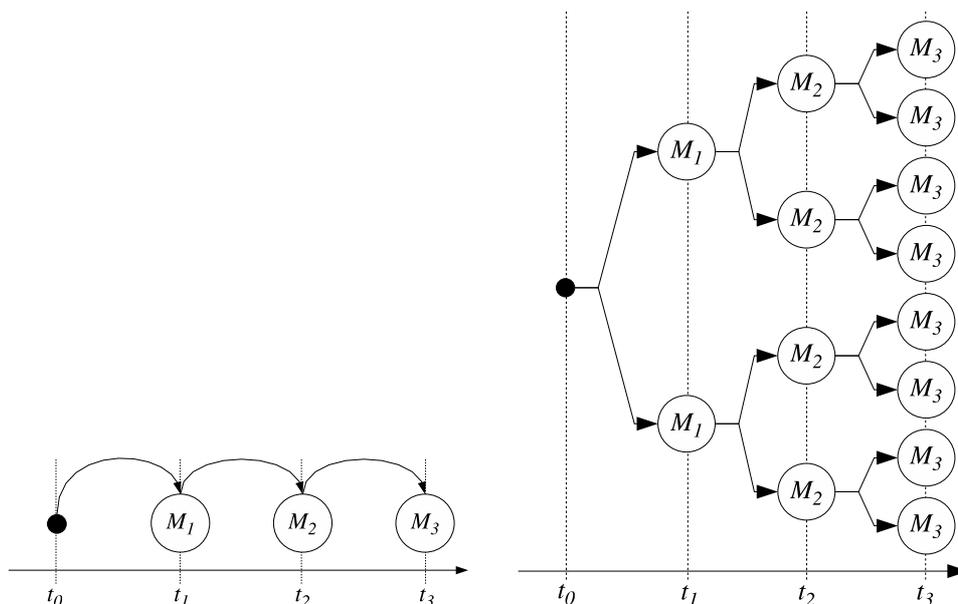


FIGURE 3.9 – Anticipation linéaire et anticipation arborescente (inspiré de [Davidsson, 2003])

3.3.2.2 Définition des effets d'une action

Prédire l'état futur de l'environnement consiste à déterminer la manière dont les actions des différents agents vont le modifier. Autrement dit, la prédiction dépend de l'estimation des conséquences d'une action. Pour prédire, nous considérons que chaque action exécutée induit un ensemble d'effets sur l'environnement. Si l'on se place du point de vue d'un agent, ces effets peuvent être directs ou indirects [Doniec *et al.*, 2005a].

Définition 3.10 Soit $\mathcal{R} = \{r_{ij} | (a_i, a_j) \in \mathcal{A}^2\}$ un ensemble de relations binaires existant entre les différents agents de \mathcal{A} . L'ensemble des connaissances de l'agent a_m relatives à \mathcal{R} peut être défini comme $\mathcal{C}_{a_m}^+ \cup \mathcal{C}_{a_m}^*$ où $\mathcal{C}_{a_m}^+ = \{r_{xy} | (x = m) \vee (y = m)\}$ et $\mathcal{C}_{a_m}^* = \{r_{xy}^* | x \in \mathcal{A} - \{a_m\} \wedge y \in \mathcal{A} - \{a_m\}\}$ avec r_{xy}^* désignant les relations qui sont estimables (au sens calculable) de manière certaine par a_m .

Autrement dit, parmi les différentes relations qui existent entre les agents d'un SMA, un agent a_m ne connaît que celles dans lesquelles il est impliqué (ensemble $\mathcal{C}_{a_m}^+$) plus celles pour lesquelles il dispose d'informations et de capacités suffisantes pour les déduire (ensemble $\mathcal{C}_{a_m}^*$). Cette définition de l'ensemble des connaissances relatives aux relations et interactions existant au sein du système nous permet de définir les effets directs et indirects d'une action s exécutée par a_m .

Définition 3.11 Les effets directs sont les conséquences induites sur \mathcal{R} par s pour les agents de l'ensemble $\{a_n \in \mathcal{A} | \exists r_{nm} \in \mathcal{C}_{a_m}^+\}$. Les effets indirects sont les conséquences induites par l'action s sur \mathcal{R} pour les agents de l'ensemble $\{a_n \in \mathcal{A} | \exists r_{nm} \in \mathcal{C}_{a_m}^*\}$.

3.3.2.3 Principe de notre approche

Le principe de notre approche consiste à construire une représentation mentale de la situation perçue. Cette représentation est une abstraction de l'ensemble des informations qu'un agent perçoit de son environnement. N'apparaît dans cette représentation qu'un sous-ensemble de \mathcal{A} correspondant aux agents perçus et impliqués dans la situation et un sous-ensemble de \mathcal{R} représentant les différentes relations perçues entre les agents.

L'agent devant anticiper doit apparaître dans la représentation qu'il se construit. Ceci est justifié par la définition même de l'anticipation donnée par Rosen : l'agent doit être capable de raisonner sur les conséquences de ses propres actions. D'autres auteurs comme Zeigler [Zeigler, 1990] viennent confirmer cette nécessité pour un agent de s'inclure dans sa représentation de l'environnement.

Tout le raisonnement de l'agent s'effectue à partir de cette représentation mentale : l'anticipation de son état futur va consister à évaluer dans cette représentation les effets directs et indirects des différentes actions pouvant être exécutées à l'instant présent. Comme nous le verrons dans la suite, ces actions peuvent être les siennes (raisonnement ego-centré) mais aussi celles d'autres agents. Cette capacité à évaluer les conséquences des actions d'autrui nécessite cependant une connaissance plus étendue de l'environnement.

La prise en compte de ces effets directs et indirects par l'agent lui permet d'inférer un certain nombre d'états. L'agent recherche ensuite parmi ces états ceux qui sont non-désirés et supprime les actions qui en sont à l'origine.

3.3.2.4 Formalisation de la représentation mentale d'un agent

La représentation mentale d'un agent s'appuie sur une structure inspirée des réseaux de contraintes [Doniec et al., 2006c].

Définition 3.12 Une représentation mentale M d'un agent est définie par la donnée :

- d'un ensemble d'agents $A = \{a_1, a_2, \dots, a_n\} \subset \mathcal{A}$,

- d'un ensemble $R = \{r_1, r_2, \dots, r_i, \dots, r_k\} \subset \mathcal{R}$ où chaque r_i établit une relation binaire¹⁸ entre deux agents de A ,
- d'un ensemble D de domaines $\{\text{dom}(a_1), \text{dom}(a_2), \dots, \text{dom}(a_n)\}$ associés à chaque agent exprimant un ensemble d'alternatives futures.

La nature du domaine associé à chaque agent peut varier suivant l'application. Le domaine peut être par exemple de nature temporelle et exprimé par exemple des fenêtres de temps relatives à une interaction particulière. Le domaine peut aussi exprimer différentes alternatives spatiales pour un agent situé. Dans l'exemple de la figure 3.10, trois agents évoluent dans un monde dont l'espace est discrétisé sous la forme d'une grille.

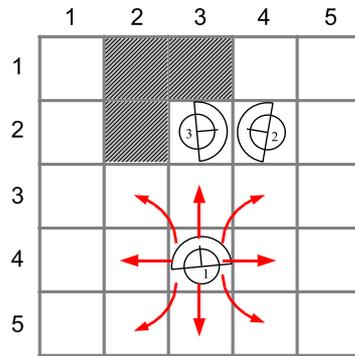


FIGURE 3.10 – Trois agents évoluant sur une grille

Chaque case de la grille peut être repérée par le biais de coordonnées : l'agent 1 par exemple se trouve sur la case (3, 4) et peut potentiellement se déplacer sur l'ensemble de la grille. Ceci est exprimé par un domaine égal à l'ensemble $(3, 4) \times [1..5]^2$. La position des agents et des obstacles¹⁹ fait que certains mouvements seront impossibles. Cela peut se traduire en utilisant des relations topologiques entre les agents et/ou les obstacles. Par exemple, du point de vue de l'agent 1, l'agent 2 est à la droite de l'agent 3. Une représentation mentale possible pour l'agent 1 pourrait donc être :

$$\begin{aligned}
 A &= \{agent_1, agent_2, agent_3, obstacle_1, obstacle_2, obstacle_3\} \\
 D &= \{\text{dom}(agent_1) = \{(3, 4) \times [1..5]^2\}, \text{dom}(agent_2) = \{(4, 2) \times [1..5]^2\}, \\
 &\quad \text{dom}(agent_3) = \{(3, 2) \times [1..5]^2\}, \text{dom}(obstacle_1) = \{(2, 1) \times (2, 1)\}, \\
 &\quad \text{dom}(obstacle_2) = \{(3, 1) \times (3, 1)\}, \text{dom}(obstacle_3) = \{(2, 2) \times (2, 2)\}\} \\
 R &= \{\text{droite}(agent_2, agent_3), \text{gauche}(obstacle_3, agent_3), \text{devant}(obstacle_2, agent_3)\}
 \end{aligned}$$

3.3.2.5 Inférence des états non désirés

Dans notre modélisation, nous considérons les états non désirés soit comme des domaines particuliers, soit comme des inconsistances au sens des réseaux de contraintes (comme par exemple la

18. lorsqu'il s'agit de représenter une relation faisant intervenir plus de deux agents, il est possible de se ramener à une relation binaire par le biais de techniques de transformation [Tsang, 1993]

19. Dans cet exemple, nous considérons que les obstacles sont des agents immobiles, d'où leur inclusion dans l'ensemble A .

présence d'un ou plusieurs domaines vides). Les effets d'une action sont considérés comme des modifications de l'ensemble R dans la représentation mentale des agents. Plus précisément, les effets directs d'une action se traduiront par l'ajout ou la suppression d'une ou plusieurs relations binaires dans R . Les effets indirects sont quant à eux calculés par propagation de ces relations binaires.

Les techniques de propagation de contraintes présentées précédemment peuvent être réutilisées. Pour illustrer ceci, reprenons l'exemple de la figure 3.10 et sa modélisation associée. La position des agents et des obstacles fait que certains mouvements seront impossibles. Ceux-ci peuvent être déterminés, en associant aux relations *droite*, *gauche* et *devant*, la sémantique suivante :

$$\left. \begin{array}{l} droite(((x_1, y_1), (x_2, y_2)), ((x_3, y_3), (x_4, y_4))) \\ gauche(((x_1, y_1), (x_2, y_2)), ((x_3, y_3), (x_4, y_4))) \\ devant(((x_1, y_1), (x_2, y_2)), ((x_3, y_3), (x_4, y_4))) \end{array} \right\} \equiv (x_1, y_1) \neq (x_4, y_4)$$

L'utilisation d'un algorithme de propagation de contraintes permet de supprimer trois des déplacements possibles de l'agent 3 : $((3, 2), (3, 1))$, $((3, 2), (4, 2))$ et $((3, 2), (2, 2))$ et ainsi simplifier le domaine correspondant : $\text{dom}(agent3) = \{((3, 2), (3, 3))\}$.

A partir de cette représentation mentale, l'agent 1 peut inférer les états non désirés relatifs à ces différentes actions possibles. Considérons par exemple l'action consistant à avancer d'une case devant cet agent. Les effets directs de cette action se traduisent par l'ajout d'une nouvelle relation binaire entre l'agent 1 et l'agent 3 : $devant(agent3, agent1)$. Les effets indirects quant à eux peuvent être obtenus en effectuant une nouvelle propagation. Compte tenu de la nouvelle relation ajoutée, le domaine de l'agent 3 se réduit encore puisque le déplacement $((3, 2), (3, 3))$ devient impossible, d'où $\text{dom}(agent3) = \emptyset$.

En supposant que l'agent 1 ait pour principal objectif de se déplacer sans gêner les autres agents, la prédiction $\text{dom}(agent3) = \emptyset$ peut constituer un état non désiré.

3.3.3 Algorithmes d'anticipation

Nous synthétisons les différents points précédemment traités sous la forme d'algorithmes généraux détaillant la construction de la représentation mentale d'un agent et son utilisation dans le cadre de l'anticipation préventive.

Dans les paragraphes qui suivent, nous supposons l'existence de fonctions et de procédures de manipulation de réseaux de contraintes : ajout, suppression de contraintes, propagation, recherche de domaines, etc.

3.3.3.1 Construction de la représentation mentale

La construction par un agent d'une représentation mentale s'appuie sur une abstraction des informations $I_{\mathcal{E}}$ perçues de l'environnement permettant d'établir les différentes relations existantes entre les agents de l'ensemble $I_{\mathcal{A}}$. Nous supposons donc l'existence d'une fonction $abstractEnv : I_{\mathcal{E}} \times I_{\mathcal{A}} \rightarrow R$.

La procédure $construitRepMentale$ prend donc en entrée l'ensemble $I_{\mathcal{E}}$ ainsi qu'une valeur d'initialisation des domaines. Le type de cette valeur d'initialisation dépend bien entendu du contexte applicatif. La procédure retourne en sortie une représentation mentale M .

```

procedure construitRepMentale(in :  $I_{\mathcal{E}}(I_{\mathcal{A}}, \mathcal{P})$ , valeurInitDomaine
                                out :  $M(A, R, D)$ )
begin
  for each  $a_i \in I_{\mathcal{A}}$  do
    miseAJourAgent( $A, a_i$ )
     $\text{dom}(a_i) \leftarrow \text{valeurInitDomaine}$ 
    miseAJourDomaine( $D, \text{dom}(a_i)$ )
  end
  for each  $p_i \in \mathcal{P}$  do
     $r \leftarrow \text{abstractEnv}(I_{\mathcal{E}}, p_i)$ 
    miseAJourRelation( $R, r$ )
  end
end

```

3.3.3.2 Procédure d'anticipation

La procédure *anticipe* prend en entrée une liste d'actions LA , la représentation mentale M de la situation, l'ensemble des informations relatives à l'environnement et une liste d'états non désirés LE . La liste d'actions LA utilisée en entrée de l'algorithme peut contenir les actions propres de l'agent et éventuellement celles d'autres agents.

Le rôle de cette procédure est de supprimer de la liste LA les actions qui conduiront dans le futur à un des états non désirés de la liste LE . La première opération de la procédure consiste à filtrer les différents domaines de M par appel à une fonction extérieure de propagation. La liste LA est ensuite parcourue et pour chaque action on détermine ses effets probables.

La fonction *calculeEffets* regroupe différents calculs de projection permettant de déterminer les effets directs de l'action. Cette fonction n'est pas générique et dépend également du contexte applicatif. Les effets indirects sont ensuite calculés grâce à une mise à jour de M et une propagation des effets directs précédemment calculés. Remarquons que lorsque la mise à jour de M consiste en une ou plusieurs suppression de relations, les domaines doivent être réinitialisés de manière à prendre en compte les valeurs qui avaient pu être retirées lors de la première propagation. Ceci ne vaut que pour l'utilisation d'algorithmes de propagation classique, nous verrons au chapitre 4 que l'utilisation d'algorithme incrémentaux permet de s'affranchir de cette opération.

La dernière partie de la procédure concerne l'analyse des prédictions obtenues pour l'action considérée. Cette analyse consiste en une comparaison des différents domaines de D avec ceux présents dans la liste des états non désirés LE . Lorsqu'un état non désiré est détecté, l'action considérée est éliminée de la liste LA .

```

procedure anticipe(in : ListeActions  $LA$ , ListeEtatsIndesires  $LE$ ,
                     $M(A, R, D)$ ,  $I_{\mathcal{E}}$ 
                    sortie : ListeActions  $LA$ )
begin
  propagation( $M$ )
   $M' \leftarrow M$ 
  for each  $action \in LA$  do
     $r \leftarrow \text{calculeEffets}(a, I_{\mathcal{E}})$ 
    miseAJourRelation( $R, r$ )
    propagation( $M$ )
     $existe \leftarrow \text{rechercheEtatsIndesires}(M, LE)$ 
    if  $existe$  then
       $LA \leftarrow LA - \{action\}$ 
    end
     $M \leftarrow M'$ 
  end
end

```

```

function rechercheEtatsIndesires(in :  $M(A, R, D)$ , ListeEtatsIndesires  $LE$ )
begin
  for each  $\text{dom}(x) \in LE$  do
     $\text{dom}'(x) \leftarrow \text{rechercheDomaine}(x, M)$ 
    if  $\text{dom}'(x) = \text{dom}(x)$  then
      return(vrai)
    end
  end
  return(faux)
end

```

Les différentes manipulations (mise à jour des relations et réduction des domaines) effectuées sur M pour chaque action de la liste LA obligent à effectuer une copie M' de manière à ce que M reste invariant pour chaque itération sur la liste LA . Cet aspect fait partie des nombreuses optimisations qui peuvent être apportées à l'algorithme de base que nous venons de présenter.

En effet, pour une structure M relativement petite (domaines de faibles tailles, nombres d'agents et de relations peu élevés) la copie de M n'engendre pas de coûts calculatoires excessifs. Lorsque la taille de M augmente, il devient intéressant de considérer les travaux réalisés autour des réseaux de contraintes dynamiques (section 3.3.1.4) et d'utiliser par exemple les algorithmes incrémentaux de propagation de contraintes [Debruyne, 1995]. Cette partie sera discutée plus longuement au chapitre 4 dans le cadre de l'implémentation du modèle au sein de la simulation de trafic ARCHISIM.

3.3.3.3 Analyse de la complexité

La complexité de la procédure de construction de la représentation mentale M d'un agent dépend de la taille de l'ensemble $I_{\mathcal{E}}$. En considérant que les opérations de mise à jour sont négligeables en terme de temps de calcul, nous obtenons pour *construitRepMentale* la complexité temporelle suivante :

$$T_{cRM} = |I_{\mathcal{A}}| + |\mathcal{P}|$$

La complexité de la procédure d'anticipation est fonction de l'algorithme de propagation de contraintes utilisé. Dans la littérature, on montre que ces algorithmes ont une complexité comprise entre $O(ed^2)$ [Mohr et Henderson, 1986] et $O(ed^3)$ [Mackworth, 1977a], d étant la taille maximale des domaines et e le nombre total de contraintes dans le réseau. En supposant que les temps des différentes affectations et des opérations de manipulation de M soient négligeables, nous obtenons pour la procédure *anticipe* la complexité suivante :

$$T_{anticipe} = T_{propagation} + |LA|(T_{propagation} + |LE|)$$

Selon l'algorithme de filtrage utilisé, on obtient :

$$|R|d^2(|LA| + |LE| + 1) \leq T_{anticipe} \leq |R|d^3(|LA| + |LE| + 1)$$

La complexité de la procédure *anticipe* dépend donc :

- du nombre d'actions possibles de l'agent ($|LA|$)
- du nombre d'états non désirés à éviter ($|LE|$)
- du nombre d'alternatives exprimées dans les différents domaines ($d = \max_{a_i \in A}(\text{dom}(a_i))$)

La complexité totale relative à l'algorithme d'anticipation est donc dans le pire des cas en $O(|R|d^3(|LA| + |LE|) + |I_{\mathcal{A}}| + |\mathcal{P}|)$. Dans la pratique, cette complexité n'induit pas forcément des coûts calculatoires extrêmement élevés. En effet, pour bon nombre d'applications, il est possible de conserver une taille des domaines raisonnable et limitée à une petite dizaine de valeurs. De même, le nombre d'actions à anticiper reste en général relativement faible.

3.3.4 Instanciation du modèle d'anticipation dans le cadre de la simulation comportementale de trafic

Comme nous l'avons vu au chapitre 2, l'algorithme de coordination utilisé pour les situations de carrefours dans ARCHISIM ne garantit pas l'absence d'interblocage au cours de la simulation. L'introduction de comportements opportunistes contribue à l'apparition d'interblocage pour diverses situations de conduite : succession de tourne-à-gauche, stockage au centre des carrefours, présence de véhicules longs, etc.

3.3.4.1 Anticipation et coordination

D'une manière générale, nous avons mis en évidence un manque d'anticipation dans les prises de décision des conducteurs simulés (agents). Il semble en effet essentiel que des agents faisant preuve de comportements opportunistes (non-respect du Code de la Route en fonction du contexte) aient des capacités d'anticipation suffisantes pour éviter les accidents et les interblocages. Le modèle d'anticipation préventive que nous venons de présenter se prête parfaitement à ces objectifs. L'idée est

d'utiliser l'algorithme précédemment introduit pour éliminer, en amont de la coordination, les actions des conducteurs simulés qui pourraient conduire à un interblocage complet en situation de carrefour [Doniec *et al.*, 2005b]. Il s'agit donc d'un pré-traitement permettant d'améliorer la coordination.

3.3.4.2 Modélisation

L'instanciation du modèle pour l'anticipation des phénomènes de blocages en carrefour requiert de définir les trois composantes de la représentation mentale M d'un agent. L'ensemble A peut ainsi être constitué des agents présents dans le carrefour qu'ils soient en mouvement ou à l'arrêt (par exemple, les véhicules stockés au centre du carrefour attendant un créneau pour s'insérer ou traverser un flux). La définition de la zone de sélection de ces agents (zone d'anticipation) sera discutée au chapitre suivant.

Le domaine associé à chaque agent de l'ensemble A est de nature temporelle et représente les prochains pas de temps de la simulation exprimés sous la forme d'un ensemble d'entiers. Ainsi, à un instant t , si une valeur x n'est pas présente dans le domaine d'un véhicule simulé, cela signifie qu'il sera bloqué à $t + x$. Par exemple :

- $dom(a_x) = [1, +\infty[\equiv$ " *a_x peut potentiellement se déplacer et agir durant l'intervalle $t + 1$ à $+\infty$* "
- $dom(a_x) = [1, 4] \cup [8, 10] \equiv$ " *a_x est bloqué de $t + 5$ à $t + 7$* "

La taille du domaine sera également discutée au chapitre suivant.

Pour l'ensemble \mathcal{R} , nous reprenons les relations de priorités définies en section 3.1.3.2 auxquelles nous ajoutons trois relations spécifiques relatives aux blocages pouvant exister entre deux véhicules. Ces trois relations sont notées bph , $bpha$ et bpr et chacune d'entre elles est associée à une sémantique bien spécifique dans un contexte de situation de conduite (tableau 3.1).

Relation binaire	Sémantique associée	Signification concrète
$bph_{a_z}(a_x, a_y)$	Blocage physique effectif entre deux véhicules	"L'agent a_x est physiquement bloqué par l'agent a_y du point de vue de l'agent a_z ."
$bpha_{a_z}(a_x, a_y)$	Blocage physique prévu entre deux véhicules	"Compte tenu des informations dont il dispose, l'agent a_z peut déduire que a_x va être physiquement bloqué par a_y ."
$bpr_{a_z}(a_x, a_y)$	Blocage logique dû aux priorités du Code de la Route	"Du point de vue l'agent a_z , a_x est bloqué par a_y qui lui est prioritaire au sens du Code de la Route."

TABLE 3.1 – Sémantiques des relations de blocage considérées pour l'anticipation

L'introduction de trois niveaux de blocage est justifiée par différents points. D'une part, il nous est apparu judicieux de distinguer le blocage physique effectif du blocage physique prévisible. En effet, lors de la phase de propagation, le traitement de la relation est différent dans les deux cas. Dans le cas

d'un blocage physique effectif, le filtrage du domaine du véhicule bloqué est obtenu simplement en supprimant les quelques pas de temps correspondant au temps de redémarrage du véhicule bloquant. En revanche, lorsque le blocage physique est prévisible mais non effectif, il est nécessaire de calculer pour le véhicule concerné un temps d'arrivée sur le véhicule bloquant. Pour cela, il est nécessaire d'effectuer des calculs cinématiques à partir de la distance entre les deux véhicules, de l'accélération et de la vitesse courante du véhicule prochainement bloqué. Le domaine résultant peut être par exemple l'union de deux intervalles.

L'utilisation de la relation bpr_{a_z} est quant à elle justifiée par la nécessité de distinguer les relations de priorité qu'un agent a_z perçoit des relations de priorité dans lesquelles il est impliqué. Là encore, le traitement est différent. Les relations de priorité dans lesquelles a_z apparaît sont le résultat de l'agrégation présentée en section 3.1.3.2 et sont utilisées en entrée de l'algorithme de coordination. La relation bpr_{a_z} décrit une relation de priorité au sens du Code de la Route entre deux mobiles appartenant à l'ensemble $\mathcal{A} - \{a_z\}$. Du point de vue de a_z , deux manières d'évaluer cette relation sont possibles :

- a_z peut considérer que les deux véhicules impliqués respecteront le Code de la Route. Le domaine du véhicule bloqué sera alors l'union de deux intervalles résultant de la suppression des pas de temps nécessaires au véhicule bloquant pour dépasser le point de conflit.
- a_z peut transposer son propre comportement (seuil d'impatience, réactivité, niveau de respect du Code de la Route, etc) sur l'agent bloqué. Autrement dit, a_z peut se mettre dans les mêmes conditions (même position sur la chaussée, même vitesse, même accélération) que le véhicule bloqué et appliquer l'algorithme de coordination pour déterminer les actions respectives des deux véhicules concernés.

3.3.4.3 Exemple

Nous illustrons ici l'instanciation du modèle d'anticipation dans le cadre de la simulation de trafic. Pour cela, nous considérons cinq véhicules simulés évoluant dans un carrefour non signalisé soumis à la règle de la priorité à droite. Les mobiles a_t , a_s et a_z sont à l'arrêt (vitesse nulle) et bloqués par l'agent a_y . Les mouvements tournant des cinq véhicules sont les suivants :

- a_s vient du nord (N) et souhaite tourner à sa gauche (mouvement N \rightarrow E)
- a_t vient de l'est (E) et souhaite tourner à sa gauche (mouvement E \rightarrow S)
- a_z vient de l'est (E) et a tourné à sa gauche (mouvement E \rightarrow S)
- a_y vient de l'ouest (O) et souhaite tourner à gauche (mouvement O \rightarrow N)
- a_x vient du sud (S) et souhaite aller tout droit (mouvement S \rightarrow N)

Les distances inter-véhiculaires ne permettent plus aux agents a_t , a_s et a_z d'avancer, seuls les agents a_y et a_x conservent une marge de manœuvre (figure 3.11).

L'agent a_x peut donc se construire à l'instant t la représentation mentale suivante :

$$\begin{aligned}
 A &= \{a_x, a_t, a_s, a_z, a_y\} \\
 D &= \{\text{dom}(a_x) = [1, +\infty[, \text{dom}(a_t) = [1, +\infty[, \text{dom}(a_s) = [1, +\infty[, \\
 &\quad \text{dom}(a_z) = [1, +\infty[, \text{dom}(a_y) = [1, +\infty[\} \\
 R &= \{bpha(a_x, a_t), bph(a_t, a_z), bpr(a_t, a_s), bph(a_s, a_z), bph(a_z, a_y), bpha(a_y, a_t), \\
 &\quad prio(a_x, a_y) \wedge \neg prio(a_y, a_x)\}
 \end{aligned}$$

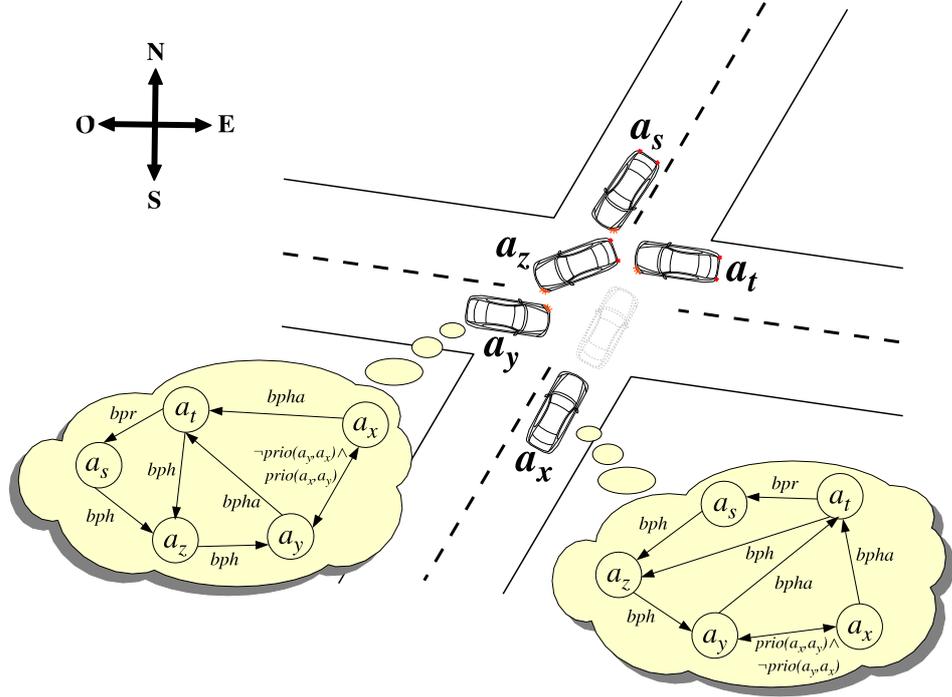


FIGURE 3.11 – Représentations mentales d'une situation de carrefour

De même pour l'agent a_y :

$$\begin{aligned}
 A &= \{a_y, a_x, a_t, a_s, a_z\} \\
 D &= \{\text{dom}(a_y) = [1, +\infty[, \text{dom}(a_x) = [1, +\infty[, \text{dom}(a_t) = [1, +\infty[, \\
 &\quad \text{dom}(a_s) = [1, +\infty[, \text{dom}(a_z) = [1, +\infty[\} \\
 R &= \{bpha(a_y, a_t), \neg prio(a_y, a_x) \wedge prio(a_x, a_y), bpha(a_x, a_t), bpr(a_t, a_s), bph(a_t, a_z), \\
 &\quad bph(a_s, a_z), bph(a_z, a_y)\}
 \end{aligned}$$

Si on applique l'algorithme donné en section 3.3.3.2 du point de vue de l'agent a_x , nous obtenons le raisonnement suivant :

1. propagation : Compte tenu des différentes relations de l'ensemble R , les domaines associés aux cinq agents de A peuvent être réduits. Par exemple, sachant que a_z est bloqué par a_y , il est possible de déduire que a_z restera immobilisé tant que a_y n'aura pas dépassé le point de conflit. En supposant que a_y accélère à l'instant t , un calcul cinématique à partir d'une accélération moyenne permet d'évaluer le temps durant lequel a_y restera sur la trajectoire de a_z . Supposons que ce temps soit égal à deux pas de temps²⁰, cela signifie que les valeurs 1 et 2 de l'intervalle $\text{dom}(a_z) = [1, +\infty[$ peuvent être supprimées. Le même raisonnement conduit à simplifier le domaine de a_s en $[5, +\infty[$ et de a_t en $[7, +\infty[$. La simplification de $\text{dom}(a_t)$ remet en cause le domaine de a_y du fait de l'existence de la relation $bpha(a_y, a_t)$. Ici encore il est nécessaire de procéder à un calcul cinématique destiné à évaluer le temps que a_y va mettre pour arriver au flanc du véhicule a_t . En supposant ce temps égal à quatre pas de simulation, le domaine de a_y

20. cette évaluation est en aucun cas réaliste, elle permet juste d'expliquer la manière de propager les relations de l'ensemble R et ainsi réduire les domaines de D

se réduit à $[1, 5] \cup [9, +\infty[$. De la même manière, le domaine de a_x se réduit à $[1, 5] \cup [9, +\infty[$ en supposant que quatre pas de temps lui sont également nécessaires pour atteindre le véhicule a_t . Au final, l'ensemble D a donc été modifié comme suit :

$$D = \{ \text{dom}(a_x) = [1, 5] \cup [9, +\infty[, \text{dom}(a_t) = [7, +\infty[, \text{dom}(a_s) = [5, +\infty[, \\ \text{dom}(a_z) = [3, +\infty[, \text{dom}(a_y) = [1, 5] \cup [9, +\infty[\}$$

2. calcul des effets des actions de a_x : L'algorithme de coordination ne gérant que l'accélération longitudinale, deux actions sont possibles pour a_x : *Go* et *Stop*. L'action *Go* conduit à un changement de position de a_x sur la chaussée (nouvelle position en pointillés sur la figure 3.11). La distance inter-véhiculaire entre a_x et a_t s'en trouve ainsi réduite et le blocage physique occasionné par a_t devient effectif. Cela se traduit par la modification de la relation $bpha(a_x, a_t)$ en $bph(a_x, a_t)$. De la même manière, la relation de priorité $prio(a_x, a_y) \wedge \neg prio(a_y, a_x)$ entre a_y et a_x évolue en une relation de blocage physique $bph(a_y, a_x)$ puisque a_x se trouve désormais sur la trajectoire de a_y . Le nouvel ensemble R résultant du choix de l'action *Go* est donc :

$$R = \{ bph(a_x, a_t), bph(a_t, a_z), bpr(a_t, a_s), bph(a_s, a_z), \\ bph(a_z, a_y), bpha(a_y, a_t), bph(a_y, a_x) \}$$

3. propagation : L'ajout de la relation $bph(a_y, a_x)$ remet en cause le domaine de a_y qui dans un premier parcours du réseau peut se réduire à $[13, +\infty[$. Or cela remet en cause tous les autres domaines dont celui de a_x . Au final, tous les domaines se réduisent à l'intervalle vide, ce qui du point de vue de notre modélisation est cohérent vu que tous les véhicules se retrouvent mutuellement bloqués :

$$D = \{ \text{dom}(a_x) = \emptyset, \text{dom}(a_t) = \emptyset, \text{dom}(a_s) = \emptyset, \text{dom}(a_z) = \emptyset, \text{dom}(a_y) = \emptyset \}$$

4. recherche des états non désirés : Du point de vue de a_x , avoir son propre domaine vide constitue un état non désiré. L'agent a_x doit donc éliminer l'action *Go* de sa liste d'actions possibles.

Dans cette situation, l'agent a_x n'a pas d'autre alternative et doit choisir de s'arrêter. Le mécanisme d'anticipation appliqué par l'agent a_y laisse inchangé sa liste d'actions possibles. Mais, de par le jeu de perception des priorités décrit en première partie de ce chapitre, l'agent a_y peut se considérer comme prioritaire au vu de la décélération pratiquée par a_x pour s'arrêter. Le mobile a_y peut donc s'engager et ainsi débloquent les trois autres mobiles a_z , a_s et a_t .

Conclusion

Dans ce chapitre, nous avons présenté notre contribution à la problématique de coordination de conducteurs simulés en carrefour. Celle-ci s'articule autour de deux apports.

Le premier concerne la reconnaissance de la situation en amont de la coordination. Nous nous sommes en particulier intéressés à la recherche des agents avec lesquels il est pertinent de se coordonner. Cette recherche s'effectue à partir des points de conflit situés sur la trajectoire de l'agent désirant se coordonner. Durant la recherche, celui-ci doit associer à chaque mobile perçu une relation de priorité. Cette relation est estimée en fonction des différents paramètres du mobile perçu (vitesse, accélération, direction, etc) et résulte d'une agrégation entre différents types de priorité (priorité par

rapport à l'impatience, à la vitesse). Une telle agrégation permet de surcharger la priorité relative au Code de la Route, offrant ainsi la possibilité aux agents de ne pas respecter strictement les règles du Code. Le modèle ainsi proposé permet aux agents de faire preuve d'un comportement opportuniste améliorant ainsi le réalisme des situations de conduite.

La deuxième partie de ce chapitre a été consacrée à la mise en place de comportements anticipatifs permettant d'éviter les interblocages au cours des simulations. L'approche retenue est celle de l'anticipation préventive proposée par Rosen. Cette classe d'anticipation a pour objectif de modifier le comportement courant d'un agent de manière à éviter certaines situations non désirées dans le futur. Dans ce cadre, nous avons proposé un modèle formel de l'anticipation préventive basé sur les principes et techniques de propagation locale dans les réseaux de contraintes.

La représentation mentale d'un agent est ainsi exprimée sous la forme d'un réseau de contraintes dans lequel un ensemble de relations binaires (contraintes) exprime les interactions perçues entre les autres agents. Le mécanisme d'anticipation proposé s'appuie sur une mise à jour de cette représentation permettant de prendre en compte les effets futurs d'une action. Les états non désirés sont associés à des inconsistances au sein du réseau.

La fin du chapitre propose une instanciation de ce modèle générique dans le cadre de la simulation de trafic. En complément des relations de priorité, nous introduisons trois relations permettant de décrire les interactions de blocage et de gênes mutuelles entre véhicules en carrefour.

Dans le chapitre suivant, nous détaillons cette instanciation en justifiant les différents choix effectués en terme d'implémentation. Nous présentons également le couplage mis en place entre l'algorithme d'anticipation et de coordination.

Chapitre 4

Application du modèle opportuniste / anticipatif à la simulation de trafic ARCHISIM

Sommaire

4.1	Choix d'implémentation	94
4.1.1	Définition des états non désirés	94
4.1.2	Distance d'anticipation	96
4.1.3	Choix de l'algorithme de propagation	98
4.1.4	Complexité en pratique	105
4.2	Intégration des comportements opportunistes et anticipatifs à ARCHISIM . .	106
4.2.1	Cycle de décision de l'agent	106
4.2.2	Couplage anticipation / coordination	108
4.3	Conclusion	109

Nous avons proposé au chapitre précédent un modèle de comportement multi-agent opportuniste et anticipatif. Une instanciation de ce modèle dans la cadre de la simulation de trafic routier a également été présentée.

Dans ce chapitre, nous discutons de l'intégration de notre proposition à la simulation de trafic ARCHISIM. Dans une première partie, nous détaillons les différents choix relatifs à cette instanciation. Nous nous intéressons en particulier à la définition des états non désirés pour un conducteur simulé. Les autres aspects abordés portent sur l'implémentation. Ils concernent notamment la distance d'anticipation, l'intervalle de temps sur lequel les actions sont anticipées et enfin le choix de l'algorithme de propagation utilisé.

La seconde partie de ce chapitre est consacrée à l'intégration de notre proposition dans le processus décisionnel d'un conducteur simulé. Nous développons en particulier le couplage entre le modèle d'anticipation proposé au chapitre 3 et l'algorithme de coordination présenté au chapitre 2.

4.1 Choix d'implémentation

Nous détaillons dans cette première section les choix effectués en terme d'implémentation et évaluons pour chacun d'entre eux l'incidence sur la complexité temporelle de l'algorithme d'anticipation.

4.1.1 Définition des états non désirés

Le modèle d'anticipation proposé au chapitre 3 repose sur l'identification des états non désirés. Dans l'exemple de la section 3.3.4, nous avons caractérisé les états non désirés d'un conducteur comme des entités associées à des variables dont le domaine est vide (le domaine initial considéré étant un intervalle de temps allant de l'instant courant jusqu'à l'infini). Nous approfondissons ici la définition des états non désirés d'un agent et présentons les solutions implémentées dans ARCHISIM.

L'objectif étant d'éviter les interblocages en carrefour, deux problématiques se posent :

- Il n'est pas concevable d'implémenter des intervalles de temps ouverts et non-bornés $([1, +\infty[)$ puisque l'anticipation est un processus limité dans le temps. Il faut donc choisir une taille de domaines acceptable et définir la durée de blocage que les conducteurs souhaitent éviter.
- La prise de décision étant individuelle, la question de la nature des états non désirés se pose : doivent-ils être ego-centrés ou doivent-ils prendre en compte l'aspect collectif d'une situation de trafic ? Autrement dit un conducteur doit-il anticiper ses propres blocages ? ou doit-il également prendre en compte les éventuels blocages subis par d'autres véhicules ?

4.1.1.1 Durée des interblocages et taille des domaines

L'implémentation du modèle d'anticipation des interblocages présenté au chapitre 3 nécessiterait de considérer des domaines de tailles finies (ie bornés et non-ouverts). Par conséquent la détection d'un interblocage infini en tant que tel (les véhicules sont mutuellement bloqués et aucun d'entre eux ne peut plus avancer) n'est pas possible en utilisant les algorithmes de propagation de contraintes classiques.

Une telle détection nécessite une analyse de la topologie du réseau de contraintes. Une manière de détecter un interblocage infini serait d'utiliser par exemple un algorithme de recherche de cycle dans les graphes. Dans le cadre de notre application, un conducteur tente généralement d'éviter les interblocages infinis mais également les interblocages de durée moyenne. Considérer des domaines de taille finie est donc suffisant pour notre application puisque cela permettra de détecter tous les interblocages dont la durée sera supérieure à un certain seuil et par conséquent les interblocages infinis.

A notre connaissance il n'existe pas d'étude en psychologie de la conduite précisant la durée maximale qu'un conducteur réel accepte de rester bloqué en carrefour. Lors de notre implémentation, nous nous sommes donc fixés une valeur t_{max} paramétrable et pouvant varier entre 10 et 20 secondes. Les domaines utilisés dans notre application sont donc des intervalles de temps allant de $t + 1$ à $t + t_{max}$ (t étant l'instant courant) et représentés sous la forme d'entiers.

4.1.1.2 Vision ego-centrée versus collective des états non désirés

Dans le cadre de la simulation de trafic en carrefour, il existe deux manières de voir les états non désirés. Un conducteur simulé peut ainsi faire preuve d'individualisme en ayant une vision ego-centrée

des états non désirés. Il ne considère alors que ses propres interblocages comme états non désirés sans tenir compte du préjudice pouvant être infligé aux autres véhicules. A l'inverse un conducteur simulé peut adopter une vision plus collective et prendre en compte non seulement ses propres interblocages mais aussi ceux des autres véhicules présents dans le carrefour.

Notre algorithme d'anticipation permet d'utiliser indifféremment l'une ou l'autre des deux approches. Pour une vision ego-centrée, la liste des états non désirés est ainsi juste constituée du domaine vide de l'agent en train d'anticiper. Pour une vision plus collective, il suffit d'ajouter à la liste des états non désirés l'ensemble : $\{\text{dom}(a) \neq \emptyset, a \in A\}$.

Le choix entre ces deux approches est ici encore arbitraire puisqu'il n'existe pas non plus d'étude en psychologie démontrant les tendances d'un conducteur réel pour l'une ou l'autre des deux visions. Nous pouvons simplement constater que dans certaines situations de trafic, il n'est pas suffisant pour un conducteur simulé de considérer uniquement ses propres interblocages.

Considérons la situation de la figure 4.1 dans laquelle deux carrefours à feux A et B sont relativement proches. Une remontée de queue provoquée par le feu du carrefour A induit une occupation de l'espace au centre du carrefour B . Si le véhicule x décide d'avancer, il bloque le flux de véhicules provenant du carrefour A . Le flux de véhicules arrivant du nord dans le carrefour A peut se retrouver bloqué si plusieurs véhicules tournent à gauche. Autrement dit le fait que le véhicule x avance peut avoir pour conséquence de bloquer totalement les deux carrefours A et B .

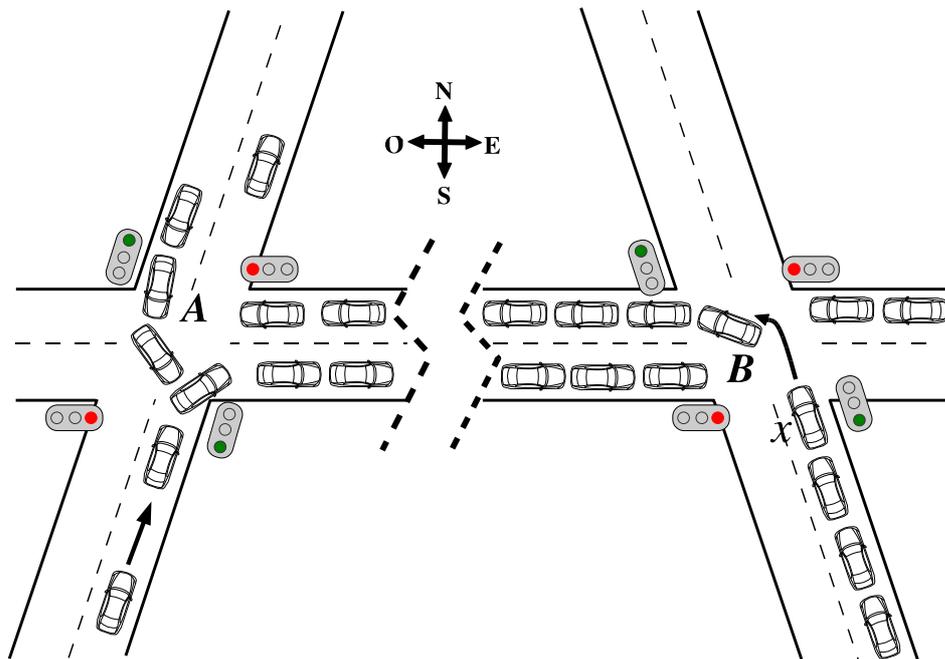


FIGURE 4.1 – Remontée de queue entre deux carrefours

Une telle situation ne peut être évitée si le véhicule x n'anticipe que ses propres interblocages. En effet, suivant la distance séparant A et B , la perception de x ne lui permet pas forcément de se représenter mentalement la situation de trafic dans le carrefour A . Par conséquent, il ne peut pas inférer tous les effets de ses actions et en particulier détecter l'interblocage précédemment décrit.

Tout en restant non détecté, cet interblocage peut tout de même être évité si le véhicule x considère comme état non désiré le fait que tout véhicule soit bloqué durant un temps assez long. En déterminant les effets directs et indirects de son action Go , le véhicule x peut ainsi évaluer que les véhicules venant du carrefour A seront bloqués pour des temps supérieurs à 20 secondes. L'agent x élimine ainsi l'action Go de sa liste d'actions possibles et choisit ainsi de s'arrêter bien que le feu soit vert.

Cet exemple montre bien que la manière de choisir les états non désirés n'est pas anodine et constitue un paramètre essentiel du comportement des agents. Dans notre application, nous avons paramétré ce choix en fonction de la densité de véhicules présents en carrefour : pour une densité élevée, les véhicules anticipent leurs propres blocages ainsi que ceux qu'ils peuvent infliger aux autres véhicules.

4.1.1.3 Incidence sur la complexité

Les différents choix que nous venons d'exposer ont des répercussions directes sur la complexité de l'algorithme d'anticipation proposé au chapitre 3. Pour rappel, la complexité temporelle théorique établie dans le pire des cas était égale à $(|R|d^3(|LA| + |LE|) + |I_{\mathcal{A}}| + |\mathcal{P}|)$ avec :

- $|R|$ le nombre de relations exprimées dans la représentation mentale d'un agent,
- d la taille maximum des domaines utilisés dans la représentation,
- $|LA|$ le nombre d'actions possibles d'un agent,
- $|LE|$ le nombre d'états non désirés,
- $|I_{\mathcal{A}}|$ le nombre d'agents perçus,
- $|\mathcal{P}|$ le nombre de propriétés décrivant les agents perçus.

Dans le cadre de notre application, nous pouvons fixer la valeur de certains de ces paramètres. L'architecture du serveur de vision d'ARCHISIM fait que chaque agent reçoit une description de son environnement local sous la forme d'une liste de vues (une vue étant un agent ou un élément de l'infrastructure, section 1.3.2.1). Dans sa version actuelle, le serveur de vision renvoie à chaque agent une liste comprenant au maximum 100 éléments, par conséquent nous avons : $|I_{\mathcal{A}}| = 100$ dans le pire des cas.

Le mécanisme de coordination présenté au chapitre 2 ne gère pour l'instant que l'accélération longitudinale, ce qui revient pour chaque agent en approche d'un carrefour à ne considérer que deux actions possibles : Go et $Stop$. Nous pouvons donc fixer : $|LA| = 2$. De la même manière, le nombre d'états non désirés est limité au nombre d'agents présents dans la représentation dans le pire des cas : $|LE| = |A|$.

$|A|$ et $|\mathcal{P}|$ restent donc les deux termes indéterminés dans l'expression de la complexité temporelle. Par hypothèse sur notre modèle, le nombre d'agents $|A|$ est borné par $|I_{\mathcal{A}}|$. Dans le cadre de notre application au trafic routier, $|A|$ va dépendre de la taille de la zone considérée par les agents pour anticiper.

4.1.2 Distance d'anticipation

L'algorithme de coordination présenté au chapitre 2 n'est actif qu'en approche et à l'intérieur du carrefour : une zone de coordination est définie autour des carrefours. De la même manière, il convient de définir une zone dans laquelle les agents anticiperont.

Compte tenu des contraintes en terme de décélération et de freinage, l'algorithme de coordination est actif à partir d'une certaine distance du carrefour. Autrement dit, l'algorithme est activé bien avant les limites du carrefour de manière à ce que les véhicules aient une distance²¹ suffisante pour pouvoir s'arrêter [Champion, 2003].

Nous n'avons pas jugé nécessaire d'activer l'algorithme d'anticipation en même temps que l'algorithme de coordination. Ce choix est justifié par plusieurs raisons :

- L'objectif de l'anticipation est la prévention des interblocages. L'algorithme proposé repose sur une analyse du contexte de trafic à l'intérieur du carrefour. Or comme nous l'avons montré dans les chapitres 1 et 2, ce contexte est extrêmement dynamique et changeant. Il est donc peu pertinent que les agents anticipent par rapport à un contexte qui sera totalement différent lorsqu'ils seront aux limites du carrefour.
- La minimisation de la zone d'anticipation peut permettre de réduire les temps de calculs : plus la zone sera petite, moins d'agents exécuteront l'algorithme.
- Dans la réalité, un conducteur ne prévoit pas les éventuels interblocages d'un carrefour 20 ou 50 mètres en amont de celui-ci. La pratique courante est plutôt d'anticiper ces interblocages juste avant d'entrer dans le carrefour.

Pour toutes ces raisons, nous avons introduit une distance d'anticipation définissant l'espace dans lequel les agents anticipent. Cette distance est notée dA sur la figure 4.2 et permet de définir autour de l'espace intérieur du carrefour quatre zones d'approche.

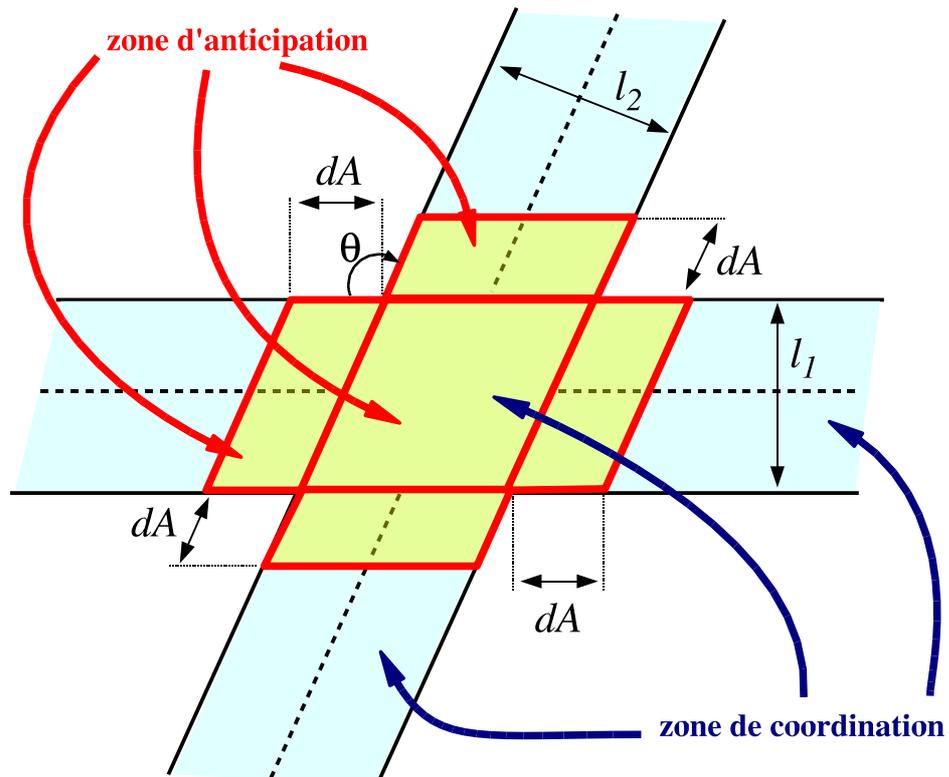


FIGURE 4.2 – Zone d'anticipation

21. à titre indicatif cette distance est actuellement fixée dans ARCHISIM à 20 mètres

Seuls les agents se trouvant dans cet espace seront susceptibles d'exécuter l'algorithme de coordination. De la même manière seuls les agents présents dans cet espace seront inclus dans les représentations mentales des agents anticipant.

4.1.2.1 Incidence sur la complexité

La complexité de l'algorithme d'anticipation va donc dépendre de la taille de cette espace. En introduisant une densité de véhicules en carrefour (c'est-à-dire un nombre de véhicules par unité de surface), nous pouvons exprimer le terme $|A|$.

L'aire de la zone d'anticipation est égale à :

$$\frac{l_1 l_2}{\sin \theta} + 2 \times dA \times (l_1 + l_2)$$

Maximisons cette expression en posant $l = \max(l_1, l_2)$:

$$\frac{l_1 l_2}{\sin \theta} + 2 \times dA \times (l_1 + l_2) \leq \frac{l^2}{\sin \theta} + 4 \times dA \times l$$

Posons δ la densité de véhicules par unité de surface. Le nombre de véhicules dans la zone d'anticipation peut être obtenu en multipliant l'aire de la zone par δ . Nous pouvons ainsi obtenir une borne supérieure plus précise de $|A|$:

$$|A| \leq \min \left(|I_{\mathcal{A}}|, \delta \left(\frac{l^2}{\sin \theta} + 4 \times dA \times l \right) \right)$$

Une manière de réduire les coûts de calculs en pratique est donc de minimiser la distance dA , les termes δ et l étant dépendant de la simulation et ne pouvant par conséquent être minimisés. Expérimentalement cette valeur dA a été fixée à 5 mètres, ce qui signifie que pour un carrefour de taille moyenne un agent anticipera la situation à partir des véhicules présents dans le carrefour et des premiers véhicules situés sur chaque branche du carrefour.

4.1.3 Choix de l'algorithme de propagation

Nous abordons maintenant le choix de l'algorithme de propagation. Nous détaillons dans un premier temps les algorithmes classiques et incrémentaux de propagation de contraintes. Puis, nous donnons quelques critères de choix généraux avant de détailler à proprement dit les choix faits pour ARCHISIM et les différentes optimisations considérées.

4.1.3.1 Algorithmes classiques

Les algorithmes classiques de propagation consistent à parcourir le réseau de contraintes et pour chaque arc parcouru à simplifier les deux domaines associés. La figure 4.3 présente l'un des premiers algorithmes de propagation introduit dans la littérature [Mackworth, 1977a].

La procédure *revise* prend en entrée un arc (X, Y) et en vertu de la propriété d'arc-consistance (section 3.3.1.3) permet de réduire le domaine de X . Chaque arc ainsi visité est placé dans une queue Q lorsque il y a réduction de domaine.

```

function revise (in : arc (X, Y))
  efface ← faux
  for each u ∈ dom(X) do
    if ∃ v ∈ dom(Y)/(X, Y) soit consistante then
      dom(X) ← dom(X) - {u}
      efface ← vrai
    end
  end
  renvoyer efface

procedure AC-3 (in : CSP G)
  Q ← {(X, Y) ∈ arcs(G)/X ≠ Y}
  change ← faux
  while Q ≠ ∅ do
    choisir un arc (X, Y) dans Q
    Q ← Q - {(X, Y)}
    if revise((X, Y)) then
      Q ← Q ∪ {(Z, X)/(Z, X) ∈ arcs(G), Z ≠ X, Z ≠ Y}
    end
  done

```

FIGURE 4.3 – Algorithme AC-3 [Mackworth, 1977a]

La complexité d'AC-3 est en $O(ed^3)$, e étant le nombre de contraintes et d la taille du plus grand domaine. D'autres algorithmes ont été introduits dans la littérature de manière à améliorer cette complexité temporelle. Cette amélioration passe par une complexification des algorithmes et l'introduction de la notion de support.

Dans un réseau de contraintes (X, D, C) , le support d'une valeur a pour une variable x_i est l'ensemble des valeurs $b \in \text{dom}(x_j)$ $x_j \in X - \{x_i\}$ telles que (a, b) satisfasse l'ensemble des contraintes portant sur les arcs (x_i, x_j) . Lorsque le support d'une valeur est vide, elle peut être supprimée du domaine auquel elle appartient. Si cette valeur appartenait elle-même à d'autres supports, il est alors nécessaire de mettre à jour ceux-ci. Cette propriété est utilisée dans différents algorithmes. Dans [Mohr et Henderson, 1986], les auteurs présentent AC-4 dans lequel un compteur de supports est maintenu par valeur et par arc. Chaque suppression d'une valeur dans un domaine s'accompagne d'une décrémentation du compteur. Si le compteur d'une valeur est égal à zéro, cela signifie que la valeur peut être supprimée.

La complexité temporelle d'AC-4 est en $O(ed^2)$. En revanche, de par la taille des structures de données qu'il manipule, AC-4 est très coûteux en terme d'espace mémoire. Des versions plus récentes d'algorithme utilisant la notion de support ne sauvegardent pas tous les supports d'une valeur et se contentent d'en mémoriser le nombre [Bessière, 1994].

4.1.3.2 Algorithmes incrémentaux

Les algorithmes incrémentaux ont été introduits pour des réseaux de contraintes dynamiques, c'est-à-dire des réseaux de contraintes modifiés par l'ajout et la suppression de contraintes. Ils améliorent les algorithmes classiques, notamment lors de la suppression de contraintes, en conservant le bénéfice des simplifications déjà effectuées.

Le premier algorithme incrémental (dnAC-4) proposé est une adaptation de l'algorithme AC-4 [Bessière, 1992]. Il utilise lui aussi des compteurs indiquant le nombre de supports d'une valeur. dnAC-4 utilise les structures de données suivantes :

- une table de booléens D représentant le domaine courant
- un ensemble cpt permettant de mémoriser pour chaque paire arc/valeur $((i, j), a)$ le nombre de supports dans le domaine courant D de (i, a) sur la contrainte c_{ij}
- un ensemble S de supports : $S_{((i,j),a)} = \{b \in \text{dom}(j) / a \text{ sur } i \text{ supporte } b \text{ sur } j\}$
- deux queues de propagation SL et RL respectivement pour les suppressions et rajouts de valeurs
- un ensemble $justif$ est donc défini comme suit :

$$(justif_{(i,a)} = j) \equiv ((i, a) \notin D \text{ et la contrainte } justificiant \text{ le retrait de } (i, a) \text{ est } c_{ij})$$

La figure 4.4 donne l'implémentation de dnAC-4 qui gère respectivement les restrictions et les relaxations de contraintes au moyen des procédures *Add* et *Relax*.

Lors de l'ajout d'une contrainte c_{ij} , l'algorithme commence par mettre à jour les ensembles S et cpt (procédure *Init-Add*). Il crée ensuite les nouvelles entrées dans les deux ensembles S et cpt et ce pour tout $a \in \text{dom}(x_i)$ et $b \in \text{dom}(x_j)$: $cpt_{((i,j),a)}$, $cpt_{((j,i),a)}$, $S_{((i,j),a)}$ et $S_{((j,i),a)}$.

Lors de la création de ces nouvelles entrées, les paires arc-valeur dont le compteur est nul sont placées dans SL . Ces paires sont ensuite supprimées lors de l'appel de la procédure *Propag-suppress*.

Lors de la suppression d'une contrainte c_{km} , l'algorithme remet en place toutes les valeurs qui avaient été supprimées lors de la révision de l'arc (k, m) (procédure *Init-Propag-Relax*). Parmi ces valeurs, certaines peuvent avoir un nombre de supports nuls. Elles sont donc éliminées lors de l'appel de la procédure *Propag-Suppress*.

La complexité temporelle de dnAC-4 est la même qu'AC-4 mais l'algorithme conserve également les inconvénients de l'occupation en mémoire des différentes structures de données. Tout comme pour AC-4, une version moins coûteuse en terme de complexité spatiale a été proposée : dnAC-6 [Debruyne, 1995].

4.1.3.3 Critères de choix généraux

Notre algorithme d'anticipation consiste en une évaluation des effets d'actions reposant sur :

- la prise en compte des effets directs d'une action par modification d'une représentation M de l'environnement
- l'inférence d'effets indirects par propagation

Il peut donc être intéressant de considérer les algorithmes de propagation incrémentaux.

Dans l'algorithme générique présenté au chapitre 3, une copie de la représentation M est faite avant sa modification (ajout ou suppression d'une relation). Cette copie permet de restaurer la représentation M après détermination des effets de l'action courante.

<pre> procedure Init for each $i \in X$ do for each $a \in \text{dom}(i)$ do $D_{(i,a)} \leftarrow \text{vrai}; \text{justif}_{(i,a)} \leftarrow \text{NULL}$ procedure Add(in : contrainte c_{ij}) $C \leftarrow C \cup \{c_{ij}\}; SL \leftarrow \emptyset;$ Init-Add($(i, j), SL$); Init-Add($(j, i), SL$); Propag-Suppress(SL) procedure Init-Add(in : arc (i, j), in/out : SL) for each $b \in \text{dom}(x_j)$ do $S_{((j,i),b)} \leftarrow \emptyset;$ for each $a \in \text{dom}(x_i)$ do $total \leftarrow 0;$ for each $b \in \text{dom}(x_j)$ do if $((i, a)(j, b)) \in c_{ij}$ then if $D_{(j,b)}$ then $total \leftarrow total + 1;$ Append($S_{((j,i),b),a}$); $cpt_{((i,j),a)} \leftarrow total;$ if $cpt_{((i,j),a)} = 0$ then Append($SL, ((i, j), a)$) procedure Propag-Suppress(in/out : SL) while $SL \neq \emptyset$ do prendre $((i, m), a)$ dans SL; if $D_{(i,a)} \wedge cpt_{((i,m),a)} = 0$ then $\text{justif}_{(i,a)} \leftarrow m; D_{(i,a)} \leftarrow \text{faux};$ for each $j/c_{ij} \in C$ do for each $b \in S_{((i,j),a)}$ do $cpt_{((j,i),b)} \leftarrow cpt_{((j,i),b)} - 1;$ if $cpt_{((j,i),b)} = 0$ then </pre>	<pre> Append($SL, ((j, i), b)$) procedure Relax(in : contrainte c_{km}) $SL \leftarrow \emptyset;$ Init-Propag-Relax(c_{km}, SL); Propag-Suppress(SL) procedure Init-Propag-Relax (in : contrainte c_{km}, in/out : SL liste) $RL \leftarrow \emptyset;$ for each $a \in \text{dom}(k)$ do if $\text{justif}_{(k,a)} = m$ then Append($RL, (k, a)$); $\text{justif}_{(k,a)} \leftarrow \text{NULL}$ for each $b \in \text{dom}(m)$ do if $\text{justif}_{(m,b)} = k$ then Append($RL, (m, b)$); $\text{justif}_{(m,b)} \leftarrow \text{NULL}$ Enlever c_{km} de C while $RL \neq \emptyset$ do prendre (i, a) dans RL; $D_{(i,a)} \leftarrow \text{vrai}$ for each $j/c_{ij} \in C$ do for each $b \in S_{((i,j),a)}$ do $cpt_{((j,i),b)} \leftarrow cpt_{((j,i),b)} + 1;$ if $\text{justif}_{(j,b)} = i$ then Append($RL, (j, b)$); $\text{justif}_{(j,b)} \leftarrow \text{NULL}$ if $cpt_{((i,j),a)} = 0$ then Append($SL, ((i, j), a)$) </pre>
--	---

FIGURE 4.4 – Algorithme dnAC-4 [Bessière, 1992]

Cette opération peut être évitée en utilisant un algorithme de propagation incrémental. Il suffit alors de travailler directement sur M et non sur une copie. Cela nécessite alors de supprimer en fin de boucle les relations qui ont été ajoutées et effectuer une nouvelle propagation de manière à remettre M dans son état initial.

Dans le cas de l'utilisation de dnAC-4, l'algorithme d'anticipation peut être modifié comme suit :

```

procedure anticipe(in : ListeActions LA, ListeEtatsIndesires LE,
                    M(A, R, D), IE
                    sortie : ListeActions LA)
begin
  for each action ∈ LA do
    r ← calculeEffets(a, IE)
    Add(r)
    existe ← rechercheEtatsIndesires(M, LE)
    if existe then
      LA ← LA − {action}
    end
    Relaxe(r)
  end
end

```

L'ajout et la suppression des effets directs se font grâce aux fonctions *Add* et *Relax* de dnAC-4. Ces fonctions prennent en charge à la fois la mise à jour du réseau mais également la propagation. L'ajout lors de la construction doit également se faire par la fonction *Add* de manière à initialiser les supports des valeurs.

Bien que *M* soit clairement un réseau de contraintes dynamique eu égard aux nombreux ajouts et suppressions de relations, l'utilisation d'un algorithme incrémental n'est pas forcément toujours pertinent. Le choix entre un algorithme classique et un algorithme incrémental peut être guidé par deux critères :

- la taille de *M*, c'est-à-dire le nombre d'agents $|A|$ considéré dans la représentation et le nombre de relations $|R|$ entre eux,
- le taille de la liste $|LA|$ autrement dit le nombre d'actions à considérer dans le processus d'anticipation.

Pour des représentations *M* de petite taille, le temps de copie peut être considéré comme négligeable comme lors de notre analyse de complexité à la fin du chapitre 3. Cela est d'autant plus vrai qu'il n'est pas forcément nécessaire de copier toute la représentation, seule une copie de l'ensemble *D* des domaines est nécessaire, ce qui en terme de complexité équivaut à une opération réalisable en temps linéaire. A l'inverse, l'appel à la fonction *Relaxe* est en $O(|R|d^2)$ avec $d = \max_{a_i \in A} (dom(a_i))$.

Même dans une version optimisée, l'utilisation d'un algorithme incrémental oblige à maintenir en mémoire un ensemble de structures de données conséquentes. Lorsque plusieurs agents anticipent en même temps, cela peut constituer un surcoût en terme d'occupation mémoire non négligeable.

Le second critère qui doit être pris en considération est le nombre d'actions qu'un agent doit examiner pour anticiper. Autrement dit, le nombre de fois que *M* sera modifié par l'ajout ou la suppression de relations. Dans le cas d'utilisation d'un algorithme classique, l'ajout de relation ne pose pas véritablement de problème puisque le travail effectué par la première propagation peut être conservé. En revanche, lors de la suppression il est nécessaire de réinitialiser tous les domaines. L'utilisation d'un algorithme incrémental permet d'éviter cette opération gérant ainsi de manière transparente aussi bien l'ajout que la suppression de relations.

L'intérêt à utiliser un algorithme incrémental dépend donc du nombre de suppressions de relations effectuées sur *M*. Ce nombre est difficile à déterminer *a priori* puisqu'il dépend de l'application et

de la dynamique de celle-ci ; il ne peut donc être évalué qu'empiriquement. En revanche, le nombre d'actions $|LA|$ à considérer durant l'anticipation peut être un indicateur : plus le nombre d'actions à évaluer est grand, plus il y a de chance d'effectuer des suppressions de relations lors de l'évaluation des effets directs et indirects.

Dans le cas général et en première approche, nous préconisons donc l'utilisation d'algorithmes de propagation classiques. L'utilisation d'algorithmes de propagation incrémentaux ne peut se révéler plus efficace que pour un nombre d'actions $|LA|$ relativement grand.

Dans l'application à ARCHISIM, nous avons utilisé une adaptation de l'algorithme AC-3. Compte tenu de la nature des relations utilisées et des domaines, nous avons pu appliquer quelques optimisations permettant de réduire la complexité en pratique.

4.1.3.4 Mise en œuvre de l'anticipation et optimisations

Les algorithmes classiques comme AC-3 sont des algorithmes génériques ne tenant pas compte de la sémantique des contraintes et considérant celles-ci comme un ensemble de valeurs interdisant l'affectation des variables. Suivant la sémantique des contraintes et la nature des domaines, les méthodes de propagation peuvent être améliorées.

L'anticipation des interblocages dans ARCHISIM est basée sur l'utilisation de domaines temporels et de relations de blocages. La sémantique donnée à ces relations induit, pour les agents impliqués, des phases d'arrêt et de déplacement. Celles-ci sont inférées par l'application du principe d'arc-consistance sur les relations et les domaines.

Il s'agit donc ici de manipuler des intervalles de temps qui peuvent dans certains cas être non convexes (union d'intervalles convexes). Le traitement des relations de blocages repose donc sur des opérations de manipulation d'intervalles.

La relation bph peut ainsi s'exprimer comme un décalage entre deux intervalles de temps correspondant au temps de redémarrage du véhicule bloqué :

$$bph(x, y) \equiv \left(\text{dom}(x) = [t_y + Tr(x), t'_y + Tr(x)] \right)$$

avec $Tr(x)$ le temps de redémarrage de x , t_y et t'_y respectivement les bornes inférieures et supérieures de $\text{dom}(y)$. La propagation de bph est illustrée par la figure 4.5.

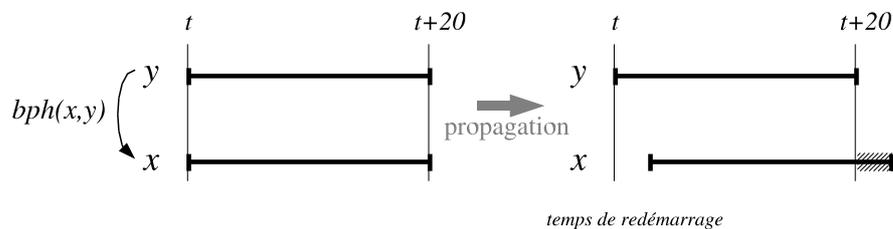


FIGURE 4.5 – Exemple de propagation de la relation bph

La relation $bpha$ peut être vue comme l'union entre l'intervalle de temps nécessaire pour que le blocage devienne effectif et l'intervalle du véhicule bloquant décalé d'une durée égale au temps de

redémarrage :

$$bpha(x, y) \equiv \left(\text{dom}(x) = \left([t_x, t_x + Tac(x, y)] \cup [t_y + Tr(x), t'_y + Tr(x)] \right) \right)$$

avec $Tac(x, y)$ le temps mis par x pour atteindre y et t_x la borne inférieure de $\text{dom}(x)$. La propagation de $bpha$ est illustrée par la figure 4.6.

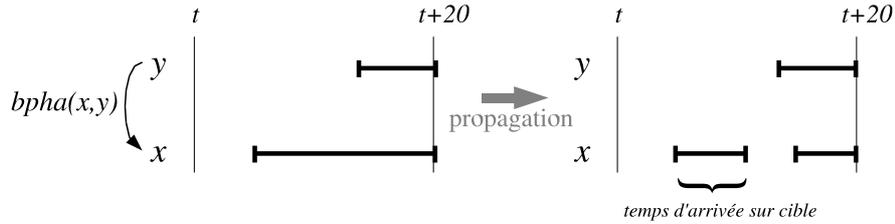


FIGURE 4.6 – Exemple de propagation de la relation $bpha$

La relation bpr , quant à elle, se définit comme la différence entre l'intervalle de temps nécessaire au véhicule bloquant pour dépasser son point de conflit et son propre intervalle

$$bpr(x, y) \equiv \left(\text{dom}(x) = \left([t_x, t'_x] - [t_y, t_y + Tdc(y, x)] \right) \right)$$

avec $Tdc(y, x)$ le temps mis par y pour dépasser son point de conflit avec x . La propagation de bpr est illustrée par la figure 4.7.

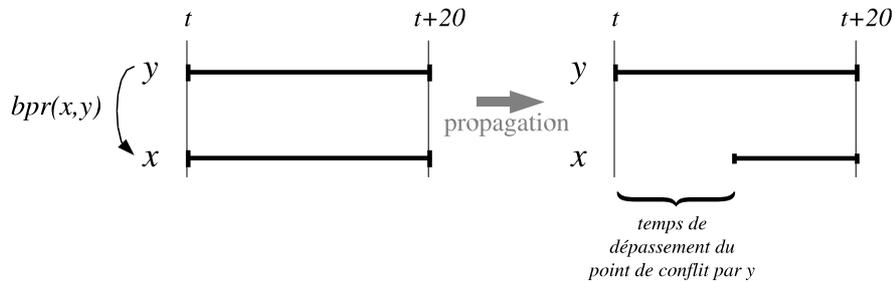


FIGURE 4.7 – Exemple de propagation de la relation bpr

La fonction *revise* a donc été adaptée de manière à prendre en compte les propagations spécifiques des trois relations de blocage de notre modèle. L'ordre dans lequel sont placées les relations dans la liste R influe sur la rapidité de propagation en limitant le nombre de remises en cause des domaines déjà réduits. Voyons sur un exemple comment optimiser le traitement d'une relation.

Considérons la situation de la figure 4.8 dans laquelle deux véhicules x et y se suivent et arrivent à une intersection où ils doivent laisser la priorité à z . Du point de vue d'un observateur, la situation peut se modéliser comme suit :

$$D = \begin{cases} \text{dom}(x) = [1, 20] \\ \text{dom}(y) = [1, 20] \\ \text{dom}(z) = [1, 20] \end{cases} \quad R = \{bph(x, y), bpr(y, z)\}$$

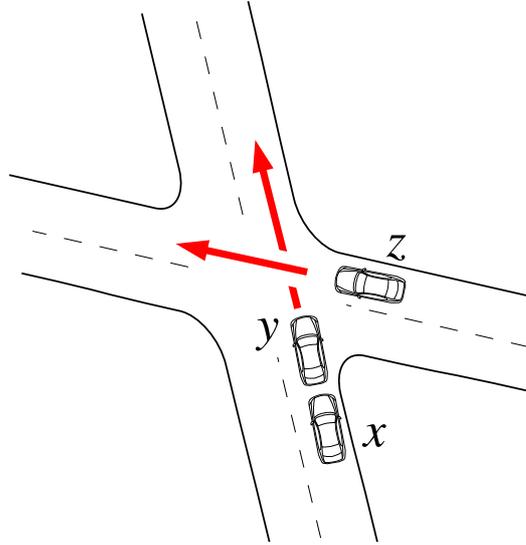


FIGURE 4.8 – Exemple de propagation des relations

Si l'agent x traite en premier la relation $bph(x, y)$, il peut réduire $\text{dom}(x)$ à $[2, 20]$ en supposant $Tr(x) = 1$. La seconde relation de R , $bpr(y, z)$ permet de réduire $\text{dom}(y)$ à $[6, 20]$ avec $Tdc(y, x) = 5$. La modification de $\text{dom}(y)$ remet en cause $\text{dom}(x)$, ce qui oblige à appliquer de nouveau la fonction *revise* sur la relation $bph(x, y)$. Au final, la propagation aura nécessité trois fois l'appel à la fonction *revise*. En ordonnant l'ensemble R de cette manière : $R = \{bpr(y, z), bph(x, y)\}$, la propagation ne nécessite plus que deux appels à la fonction *revise*.

A la construction de chaque représentation mentale, les relations de type de bp_- (bph , $bpha$ ou bpr) sont donc ordonnées de manière à maximiser les chaînes dans le réseau :

$$\{bp_-(a_1, a_2), bp_-(a_2, a_3), \dots, bp_-(a_i, a_j), bp_-(a_j, a_k), \dots\}$$

4.1.4 Complexité en pratique

Nous récapitulons ici les différentes optimisations présentées et évaluons leur impact sur la complexité de l'algorithme d'anticipation. La propagation d'une relation étant basée sur des opérations issues de l'arithmétique des intervalles, chaque appel à la fonction *revise* ne nécessite plus d^2 comparaisons (d étant la taille maximale des domaines) puisque le calcul d'un intervalle se fait par manipulation de ses bornes. Les propagateurs introduits dans la section précédente requièrent en moyenne une dizaine d'opérations élémentaires. La complexité de l'algorithme de propagation se trouve ainsi réduite et est en $O(10|R|)$.

La complexité de l'algorithme d'anticipation vaut donc dans le cadre de notre application :

$$(10|R| (|LA| + |LE|) + |I_{\mathcal{A}}| + |\mathcal{P}|)$$

En intégrant les différentes analyses de complexité décrites dans cette section, nous pouvons donner une complexité pratique de l'algorithme d'anticipation :

$$(10|R| (2 + |A|) + |I_{\mathcal{A}}| + |\mathcal{P}|)$$

A partir de ce résultat, nous pouvons évaluer numériquement la complexité. Le nombre de relations $|R|$ est fonction du nombre d'agents $|A|$ présents dans la représentation. Dans notre cas, chaque agent est impliqué en moyenne dans 3 relations, $|R|$ est donc égal à $\frac{3}{2}|A|$. Évaluons $|A|$ en pratique grâce aux résultats obtenus en section 4.1.2.1. La largeur moyenne d'une chaussée en zone urbaine est de 6 mètres et l'angle entre les deux axiales d'un carrefour est généralement proche des 90 degrés. Cela nous donne donc une zone d'anticipation dont l'aire est égale à environ $150m^2$. La densité de véhicules δ est généralement comprise entre 0 et 0,07 (0,07 correspondant à une situation de trafic très dense). En considérant une densité moyenne égale à 0,04 véhicules par m^2 , on obtient : $|A| = 7$. Nous avons vu que dans ARCHISIM, $|I_{\mathcal{A}}| < 100$; reste donc à donner une valeur à $|\mathcal{P}|$. Parmi les propriétés fournies par le mécanisme de vision, une dizaine sont utilisées par un agent pour construire sa représentation mentale. $|A|$ étant égal à 7, cela nous donne donc : $|\mathcal{P}| = 70$.

Au final, la complexité est égale à $40|A| + |I_{\mathcal{A}}| + |\mathcal{P}|$ et évaluée numériquement à environ 450 opérations élémentaires. Une étude empirique nous a permis de constater que l'ajout de capacités d'anticipation pouvait aller jusqu'à doubler les temps de calculs requis pour la prise de décision de chaque agent. Le temps de calcul d'une simulation pour un trafic relativement dense se trouve donc augmenté mais reste suffisamment court pour permettre une simulation en temps réel. En condition de circulation fluide, les temps de calculs demeurent quasiment inchangés.

4.2 Intégration des comportements opportunistes et anticipatifs à ARCHISIM

Dans cette seconde section, nous situons notre apport au sein d'ARCHISIM. Nous détaillons dans un premier temps l'intégration de notre modèle de comportement opportuniste et anticipatif dans le cycle de décision de l'agent. Puis, en seconde partie, nous développons le couplage entre l'algorithme de coordination existant et notre algorithme d'anticipation.

4.2.1 Cycle de décision de l'agent

L'introduction de comportements opportunistes se situe au niveau de l'interprétation du contexte obtenu par un agent suite à une requête de vision. Cela consiste en deux étapes successives : la détermination des relations de priorités et la recherche des agents avec lesquels il peut se coordonner. Cette première partie de notre apport vient remplacer les deux premières étapes de l'algorithme de coordination existant [Champion, 2003].

La seconde partie de notre apport, à savoir un algorithme d'anticipation permettant d'éviter les interblocages, vient se placer en amont de la coordination. Il permet, avant la coordination, de filtrer la liste des actions possibles et d'éliminer celles conduisant à un interblocage. L'anticipation s'intègre donc entre la phase de décision et la phase d'analyse de l'environnement.

La figure 4.9 résume cette intégration en présentant le cycle "perception-décision-action" d'un agent. L'architecture d'ARCHISIM repose sur une exécution pseudo-parallèle des agents et ne permet donc l'exécution que d'un seul cycle par un agent pour un pas de temps donné. Cela signifie que la phase "perception de l'environnement" marque le début du pas de temps pour l'agent et que la phase "action" correspond à la fin de ce même pas de temps.

Ayant présenté une vue d'ensemble du cycle d'un agent et l'intégration de nos travaux au sein de ce cycle, nous allons maintenant détailler le couplage réalisé entre l'algorithme d'anticipation et de coordination.

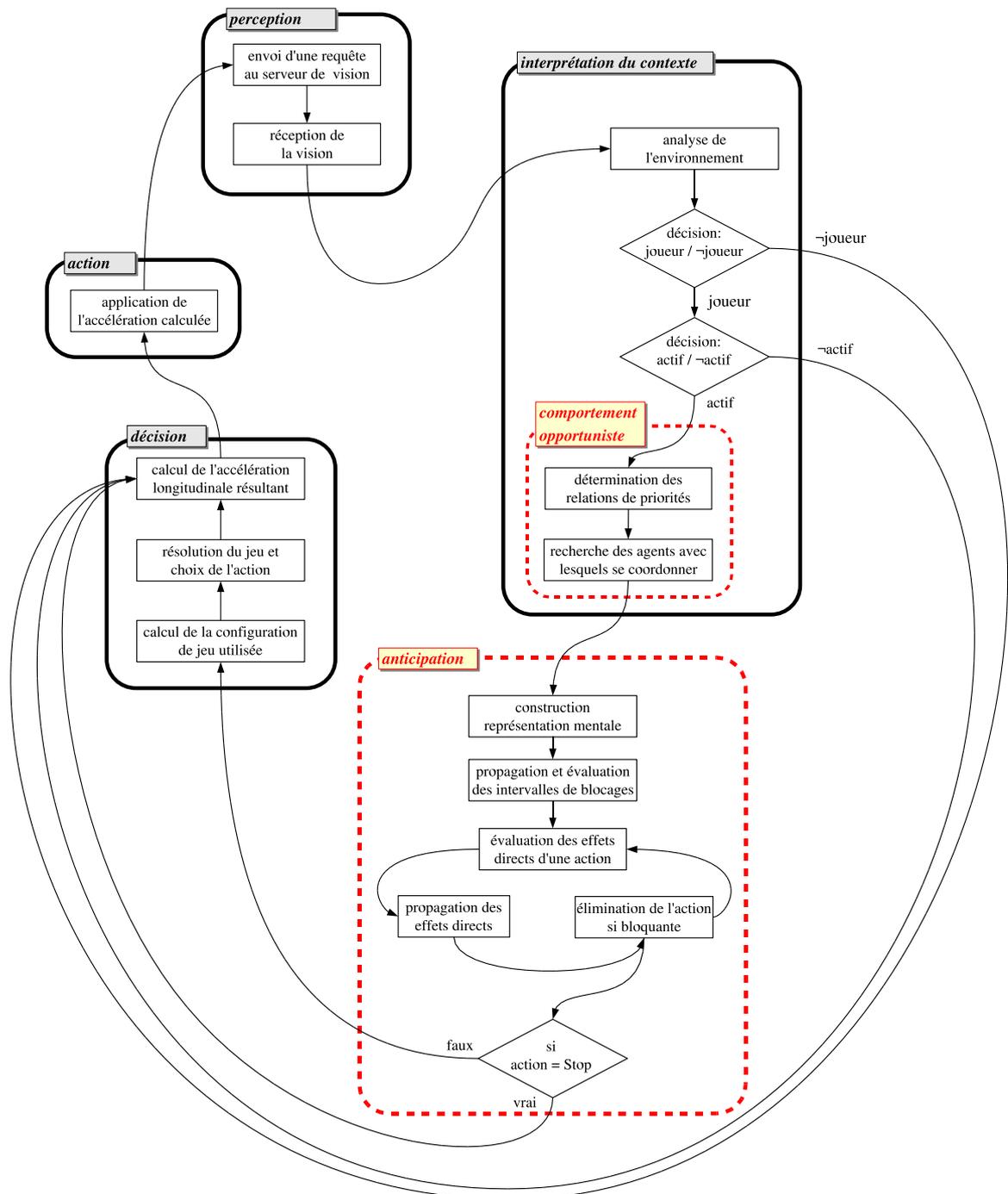


FIGURE 4.9 – Cycle de décision d'un agent

4.2.2 Couplage anticipation / coordination

L'algorithme d'anticipation élimine en amont de la coordination les actions potentiellement interbloquantes. Les actions considérées par l'algorithme de coordination sont pour l'instant au nombre de deux : accélérer (*Go*) / freiner (*Stop*) et permettent de gérer l'accélération longitudinale²². Lorsque l'algorithme d'anticipation élimine l'action *Go*, il ne reste plus que l'action *Stop* comme alternative à l'agent. Un test en sortie de l'algorithme d'anticipation permet donc de passer directement au calcul de la décélération sans passer par les phases de recherche de configuration de jeux et de résolution de conflits (figure 4.9).

L'algorithme de coordination calcule les accélérations et décélérations à partir d'un ensemble de distances : distance au véhicule le plus contraignant, distance au carrefour, distance inter-véhiculaire avec le véhicule précédant (section 2.3.4.3). Lorsqu'un interblocage est détecté, deux cas se présentent :

- soit le véhicule n'est pas encore entré dans le carrefour, auquel cas sa distance d'arrêt utilisée pour le calcul de la décélération est la distance le séparant des limites du carrefour.
- soit le véhicule est déjà à l'intérieur du carrefour, auquel cas il doit s'arrêter le plus rapidement possible, la décélération calculée est alors la décélération maximale autorisée pour ce type de véhicule.

Lorsqu'un véhicule prioritaire décide de s'arrêter pour éviter de créer un interblocage, les autres véhicules viennent à se considérer comme prioritaire au bout d'un certain temps correspondant à leur propre impatience. Ce temps peut se révéler long suivant les caractéristiques personnelles des agents. Pour accélérer le temps de réaction des agents, nous avons renforcé le couplage entre les deux algorithmes en modifiant certaines variables utilisées pour la coordination.

L'algorithme de coordination utilise en effet deux variables booléennes d'état, l'une indiquant si un agent se trouve en situation de carrefour (*joueur* / \neg *joueur*), l'autre indiquant si l'agent va effectivement prendre part à la coordination (*actif* / \neg *actif*) (figure 4.9). Pour optimiser la recherche des agents en conflit, ces variables font partie des attributs d'un agent et sont visibles par les autres au même titre que la vitesse ou l'accélération [Champion, 2003]. Un agent se déclarant comme inactif n'est donc pas pris en compte dans les processus de coordination des agents l'entourant.

Pour accélérer le temps de réaction des agents lors de l'anticipation d'un interblocage, nous avons utilisé la variable *actif*. Lorsque l'algorithme d'anticipation détecte un interblocage il vient surcharger à faux la variable *actif* avant de calculer la décélération à appliquer. Les autres agents en situation de carrefour détectent alors que l'agent venant d'anticiper se place en état \neg *actif* et par conséquent ne le considèrent plus dans leur processus de coordination.

Ceci permet d'accélérer grandement le temps de réaction des agents au carrefour. Transposée au monde réel, l'utilisation de cette variable peut s'apparenter au signe de main que les conducteurs font parfois pour indiquer qu'ils ne s'engagent pas et laissent la priorité aux autres. D'un point de vue multi-agent, cela peut s'apparenter à une sorte de communication indirecte : l'agent anticipant laisse percevoir son intention par le biais d'un marqueur visible de l'ensemble des agents.

22. nous verrons dans le chapitre 5 une extension possible de cette liste pour la gestion de l'accélération latérale

4.3 Conclusion

Dans ce chapitre, nous avons développé la mise en œuvre des modèles de comportement opportuniste et anticipatif présentés au chapitre 3. Le modèle générique d'anticipation proposé a été détaillé dans le cadre de son application à ARCHISIM. Les différents choix en terme d'implémentation ont été exposés et pour chacun d'entre eux une analyse a été faite pour évaluer leur impact en terme de complexité algorithmique.

Dans une seconde partie, nous nous sommes focalisés sur l'intégration de nos apports à l'architecture d'ARCHISIM. Notre modèle opportuniste / anticipatif a été situé dans le cycle "perception-décision-action" des agents et nous avons détaillé le couplage entre les algorithmes de coordination et d'anticipation.

Dans le prochain chapitre, nous exposons les différentes expérimentations menées pour évaluer la pertinence de notre modèle opportuniste/anticipatif dans le cadre de la simulation de trafic.

Chapitre 5

Évaluations, validations et perspectives

Sommaire

5.1	Expérimentations sur la base de scenarii	112
5.1.1	Réduction des phénomènes d'oscillations	112
5.1.2	Anticipation et interblocages en carrefour isolé	114
5.1.3	Succession de carrefours	119
5.2	Simulation d'un carrefour réel et validation	122
5.2.1	Démarche	122
5.2.2	Résultats de simulation	124
5.2.3	Reproductibilité de la simulation	127
5.3	Perspectives	127
5.3.1	Perspectives de recherche dans le cadre d'ARCHISIM	128
5.3.2	Perspectives générales	130

Nous abordons dans ce chapitre les différentes expérimentations menées dans le cadre de l'évaluation et de la validation de notre modèle.

La première partie présente une évaluation basée sur l'utilisation de différents scenarii visant à mettre les conducteurs simulés dans des situations de trafic particulières pour lesquelles un comportement bien précis est attendu. Cette première évaluation est basée sur l'observation de comportements locaux et la mesure de variables cinématiques comme la vitesse et l'accélération. Pour chaque scénario, nous comparons notre proposition de comportements opportunistes et anticipatifs avec le comportement issu des travaux antérieurs [Champion, 2003].

Dans la seconde partie de ce chapitre, nous nous intéressons à la validation du trafic résultant des comportements locaux mis en évidence en première partie. L'objectif est d'évaluer l'amélioration apportée par notre contribution sur la validité des situations de trafic obtenues. Pour cela, nous simulons un carrefour réel pour lequel nous disposons de données trafic mesurées sur le terrain. Nous présentons une comparaison entre les débits réels mesurés sur ce carrefour et les débits obtenus en simulation.

La dernière section du chapitre discute des différentes perspectives envisageables à notre travail. Nous y présentons les extensions possibles dans le cadre de la simulation de trafic ARCHISIM, puis nous généralisons en détaillant quelques pistes de recherche.

5.1 Expérimentations sur la base de scenarii

Nos expérimentations consistent à placer des conducteurs simulés dans des situations de trafic particulières de manière à :

- vérifier que le comportement observé est bien celui attendu,
- évaluer l'amélioration par rapport au comportement issu des travaux précédents [Champion, 2003].

Les améliorations présentées portent sur la réduction des phénomènes d'oscillations et du nombre d'interblocages.

5.1.1 Réduction des phénomènes d'oscillations

Les phénomènes d'oscillations sont des problèmes bien connus dans les systèmes multi-agents. Il s'agit le plus souvent d'oscillations de décision : l'agent hésite entre deux actions et choisit alternativement l'une puis l'autre. Ces problèmes d'oscillations se rencontrent de manière générale dans la plupart des approches réactives. Dans [Koren et Borenstein, 1991], les auteurs montrent que les oscillations sont une des principales faiblesses des champs de potentiels utilisés dans la navigation autonome de robots. Certains phénomènes d'oscillations peuvent être émergents : ils apparaissent dans des situations particulières pour lesquelles les agents enchaînent successivement une combinaison d'actions se répétant en cycle [Simonin, 2001].

En simulation, les phénomènes d'oscillations sont un vrai problème car ils confèrent aux entités simulées des comportements d'hésitation. Lorsque ces comportements sont perceptibles pour un observateur extérieur et ne peuvent être justifiés par le contexte, ils contribuent à décrédibiliser le réalisme de la simulation.

Dans le cadre de notre application, des oscillations sont perceptibles avec le comportement issus des travaux antérieurs [Champion, 2003]. Elles résultent d'un manque de stabilité dans :

- le choix des agents avec lesquels se coordonner,
- la perception des priorités.

Cela signifie que dans certains cas, un agent x peut se coordonner avec un autre agent y au pas de temps t , pour ensuite se coordonner avec un agent z au pas $t + 1$ et reconsidérer l'agent y au pas $t + 3$ alors que le contexte est resté globalement inchangé durant cette période de temps.

Dans [Champion, 2003], le choix des agents et la perception des priorités reposent sur des comparaisons entre les temps au conflit ou les distances dans le cas d'une vitesse nulle. Les problèmes d'oscillations dans la reconnaissance des priorités peuvent apparaître lorsque par exemple les vitesses des mobiles varient autour de zéro ou lorsqu'un des deux véhicules décélèrent pour une tierce raison. En simulation, cela se traduit par des phases d'accélération et de décélération se succédant et ne durant que quelques pas de temps. Ces hésitations sont particulièrement flagrantes lorsque le trafic est observé par le biais du visuel 3D (simulateur de conduite).

Nous avons pu constater au travers différents scenarii que les règles d'agrégation proposées au chapitre 3 permettent de réduire ces phénomènes d'oscillations en assurant une plus grande stabilité dans la reconnaissance des priorités. Nous présentons à la figure 5.1 une situation de trafic dans laquelle un conducteur simulé est en approche d'un carrefour sur une branche non prioritaire. Le graphique montre la courbe de vitesse du véhicule et illustre ainsi la phase de décélération résultant de la coordination avec les autres mobiles de l'axe prioritaire.

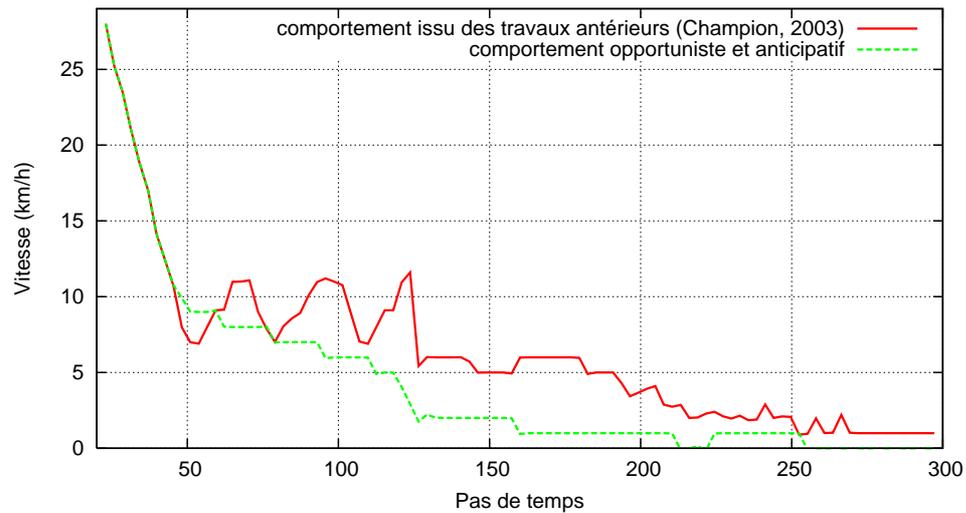


FIGURE 5.1 – Décélération d'un agent en approche d'un carrefour

Ce scénario a été simulé dans un premier temps avec des agents appliquant le comportement issus des travaux antérieurs [Champion, 2003], puis dans un second temps avec des agents appliquant notre proposition. Les deux simulations sont exécutées sur la même machine avec le même pas de temps et une demande de trafic identique (même débit, même temps intervéhiculaire, même vitesse initiale, etc). La mesure de vitesse a été effectuée sur une période d'une trentaine de secondes et débute aux alentours de trente kilomètres/heure.

La courbe en trait plein correspond à la vitesse observée avec le comportement issus des travaux antérieurs [Champion, 2003]. Deux phases d'oscillations sont remarquables. La première se situe entre les pas de temps 50 et 125. Sur cette période, nous pouvons constater que la vitesse oscille autour d'une valeur médiane égale à 8 km/h avec des amplitudes de 5 km/h environ. La seconde phase d'oscillations est moins prononcée et se situe entre les pas de temps 200 et 275. La vitesse passe de 4 à 0 km/h de manière saccadée avec la présence de pics aux pas 240, 260 et 270 indiquant une réaccélération de courte durée.

La courbe en trait pointillé correspond à la vitesse obtenue avec le comportement opportuniste et anticipatif. L'aspect général de la courbe est globalement plus lisse. En particulier, les deux phases d'oscillations précédemment décrites n'apparaissent plus. Entre les pas de temps 50 et 125, nous pouvons ainsi observer une décélération soutenue et continue. De la même manière, l'intervalle de temps allant du pas 200 au pas 275 ne présente plus d'oscillation.

D'une manière générale, il a été constaté que notre approche permettait d'obtenir des comportements moins hésitants pour bon nombre de situations de conduite améliorant ainsi le réalisme local d'un point de vue visuel. L'impact de notre approche au niveau du trafic obtenu sera évalué dans la deuxième partie de ce chapitre. Nous nous focalisons maintenant sur l'évaluation de la partie anticipative de notre approche.

5.1.2 Anticipation et interblocages en carrefour isolé

La partie anticipative de notre approche a été évaluée sur sa capacité à éviter les phénomènes d'interblocages en carrefour. La démarche suivie pour nos expérimentations consiste à placer des conducteurs simulés dans des situations de trafic propices aux interblocages.

Pour chaque situation, deux simulations sont effectuées : l'une basée sur l'utilisation de l'algorithme de coordination seul, l'autre basée sur l'utilisation de notre algorithme d'anticipation en amont de la coordination. Dans un premier temps, de manière à pouvoir identifier et valider le comportement anticipatif des agents, un certain nombre de véhicules est placé dès le début de la simulation dans une configuration particulière favorisant l'apparition d'un interblocage. Dans un second temps, des demandes de trafic sont générées aléatoirement pour différents débits et une moyenne du nombre d'interblocages est calculée. Cette seconde partie de l'évaluation nous permet de quantifier l'efficacité de l'algorithme d'anticipation.

5.1.2.1 Mise en évidence du comportement anticipatif

L'intersection considérée pour cette expérimentation est un carrefour non signalisé et soumis à la règle de la priorité à droite. Les véhicules sont placés dans la configuration initiale suivante :

- le véhicule 8 bloque le peloton composé des véhicules 0, 2, 3 et 4,
- le véhicule 2 bloque le peloton composé des véhicules 5, 6 et 7,
- le véhicule 6 bloque les véhicules 11 et 12,
- tous les véhicules ont une vitesse initiale nulle à l'exception des véhicules 8 et 13,
- tous les véhicules ont pour objectif de continuer dans leur direction.

La première image de la figure 5.2 illustre cette configuration.

Dans cette situation, seuls les mobiles 8 et 13 peuvent encore avancer, tous les autres sont bloqués. L'interaction se situe donc entre ces deux véhicules. Détaillons leurs comportements dans le cas où seul l'algorithme de coordination est utilisé.

Au pas de temps, $t = 1$, les véhicules 13 et 8 sont en conflit et doivent donc se coordonner. Compte tenu de sa vitesse nulle et de la distance le séparant du point de conflit, le véhicule 8 estime être dans la configuration suivante : $\neg prio(8, 13) \wedge prio(13, 8)$. De la même manière, le véhicule 13 estime être dans la configuration $prio(13, 8) \wedge \neg prio(8, 13)$. Sur la base de cette reconnaissance des priorités, l'algorithme de coordination conduit respectivement aux décisions *Go* et *Stop* pour les véhicules 13 et 8.

Le véhicule 8 commence donc à décélérer de manière à pouvoir s'arrêter avant le point de conflit et laisser passer le véhicule 13. Bien que sa décision soit *Go*, l'accélération du véhicule 13 est également négative puisqu'il est contraint par le véhicule 12 situé devant lui sur sa voie. Le véhicule 13 décélère donc (2ème image de la figure 5.2, pas de temps $t = 28$) de manière à venir s'arrêter derrière le véhicule 12 (4ème image de la figure 5.2, pas de temps $t = 270$).

En avançant, le véhicule 8 libère un espace pour les véhicules 1 et 9. Ceux-ci suivent le même raisonnement que les véhicules 8 et 13 et se coordonnent de la manière suivante : le véhicule 1 reste à l'arrêt et le véhicule 9 avance (3ème image de la figure 5.2).

La dernière image de la figure 5.2 montre le résultat de l'enchaînement des comportements précédemment décrits. Les véhicules sont mutuellement bloqués et ne peuvent plus avancer.

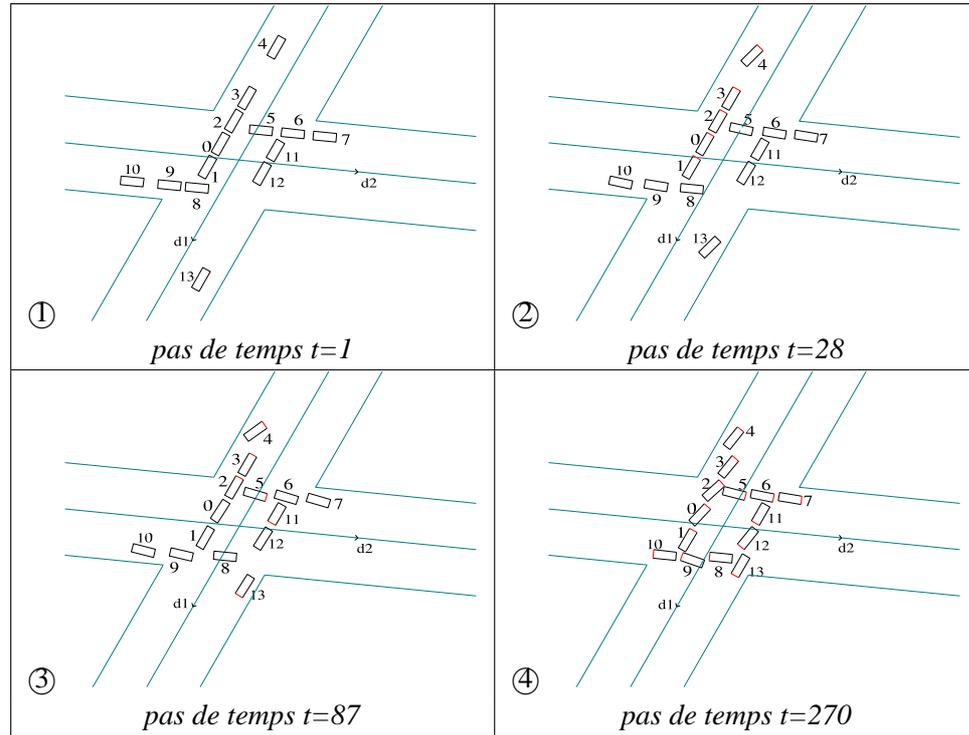


FIGURE 5.2 – Impressions écrans d’une simulation sans anticipation sur un scénario d’interblocages

Détaillons maintenant les différentes actions des agents dans le cadre du comportement anticipatif. La situation initiale est identique : 8 et 13 doivent se coordonner pour résoudre leur conflit (1ère image de la figure 5.3).

Le mobile 13 évalue de la même manière sa relation de priorité avec le véhicule 8 : $prio(13, 8) \wedge \neg prio(8, 13)$. Il continue son raisonnement en se construisant une représentation mentale de la situation sur la base des relations de blocage suivantes : $bpha(13, 12)$, $bph(12, 11)$, $bph(11, 6)$, ..., $bph(0, 1)$, $bph(1, 8)$. Ensuite, le mobile 13 prend en compte les véhicules non engagés dans le carrefour. Il ajoute ainsi pour le véhicule 10 les relations : $bpr(10, 0)$, $bpr(10, 2)$, etc.

Le mobile 13 termine la construction de sa représentation mentale en associant un domaine à chaque variable. Avant l’appel à l’algorithme d’anticipation, tous les domaines sont initialisés à $[1, +\infty[$. Après le premier appel à la fonction de propagation, le mobile 13 a une approximation des intervalles de temps de blocage de chaque véhicule, y compris de lui-même : $dom(13) = [1, 4] \cup [20, +\infty[$.

L’étape suivante de l’algorithme consiste à déterminer les effets de l’action Go . A partir de sa position, de sa vitesse et de son accélération courante, le mobile 13 détermine sa position future. De cette nouvelle position, il calcule une nouvelle vision de la situation et obtient ainsi les nouvelles relations topologiques induites par l’action Go . Ces relations sont ensuite interprétées en relation de blocage et de priorité.

Les nouvelles relations ainsi obtenues sont ajoutées à la représentation et une nouvelle propagation est réalisée. Le mobile 13 recherche alors les états non désirés et constate que son domaine a été réduit à l’ensemble vide ($dom(13) = \emptyset$), ce qui constitue un blocage infini. Par conséquent, il supprime Go

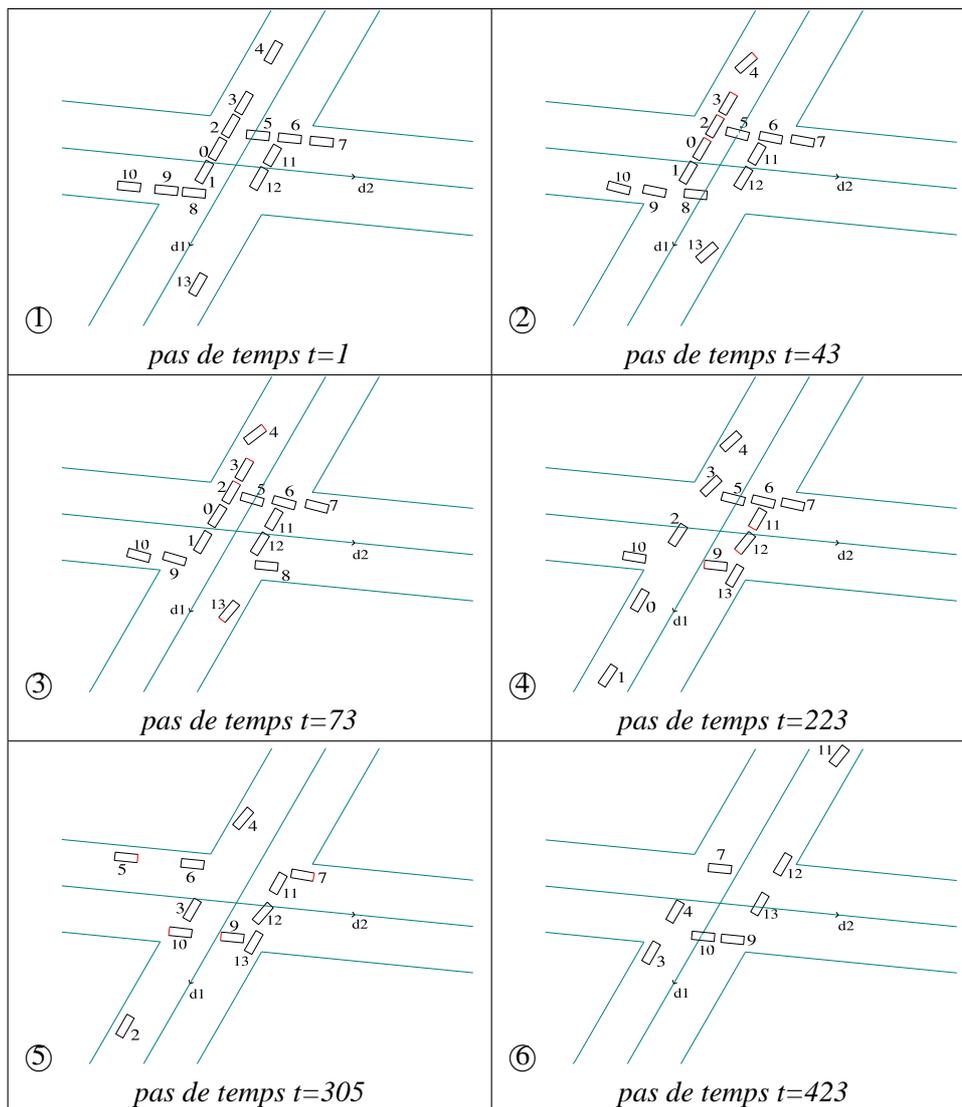


FIGURE 5.3 – Impressions écrans d’une simulation avec anticipation sur un scénario d’interblocage

de sa liste d’actions. Finalement, le mobile 13 n’a plus que *Stop* comme action possible. Il calcule donc sa décélération de manière à pouvoir s’arrêter avant le point de conflit et place sa variable d’état *actif* à faux.

Au pas suivant, le véhicule 8 détecte l’état inactif du véhicule 13 et ne le considère plus dans sa phase de coordination. Le véhicule 8 est donc libre d’avancer (images 2 et 3 de la figure 5.3). Une fois le véhicule 8 sorti du carrefour, le risque d’interblocage est levé et le véhicule 13 peut redémarrer pour venir derrière le véhicule 12. Pendant ce temps, la file de véhicules venant du nord se vide libérant ainsi la file de véhicules venant de l’est (images 4 et 5 de la figure 5.3). A la dernière image, les derniers véhicules réaccélèrent pour sortir du carrefour.

Les graphiques des figures 5.4 et 5.5 illustrent la dynamique des deux simulations décrites en présentant les courbes de vitesse et d’accélération du véhicule 13 dans le cas de l’anticipation et dans celui de la coordination seule.

Entre les pas 0 et 25, la variation de vitesse du véhicule 13 est identique pour les deux simulations : le mobile 13 réduit sa vitesse pour ne pas percuter le véhicule 12 à l'arrêt. Entre les pas 25 et 50, l'algorithme de coordination indique au mobile 13 qu'il peut avancer : il continue donc de décélérer mais plus faiblement. Sur le même intervalle de temps, l'algorithme d'anticipation contraint le véhicule 13 à éliminer *Go* de sa liste d'actions possibles. Nous pouvons donc constater sur la courbe de la figure 5.5 une décélération plus longue qui se traduit sur la courbe de la figure 5.4 par un arrêt du véhicule aux environs du pas 50.

Dans le même temps, le véhicule 8 accélère et le processus de déblocage du carrefour s'amorce. Cela a pour conséquence le désengorgement progressif du carrefour : à partir du pas 320, le mobile 13 peut ainsi réaccélérer jusqu'à sa vitesse désirée. Dans le cas de la simulation sans anticipation, le véhicule 13 est complètement bloqué à partir du pas 300 et sa courbe de vitesse tend vers zéro.

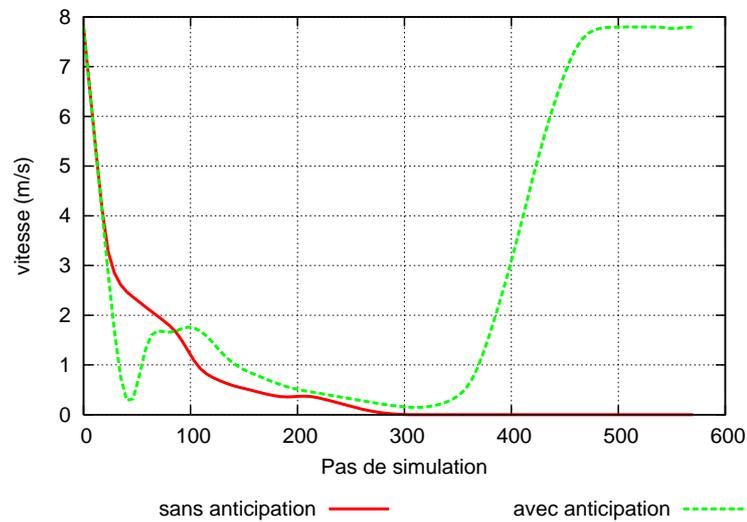


FIGURE 5.4 – Courbe de vitesse du mobile 13

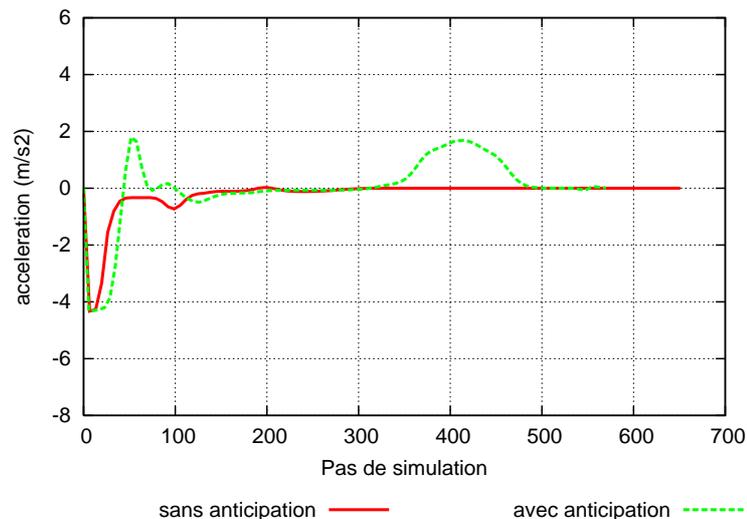


FIGURE 5.5 – Courbe de l'accélération du mobile 13

Ayant mis en évidence et illustré au travers de cet exemple le comportement anticipatif des agents en carrefour, nous essayons dans la suite de quantifier la réduction du nombre d'interblocages.

5.1.2.2 Réduction du nombre d'interblocages

Pour cette seconde partie de l'expérimentation, nous cherchons à évaluer la réduction du nombre d'interblocages. Nous proposons donc de mesurer le nombre d'interblocages obtenus en simulation avec et sans anticipation pour différents débits simulés [Doniec *et al.*, 2006b].

Le carrefour utilisé est le même que précédemment. Les véhicules sont créés 50 mètres en amont de l'intersection suivant une demande de trafic préalablement générée. Chaque demande de trafic est différente et caractérisée par un débit de véhicules. La durée de chaque simulation est d'une heure environ. Chaque interblocage se produisant à l'intérieur du carrefour est comptabilisé pour chaque demande de trafic. Une moyenne est ensuite faite entre les différentes demandes de trafic générées pour un même débit.

La méthodologie observée pour le comptage est la suivante : un processus de supervision, intégré au moteur de la simulation et totalement indépendant de la prise de décision de chaque agent, vérifie à chaque pas de temps la présence d'un interblocage. Lorsqu'une telle situation est avérée, la simulation est suspendue, le carrefour est vidé (destruction des véhicules bloqués) puis la simulation reprend.

Les résultats obtenus sont illustrés par la courbe de la figure 5.6. Celle-ci présente la variation du nombre d'interblocages en fonction du débit simulé. Nous pouvons ainsi constater que pour une simulation d'une heure environ, le nombre d'interblocages comptabilisé varie de 0 à 35 en fonction du débit pour une simulation sans anticipation. Ainsi pour des débits relativement faibles, les interblocages sont quasiment inexistantes. En revanche, à partir de 300 véhicules par heure, le nombre d'interblocages croît de manière quasi linéaire.

Dans le cas d'une simulation basée sur des agents anticipatifs, nous pouvons remarquer que le nombre d'interblocages reste quasiment constant et proche de zéro pour des débits simulés allant jusque 800 véhicules/heure par branche. Ces résultats montrent tout l'intérêt de notre approche et prouvent que les interblocages peuvent être évités sans avoir recours à un agent superviseur comme dans [Paruchuri *et al.*, 2002].

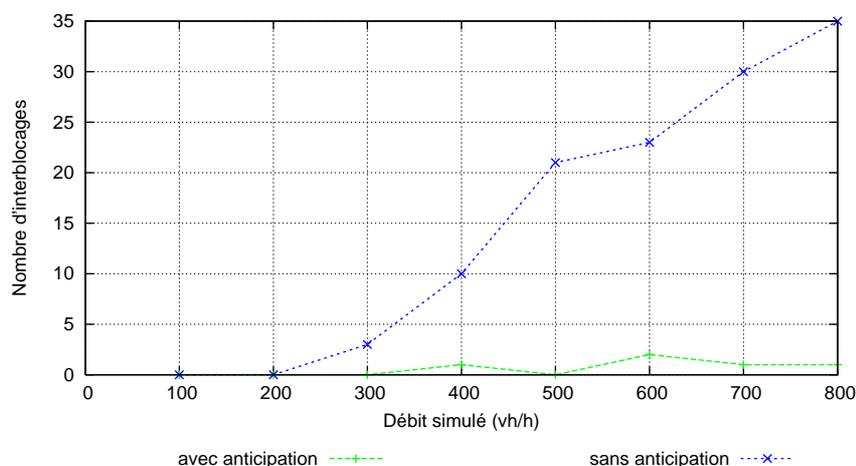


FIGURE 5.6 – Variation du nombre d'interblocages en fonction du débit simulé

Le réalisme de ces résultats peut toutefois être discuté. En effet, dans la réalité il n'est pas rare d'assister à de tels phénomènes d'interblocages. Le désengorgement du carrefour est alors obtenu soit par l'intervention d'un agent des forces de l'ordre, soit par une prise d'initiatives de certains conducteurs qui opèrent alors une série de manœuvres (marche arrière, crénaux, etc) pour débloquer une branche du carrefour. Pour les raisons déjà évoquées au chapitre 2, la gestion de la marche arrière n'est pas traitée. En l'absence d'une telle gestion, il convient donc d'anticiper et d'éviter le maximum de situations d'interblocages, quitte à s'écarter un peu de la réalité.

5.1.3 Succession de carrefours

Nous nous intéressons maintenant au cas de la succession de carrefours rapprochés. Une telle configuration génère souvent des phénomènes de remontées de queue qui peuvent également engendrer des interblocages. Une remontée de queue peut avoir différentes origines :

- la présence d'un contrôleur de feu dont le réglage des phases est mal adapté aux débits (cas vu dans le chapitre 4),
- la succession de mouvements tournants vers la gauche,
- une perturbation comme par exemple un accident ou un stationnement gênant.

Pour tester l'effet de l'anticipation sur ce genre de situation, nous avons considéré le scénario suivant. Deux carrefours soumis à la règle de la priorité à droite sont proches (séparés d'une trentaine de mètres). Les deux carrefours sont formés de l'intersection d'un axe transversal $d1$ coupant deux rues parallèles $d3$ et $d2$. Des véhicules sont générés aléatoirement aux extrémités de $d1$. Les pourcentages de mouvements tournants aux deux carrefours sont fixés de manière à privilégier les tournes-à-gauche : 80% de véhicules tournant à gauche et 20% de véhicules allant tout droit. L'attente d'un véhicule voulant traverser le flux de trafic opposé pour tourner à gauche peut alors engendrer derrière lui une queue de véhicules remontant jusqu'au carrefour situé en amont.

La figure 5.7 présente sous la forme de quatre images la simulation de ce scénario avec des agents dépourvus de capacité anticipative. La première image de la figure 5.7 présente le début de la simulation. Suite à l'attente respective des véhicules 16 et 14 pour tourner à gauche, deux remontées commencent à apparaître : la première sur l'axe $d1$ dans le sens est-ouest et la seconde en sens inverse. Cette double remontée de queue se précise à la deuxième image. Deux interactions vont être déterminantes pour l'évolution de la situation : l'interaction du véhicule 17 avec le véhicule 37 et l'interaction entre les véhicules 34 et 22.

Les véhicules 37 et 34 estiment leurs différentes relations de priorités et se considérant comme prioritaires décident d'avancer. Les deux conflits sont alors résolus : le véhicule 37 passe avant le véhicule 17 de même que le véhicule 34 passe avant le véhicule 22. Mais ces deux décisions engendrent une situation d'interblocage entre les deux carrefours : le flux de véhicules provenant de l'est bloque le flux provenant de l'ouest et inversement (3ème image de la figure 5.7). Le blocage total des deux carrefours se confirme à la dernière image : le véhicule 40 s'est également engagé dans le carrefour ne laissant ainsi plus aucune marche de manœuvre pour débloquer la situation.

Le même scénario a été simulé avec des agents appliquant le comportement anticipatif. La figure 5.8 présente le déroulement de la simulation. Les deux premières images sont identiques à celle de la première simulation : la double remontée de queue s'installe entre les deux carrefours. Mais, l'interaction entre les véhicules 34 et 22 va se passer différemment. En construisant et en analysant sa représentation mentale de la situation, le véhicule 34 est capable d'anticiper qu'il va se bloquer lui-même s'il décide d'avancer. Il opte donc pour l'action *Stop*, laissant ainsi passer le véhicule 22

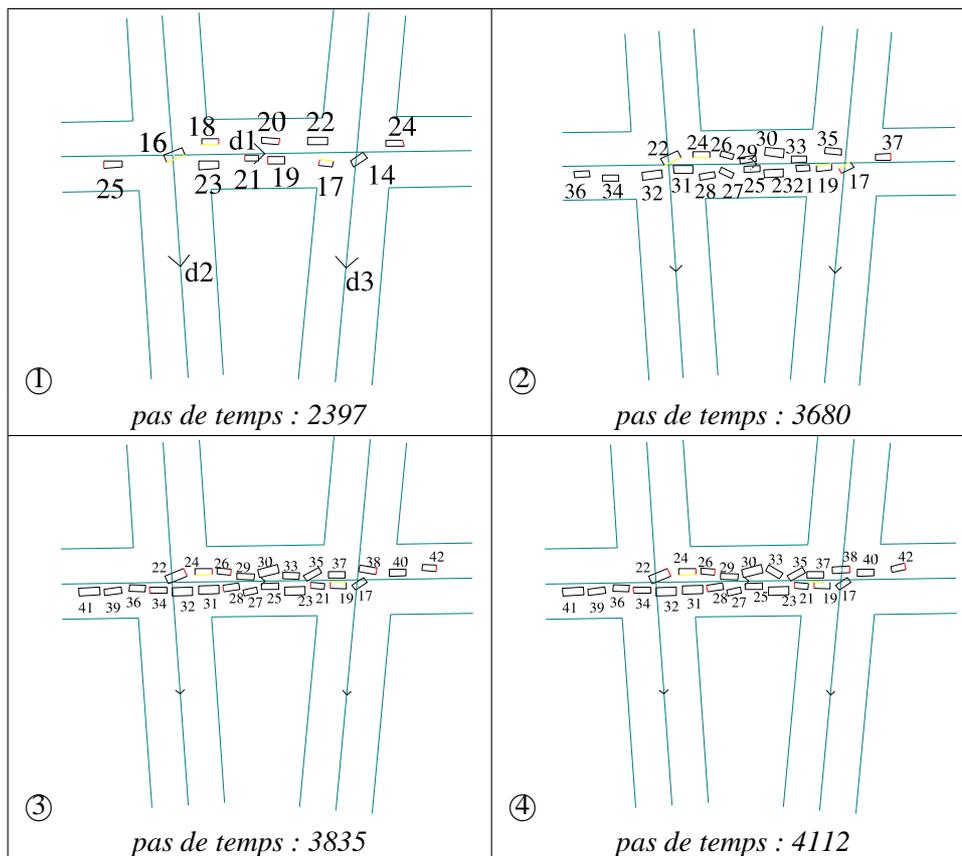


FIGURE 5.7 – Impressions écrans d’une simulation sans anticipation sur un scénario impliquant deux carrefours successifs

(images 3 et 4). Le même raisonnement se répète permettant aux véhicules 24 et 17 de sortir du carrefour (image 5).

Pour pouvoir traiter ce scénario, nous avons dû paramétrer notre algorithme de manière à étendre la zone d’anticipation décrite au chapitre 4. Nous avons étendu la sélection des agents (pour la construction de la représentation) au-delà de la zone d’anticipation décrite au chapitre 4. Ainsi, seuls les agents évoluant dans l’un des deux carrefours anticipent mais ils considèrent dans leurs représentations tous les agents présents en amont et en aval des deux intersections. Ce scénario est particulièrement intéressant car il aborde la question de la visibilité, de la prise d’informations, de l’anticipation en carrefour.

Dans la réalité, un conducteur réel n’aurait sans doute pas été capable d’anticiper l’interblocage complet des deux carrefours. Le manque de visibilité par exemple ne lui aurait pas permis de prendre connaissance de la situation au carrefour suivant. Il convient donc de trouver un compromis acceptable entre les capacités d’anticipation d’un conducteur réel et d’un conducteur simulé. Pour des carrefours relativement proches (moins de 50m), il nous est apparu intéressant que les agents puissent anticiper sur les deux intersections. Au-delà, les risques d’interblocages sont moindres et l’anticipation peut être cantonnée à un seul carrefour.

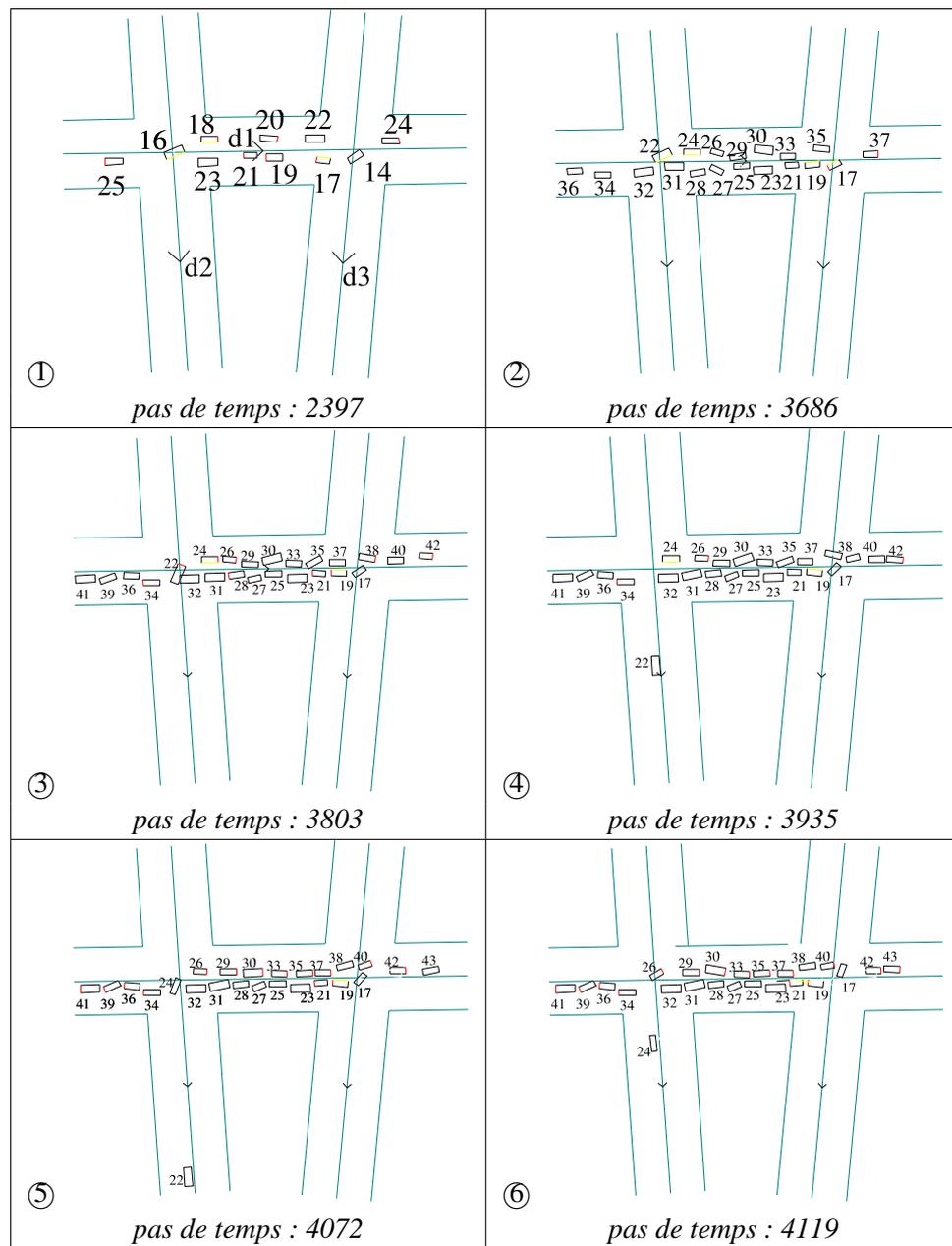


FIGURE 5.8 – Impressions écrans d’une simulation avec anticipation sur un scénario impliquant deux carrefours successifs

Dans cette première section, nous avons présenté les évaluations des comportements locaux des agents. Nous nous intéressons dans la deuxième partie de ce chapitre à la validation du trafic résultant de ces comportements locaux.

5.2 Simulation d'un carrefour réel et validation

Dans le domaine de la simulation de trafic pour les carrefours, il est relativement difficile de disposer de données réelles relevées sur le terrain. Les gestionnaires des infrastructures routières sont souvent amenés à poser sur le bord des routes des capteurs permettant de mesurer différentes variables de trafic comme le débit ou la vitesse moyenne. Lorsque ces données sont mises à disposition, il est souvent difficile de les exploiter en simulation comportementale car lors de mesures réalisées sur un tronçon, les informations telles que l'origine et la destination des véhicules sont rarement relevées.

Dans le cas d'un carrefour, ces informations sont extrêmement importantes car elles permettent de connaître la répartition des véhicules entre les différents mouvements tournants. Le relevé de données en carrefour (en vue d'une utilisation en simulation) nécessite donc d'effectuer un comptage manuel des véhicules (éventuellement sur la base d'enregistrements vidéo), ce qui représente un travail considérable. Cela explique le manque de données de trafic en carrefour disponibles et accessibles aux chercheurs du domaine.

Les données que nous avons utilisées pour notre validation ont été mesurées par l'Université Reggio de Calabre (Italie) et nous ont été mises à disposition dans le cadre d'une collaboration avec l'INRETS.

5.2.1 Démarche

L'intersection utilisée pour notre validation est située dans la ville de Reggio au sud de l'Italie. Celle-ci marque la jonction entre un axe principal orienté Nord / Sud direction Zerbi, et un axe secondaire orienté Est / Ouest direction Roma. Ce dernier est signalisé par deux panneaux stop. Les branches Est et Ouest du carrefour sont à deux voies, les branches Nord et Sud comportent une voie de circulation et une voie de stationnement. L'espace au centre est suffisamment large pour permettre le stockage de véhicules (figures 5.9 et 5.10).

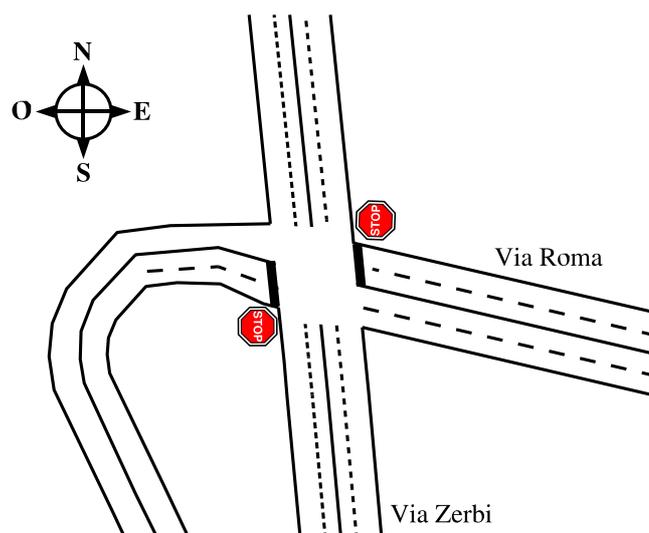


FIGURE 5.9 – Plan de l'infrastructure du carrefour réel (Italie, Reggio de Calabre)



Nord vers Sud



Est vers Ouest



Sud vers Nord



Ouest vers Est

FIGURE 5.10 – Photos du carrefour "Roma-Zerbi"

Les données trafic mesurées par l'université de Reggio de Calabre à l'intersection décrivent une période d'une heure entre 12h30 et 13h30, un jour normal de semaine. Pour chaque branche du carrefour, nous disposons du débit entrant exprimé en nombre de véhicules par heure ainsi que la répartition exprimée en pourcentage des différents mouvements tournants à l'intérieur du carrefour (tourne-à-droite, tourne-à-gauche, tout-droit).

A partir de ces données, nous avons généré une demande de trafic équivalente à celle mesurée sur le terrain et respectant les différents pourcentages de mouvements tournants. Nous avons ensuite placé dans la simulation des capteurs virtuels dont le fonctionnement est semblable aux capteurs à boucle magnétique utilisés dans la réalité. Ces capteurs virtuels ont été placés en entrée et en sortie du carrefour de manière à pouvoir comptabiliser les flux de véhicules entrant et sortant du carrefour. Les véhicules simulés sont créés environ 200 mètres en amont de ces capteurs avec une vitesse désirée correspondant à la vitesse moyenne relevée en condition réelle.

Les données enregistrées en cours de simulation par les capteurs virtuels fournissent un ensemble de fichiers texte regroupant des informations relatives à chaque véhicule ayant passé sur l'un des capteurs. Parmi ces informations figure en particulier le temps écoulé entre le début de la simulation et l'instant où le véhicule franchit le capteur.

L'utilisation de l'outil de dépouillement de données présenté au chapitre 1 permet d'agréger ces données de manière à obtenir les débits de véhicules pour chaque axe et sens de circulation sur une période de temps donnée. Dans le cadre de notre évaluation, les données enregistrées durant la simulation ont été agrégées par périodes de 5 min.

Ne disposant que des débits entrants, seules les données issues des capteurs situés à l'entrée du carrefour ont été utilisées dans les comparaisons qui suivent. Les débits simulés en sortie ont permis de s'assurer que les pourcentages directionnels étaient bien respectés.

La demande de trafic est simulée dans un premier temps avec l'algorithme de coordination et le comportement issus des travaux de [Champion, 2003]. La même demande est ensuite resimulée avec des agents faisant état du comportement opportuniste et anticipatif introduit dans ce mémoire.

5.2.2 Résultats de simulation

5.2.2.1 Courbes des débits

Pour chacun des quatre axes du carrefour, nous avons comparé les débits réels avec les débits obtenus en simulation. Chacune de ces comparaisons est illustrée par un graphique représentant les débits atteints par périodes de cinq minutes. Les tracés en trait plein correspondent aux débits réels mesurés sur le terrain. Les tracés en pointillé clair correspondent aux débits obtenus en simulation pour le comportement de [Champion, 2003] et les tracés en pointillé foncé aux débits obtenus pour le comportement opportuniste et anticipatif.

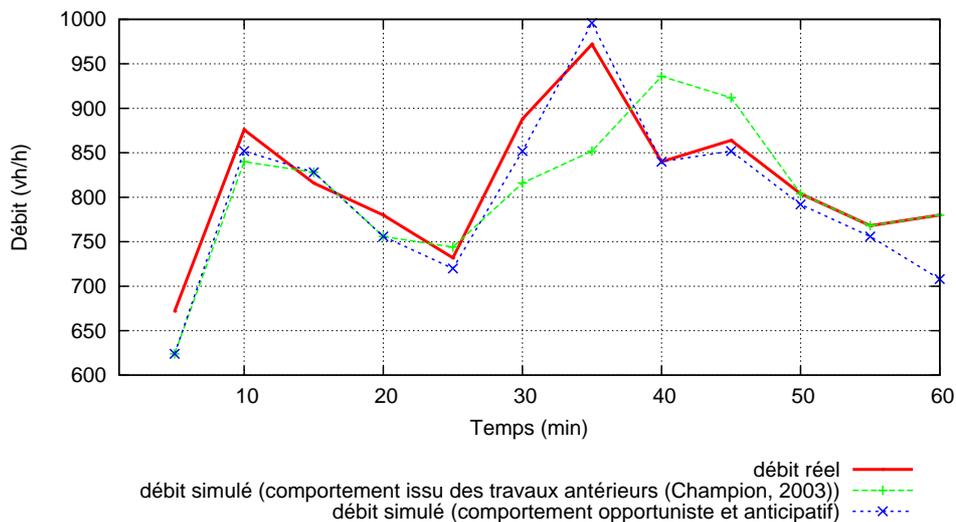


FIGURE 5.11 – Débits obtenus en simulation pour l'axe nord / sud

La figure 5.11 représente la courbe des débits pour l'axe principal dans le sens de circulation nord / sud. Nous observons que l'introduction du comportement opportuniste et anticipatif améliore la validité du débit notamment en milieu de simulation. En effet, aux environs de la 30^{ème} minute, les débits obtenus avec le comportement issus des travaux antérieurs sont en baisse par rapport au débit réel : 820 au lieu de 900. Cette baisse s'accroît 5 minutes plus tard et on peut observer un écart de 125 véhicules/heure avec le débit réel. La première simulation atteint son maximum en terme de débit

aux alentours de la 40ème minute alors que dans la réalité le pic est atteint 5 minutes plus tôt. Ce phénomène n'apparaît plus dans la deuxième simulation dont le débit simulé atteint son maximum à la 35ème minute comme pour le débit réel.

La figure 5.12 présente la courbe des débits pour l'axe principal dans le sens de circulation sud / nord. Le comportement existant offrait déjà de bons résultats sur cet axe avec une courbe de débit simulé proche de celle du débit réel. L'introduction de notre modèle de comportement conserve ces résultats.

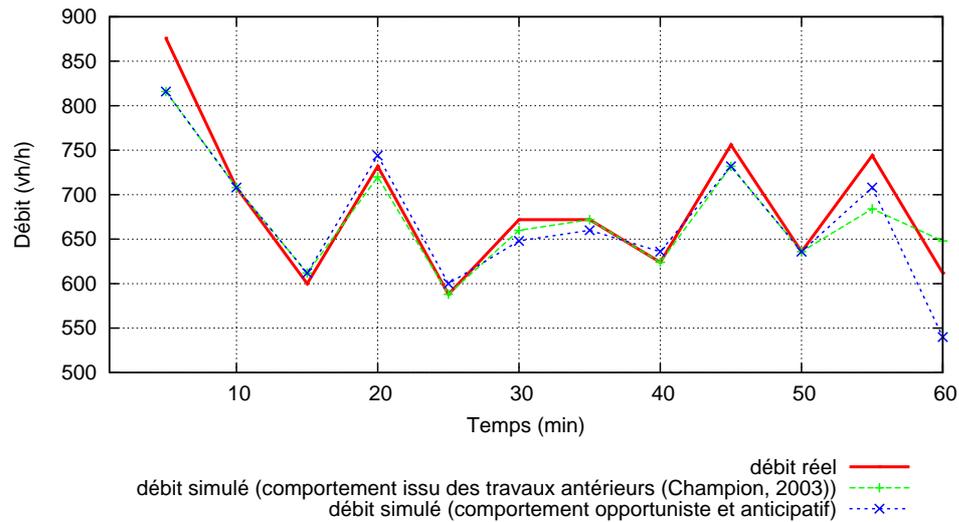


FIGURE 5.12 – Débits obtenus en simulation pour l'axe sud / nord

La figure 5.13 présente la courbe des débits pour l'axe secondaire dans le sens de circulation est / ouest. Cet axe est non prioritaire et est signalisé par la présence d'un stop. Dans [Champion, 2003], l'auteur indique qu'une des faiblesses de son modèle concerne l'insertion de véhicules non prioritaires dans un flux de trafic conflictuel. Nous constatons ainsi sur ce graphique que la première simulation réalisée avec le comportement issu des travaux antérieurs présente des écarts importants en terme de débit avec la réalité. A 10 minutes de simulation par exemple, un pic de débit a été mesuré à 600 véhicules/heure sur le terrain. En simulation, le débit observé sur la même période tombe à 420. La seconde simulation réalisée améliore grandement les débits obtenus. D'une manière générale, l'allure de la courbe des débits simulés est identique à celle des débits réels. L'écart à la 10ème minute est réduit et passe de 180 à 100 véhicules/heure.

Cette amélioration des résultats est due en grande partie aux comportements opportunistes introduits qui facilitent l'insertion de véhicules dans un flux de trafic prioritaire. Dans le cas ici présent, cela se traduit par des temps d'attente au stop plus courts et plus proches de ceux observés dans la réalité.

La figure 5.14 présente la courbe des débits pour l'axe secondaire dans le sens ouest / est. Que ce soit avec le comportement existant [Champion, 2003] ou le comportement opportuniste, les débits simulés sont très proches de ceux mesurés sur le terrain. Ces résultats ne nous semblent pourtant pas significatifs eu égard aux faibles débits observés sur cette branche du carrefour. Le débit maximum atteint est en effet de 60 véhicules/heure, ce qui est très peu au regard des débits sur les trois autres branches de l'intersection.

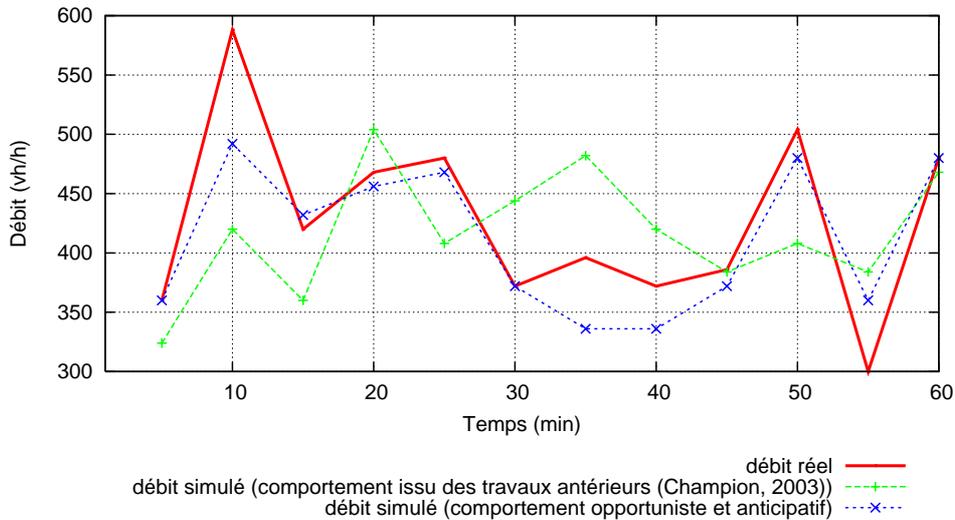


FIGURE 5.13 – Débits obtenus en simulation pour l’axe est / ouest

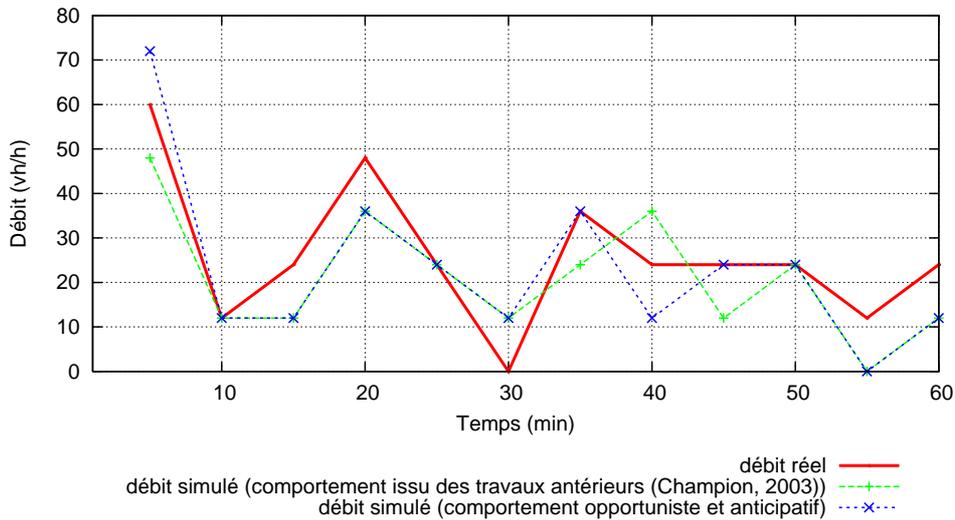


FIGURE 5.14 – Débits obtenus en simulation pour l’axe ouest / est

5.2.2.2 Indicateurs

En ingénierie du trafic, il est courant d'utiliser différents indicateurs permettant d'évaluer la qualité des résultats obtenus par une simulation de trafic. L'un des plus utilisés est l'EQMR. Il correspond à l'erreur moyenne quadratique entre le débit réel y_i et le débit simulé x_i et est exprimé par la formule suivante :

$$EQMR = \sqrt{\frac{\sum_i (x_i - y_i)^2}{\sum_i y_i^2}}$$

Cet indicateur a été calculé pour chaque branche du carrefour. Les résultats sont reportés dans le tableau 5.1 ainsi que la plage des débits simulés. Nous pouvons ainsi constater que les meilleurs résultats en terme d'écart avec les débits réels sont obtenus avec le comportement opportuniste et

	EQMR (comportement opportuniste et anticipatif)	EQMR (comportement issu des travaux antérieurs [Champion, 2003])	Min débit (vh/h)	Max débit (vh/h)
axe Nord-Sud	0,04	0,06	600	900
axe Sud-Nord	0,03	0,03	680	980
axe Est-Ouest	0,06	0,15	300	600
axe Ouest-Est	0,30	0,32	0	60

TABLE 5.1 – EQMR

anticipatif. L'amélioration par rapport au comportement issu des travaux antérieurs est donc tout à fait notable et particulièrement flagrante pour l'axe est-ouest sur lequel le taux d'erreur est réduit de plus de la moitié.

Les valeurs d'EQMR obtenues sont, pour trois des axes, inférieures à 0,1 ce qui correspond à une erreur inférieure à 10%. Dans la littérature, une bonne simulation de trafic est généralement présentée avec un taux d'erreur inférieur à 15%. Notons encore une fois que les résultats de la courbe de la figure 5.14 sont peu représentatifs puisque les débits simulés sont faibles et peu significatifs.

5.2.3 Reproductibilité de la simulation

Au delà des bons résultats obtenus en terme de trafic, il nous a semblé pertinent d'analyser la reproductibilité des phénomènes de trafic en fonction du pas de temps utilisé. ARCHISIM offre en effet la possibilité de paramétrer la durée d'un pas de simulation. La simulation peut s'effectuer à pas fixe avec une durée de pas paramétrable entre 1 et 30 millisecondes. Le fonctionnement d'ARCHISIM peut également être en temps réel lors de l'utilisation du simulateur de conduite : 1 seconde de calcul équivaut alors à 1 seconde de simulation.

Nous nous sommes donc intéressés à la stabilité des résultats précédemment présentés en fonction du pas de temps utilisé. La validation précédente a donc été rejouée pour différentes durées de pas de temps : temps réel (pas de temps variable), pas de temps fixe égal à 1 ms, 5 ms, 10 ms et 20 ms. Pour chaque branche du carrefour, nous avons pu évaluer la variation d'EQMR observée entre les différentes simulations. Ces résultats sont exposés au travers des graphiques de la figure 5.15.

L'EQMR varie en moyenne de 0,009. La plus forte variation est constatée sur l'axe Est-Ouest avec des écarts allant jusqu'à 0,02. Ce résultat tend donc à montrer que l'insertion de véhicules provenant d'un flux non prioritaire est un phénomène relativement sensible.

Cette analyse exhibe une certaine stabilité du modèle quelque soit la durée du pas de temps utilisé. Ce critère d'évaluation est rarement appliqué dans la littérature. Il nous semble pourtant important : il permet en effet de s'assurer d'une certaine reproductibilité des situations de trafic simulées quelque soit le paramétrage du pas de simulation.

5.3 Perspectives

Dans cette dernière partie, nous discutons des différentes extensions possibles du modèle dans le cadre d'ARCHISIM et détaillons quelques perspectives de recherche plus générales.

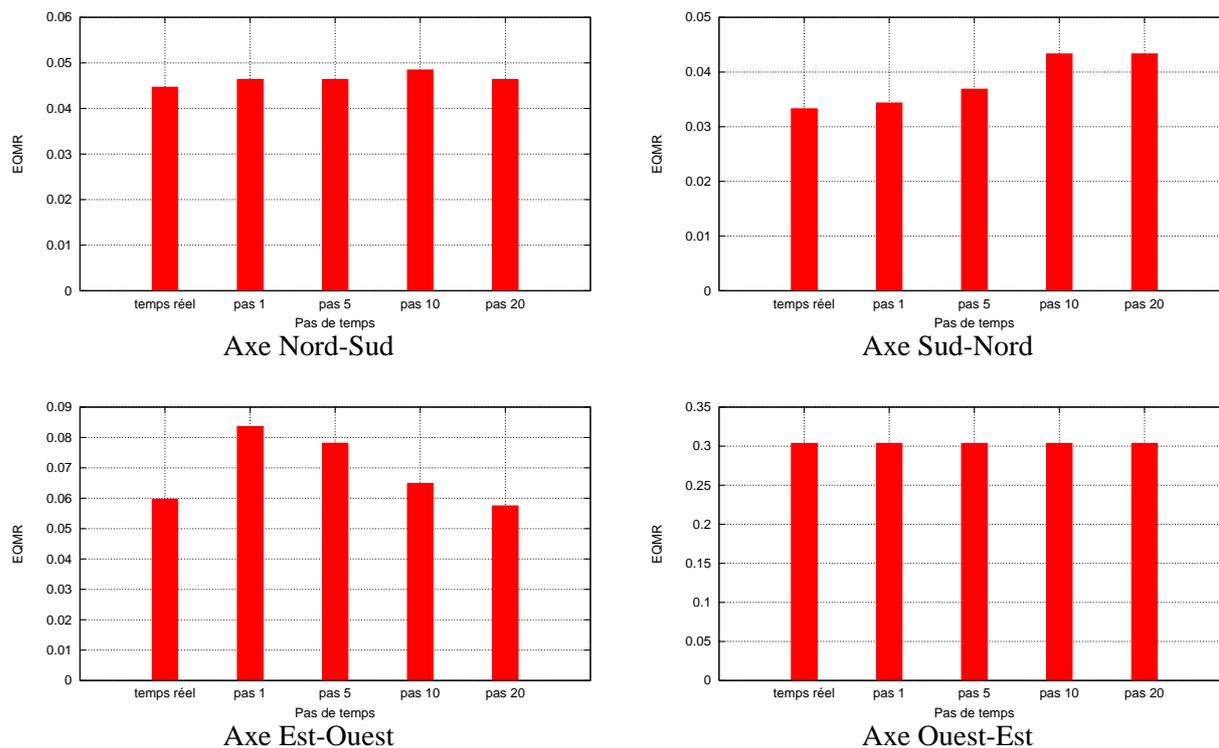


FIGURE 5.15 – Variation de l'EQMR en fonction de la durée du pas de temps

5.3.1 Perspectives de recherche dans le cadre d'ARCHISIM

5.3.1.1 Extensions du modèle pour la gestion de l'accélération latérale

Les travaux présentés dans ce mémoire ont été appliqués à la gestion de l'accélération longitudinale des véhicules en carrefour. Une extension du modèle concerne la gestion de l'accélération latérale. Il s'agit de permettre aux conducteurs en situation de carrefour de changer de voie ou d'effectuer des écarts par rapport au centre de leur voie.

Cette extension est essentielle pour une meilleure prise en compte de l'espace roulant au centre des carrefours. Elle permettra aux véhicules de mieux se positionner sur la largeur de la route de manière à gérer plus facilement les interactions lors des phases d'insertion et de traversée de flux.

La gestion de l'accélération latérale nécessite de répondre à plusieurs problématiques. Un conducteur simulé doit en effet être capable :

- de se créer un marquage virtuel ou de se réapproprié un marquage existant au centre des carrefours [Bonte *et al.*, 2006],
- d'évaluer quelle est la voie la plus appropriée en fonction :
 - du contexte du conducteur simulé dans le trafic : position courante sur la chaussée, occupation des différentes voies
 - des objectifs propres du conducteur simulé : direction au prochain carrefour, vitesse désirée
 - des caractéristiques propres au conducteur : longueur de son véhicule et capacité à réaliser la manœuvre

Au niveau de la coordination, cela revient à considérer des actions supplémentaires représentant le déport du véhicule sur la droite et sur la gauche. Un agent aura par exemple le choix entre quatre actions possibles : *Go*, *Stop*, *ADroite*, *AGauche*. Ces deux dernières actions pourront éventuellement être vues comme deux sous-actions de *Go* puisqu'une voiture n'est pas capable d'avoir une accélération latérale sans une accélération longitudinale.

Cette extension ne pose *a priori* pas de problème en ce qui concerne le modèle d'anticipation puisque notre algorithme considère la liste d'actions d'un agent comme un paramètre d'entrée. Les actions *ADroite* et *AGauche* pourront donc être anticipées de la même manière que l'action *Go*.

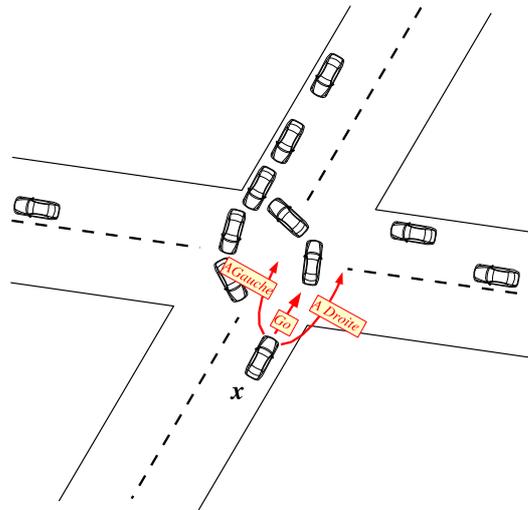


FIGURE 5.16 – Extension possible du modèle d'anticipation pour la gestion de l'accélération latérale

Cette extension permettrait d'offrir une plus grande richesse et variété de comportement. Considérons l'exemple de la figure 5.16. Si l'objectif de x est par exemple de tourner à gauche, trois alternatives demeureront pour x après l'élimination de l'action *Go* par l'algorithme d'anticipation. Soit x pourra s'arrêter (comme cela est actuellement le cas), soit il pourra aller se stocker dans l'espace libre au centre du carrefour (action *AGauche*), soit il pourra se déporter sur la droite et se stocker au même niveau que le véhicule qui le précède.

Puisque notre algorithme repose sur des calculs d'intervalles de temps, il est envisageable d'exploiter ces intervalles pour évaluer la pertinence d'un changement de voie. La voie la plus adéquate pourrait être par exemple celle assurant un temps d'arrêt minimum au centre du carrefour.

5.3.1.2 Validation

Du point de vue de la validation des phénomènes de trafic en carrefour, il serait intéressant de considérer d'autres variables que le débit. La vitesse moyenne pratiquée en amont et au centre du carrefour ainsi que le temps d'attente moyen au stop sont autant d'éléments qu'il serait pertinent de prendre en compte pour une évaluation encore plus précise du trafic.

Compte tenu des possibilités offertes par ARCHISIM, un autre type de validation serait également envisageable. Il s'agirait de confronter des enregistrements vidéo de carrefours réels avec une

simulation en temps réel. Cet exercice serait très enrichissant car il permettrait de comparer les comportements locaux des agents aux comportements effectivement rencontrés dans la réalité.

Une telle approche est bien entendu lourde à mettre en place, puisqu'avant de pouvoir simuler la situation filmée, il faut procéder à un dépouillement manuel de la vidéo de manière à repérer : les temps intervéhiculaires, les origines et destinations de chaque véhicule, les temps d'arrêt, les vitesses pratiquées, etc. Par ailleurs, cet exercice n'a de sens que s'il permet également d'améliorer le modèle comportemental en cas d'écart entre le comportement réel et le comportement simulé. Pour cela, il faut être capable d'analyser et d'identifier à travers la vidéo les différents motifs qui ont conduit à la prise de décision du conducteur filmé.

5.3.2 Perspectives générales

5.3.2.1 Généralisation à d'autres types de mobiles

Les résultats expérimentaux exposés dans ce chapitre confirment l'intérêt et la pertinence d'une approche alliant les comportements opportunistes avec les capacités d'anticipation. Notre algorithme d'anticipation basé sur les réseaux de contraintes est générique puisqu'il permet d'anticiper les actions qui lui sont fournies en paramètres d'entrée. Cette approche peut donc être généralisée à d'autres types de mobiles sous condition d'être capable de décrire les différentes actions primitives du mobile et de pouvoir appliquer l'algorithme avec des temps de calcul raisonnables. Rappelons que la complexité de notre approche dépend du nombre d'agents perçus, du nombre de relations entre ces agents et du nombre d'actions à anticiper.

En particulier, nous envisageons donc d'appliquer notre approche à d'autres types de mobiles. En particulier, notre travail peut s'avérer pertinent pour les AGV (Automated Guided Vehicle). Un cas d'application serait par exemple le déplacement de transpalettes automatisés dans un entrepôt. Une telle application requiert en effet des capacités de déplacement et d'anticipation dans un contexte d'environnement fortement dynamique où les collisions et interblocages doivent être évités [Helleboogh *et al.*, 2005].

5.3.2.2 Prise en compte des relations n-aires

Le modèle d'anticipation que nous avons proposé repose sur la reconnaissance et l'utilisation de relations binaires entre les agents. Ces relations sont considérées comme des contraintes et permettent de construire pour chaque agent une représentation mentale dont la structure s'apparente à un réseau de contraintes.

Pour certaines applications, il peut être intéressant de gérer au niveau de l'anticipation des relations n-aires et non plus binaires. Rappelons que les réseaux de contraintes sont un cas particulier des problèmes de satisfaction de contraintes (CSP) où les contraintes portent sur au maximum deux variables. Cette propriété portant sur l'arité des contraintes permet d'établir la correspondance avec les graphes²³ où les variables sont représentées par des sommets et les contraintes par des arêtes.

Dans la littérature sur les CSP, deux approches sont utilisées pour le traitement des contraintes n-aires. Les contraintes n-aires peuvent être encapsulées à l'intérieur de variables qui viennent s'ajouter

23. d'où le nom de réseaux de contraintes donné aux CSP binaires

au CSP initial. Le CSP initial se trouve ainsi modifié. Transposé à notre approche de l'anticipation, cela signifie qu'un agent devant anticiper devra se représenter un agent fictif servant à encapsuler la relation n-aire à modéliser. Plusieurs questions se posent alors : quelle interprétation peut-on donner à un tel agent fictif ? Comment conserver la sémantique originale de la relation n-aires lors de sa décomposition en relations binaires ?

La seconde approche consiste à utiliser des algorithmes généralisés aux CSP non-binaires. Ainsi l'algorithme AC-3 que nous avons adapté pour notre application possède une version généralisée GAC-3 [Mackworth, 1977b]. Une telle approche apparaît plus naturelle puisqu'elle conserve les sémantiques originales des relations.

5.3.2.3 Vers un modèle d'anticipation collective

Notre proposition repose sur l'utilisation de réseaux de contraintes comme représentation mentale pour les agents. Nous faisons par ailleurs l'hypothèse que tous les agents du système anticipent de manière individuelle sur la base d'une représentation mentale de l'environnement qui leur est propre. Or, dans certains cas, un ensemble d'agents peut avoir comme objectif l'anticipation d'états non désirés communs (cas de la formation de coalitions [Aknine *et al.*, 2000] par exemple). L'anticipation prend alors une dimension collective : les agents raisonnent ensemble à la détection d'états non désirés. Ayant des objectifs communs (et donc potentiellement des états non désirés identiques), il est naturel de se demander si les agents ne pourraient pas partager une seule et même représentation mentale. Dans cette optique, il pourrait être intéressant de considérer une extension des réseaux de contraintes : les problèmes de satisfaction de contraintes distribués (disCSP).

Les CSP distribués ont été introduits pour modéliser ces problèmes naturellement distribués : le but recherché n'est pas la parallélisation [Piechowiak et Hamadi, 2002](ie la distribution d'un problème centralisé sur plusieurs unités de calcul en vue de le résoudre en un temps plus court). Le traitement des CSP distribués s'inscrit plus dans le cadre d'une coopération entre sous-problèmes permettant la convergence vers une solution globale.

Définition 5.1 *Un disCSP est un quadruplet (X, D, C, A) où :*

- X est un ensemble fini de n variables : $\{x_1, x_2, \dots, x_n\}$
- D l'ensemble des domaines associés à ces variables :
 $D = \{\text{dom}(x_1), \text{dom}(x_2), \dots, \text{dom}(x_n)\}$
- C est un ensemble fini de m contraintes : $\{c_1, c_2, \dots, c_m\}$
- A est un ensemble fini d'agents : $\{A_1, A_2, \dots, A_p\}$ où chaque agent possède un sous-ensemble des variables du problème de telle manière que A soit une partition de X .

Un CSP distribué peut donc être vu comme un ensemble de sous-réseaux distribués parmi un ensemble d'agents. Les algorithmes proposés dans la littérature pour la simplification ou la résolution de disCSP sont distribués de manière à éviter une centralisation des données coûteuse en temps de calcul et en espace mémoire ([Yokoo, 2001],[Doniec *et al.*, 2005c]). Une grande majorité des algorithmes repose sur l'échange de messages entre les agents.

Il nous apparaît intéressant d'appliquer ce formalisme à notre modèle d'anticipation. Cela permettrait aux agents d'anticiper grâce à une seule représentation commune à tous. Chaque agent aurait en charge une partie de cette représentation. La répartition des ensembles $|A|$ et $|R|$ parmi les agents du système resterait à définir en fonction de l'application. Pour que l'utilisation du formalisme soit efficace, cette répartition devra :

- limiter la redondance des informations et la duplication des données, en particulier pour des agents impliqués dans différentes relations
- limiter les communications entre agents lors des phases de simplification des réseaux

Plusieurs algorithmes de propagation de contraintes ont été proposés dans la littérature [Hamadi, 1999] et semblent être applicables à notre modèle. La seule condition est de pouvoir assurer des communications par messages entre les agents. Dans le cas d'ARCHISIM par exemple, cette piste de recherche nécessiterait de modifier une partie de l'architecture de simulation de manière à autoriser des communications directes entre les agents.

Conclusion

Dans ce chapitre, nous avons présenté les différentes expérimentations menées pour l'évaluation et la validation de notre approche proposée au chapitre 3 et appliquée à la simulation de trafic au chapitre 4.

La première partie de notre validation s'est intéressée à une évaluation visuelle des comportements individuels obtenus au sein de la simulation. Divers scénarii ont été testés de manière à évaluer les capacités de notre algorithme d'anticipation à réduire les phénomènes d'interblocages dans diverses situations de trafic.

La seconde partie de la validation a consisté à simuler un carrefour existant. Ce travail est le fruit d'une collaboration avec l'Université de Reggio de Calabre qui nous a fourni des données statistiques relatives aux débits et mouvements tournants mesurés sur un carrefour réel situé dans le sud de l'Italie.

La comparaison entre les débits réels mesurés sur le terrain et ceux obtenus en simulation indique que le trafic obtenu est valide d'un point de vue statistique. En particulier, l'utilisation d'un indicateur couramment employé dans le domaine de la simulation de trafic, l'EQMR, montre qu'il est possible de simuler des flux de trafic en carrefour avec un taux d'erreur inférieur à 6% par rapport à la réalité.

Nous avons terminé ce chapitre en présentant les différentes perspectives offertes par notre travail. Nous avons présenté les extensions possibles dans le cadre d'ARCHISIM puis envisagé plusieurs pistes de recherche. L'une d'elles en particulier vise à utiliser le formalisme des réseaux de contraintes distribués de manière à donner une dimension collective à l'anticipation.

Conclusion générale

Le trafic routier est composé de phénomènes complexes issus du déplacement de véhicules sur un réseau de capacité limitée. Au même titre que d'autres phénomènes dynamiques, la simulation de trafic routier s'avère être un problème difficile notamment lorsqu'un certain niveau de détail est recherché. Les simulations de trafic de type microscopique sont de plus en plus courantes. Elles s'intéressent à la reproduction réaliste du déplacement individuel de véhicules et offrent ainsi une granularité extrêmement fine.

Nous avons vu au chapitre 1 que de nombreux outils de simulation microscopique sont basés sur l'utilisation d'équations différentielles (loi de poursuites) faisant le lien entre les différentes variables cinématiques des véhicules. Cette approche analytique manque toutefois de généralité puisque les équations sont obtenues à partir de données mesurées sur le terrain et décrivant une situation de trafic particulière. Une alternative crédible à cette approche analytique se retrouve dans l'utilisation des systèmes multi-agents. Il s'agit d'utiliser le principe d'émergence pour simuler des situations de trafic réalistes. Celles-ci sont la conséquence des différentes actions et interactions des véhicules simulés, autrement dit le trafic est un phénomène global résultant des comportements locaux de conducteurs. Une telle approche est dite comportementale.

La simulation de trafic ARCHISIM développée à l'INRETS est basée sur ce principe. Chaque conducteur simulé est un agent possédant ses propres caractéristiques de conduite et ayant un itinéraire à suivre. L'architecture d'ARCHISIM est distribuée lui permettant d'accueillir un simulateur de conduite. ARCHISIM peut donc être utilisé comme un outil de simulation microscopique de trafic mais aussi comme un générateur de trafic ambiant permettant d'immerger un conducteur humain dans un environnement virtuel dans lequel des conducteurs simulés interagissent avec lui.

L'intérêt d'une approche comportementale par rapport aux approches analytiques classiques est encore plus marqué pour des situations de trafic en carrefour. En effet, les approches à base de lois de poursuites ne sont pas applicables dans les situations impliquant des intersections. La plupart des outils de simulation ont donc souvent recours à des méthodes d'ordonnement entre les véhicules des différentes branches de l'intersection. Nous avons montré au chapitre 2 que ces méthodes constituent une énorme simplification puisque les conflits entre véhicules sont plutôt évités que véritablement résolus. A l'inverse, dans une approche comportementale, le franchissement d'intersections est basé sur la résolution effective des conflits entre véhicules.

Il est intéressant d'étudier la résolution de ces conflits dans un cadre de coordination multi-agent. Les différents mécanismes de coordination existant dans la littérature ont ainsi été présentés au chapitre 2. Un algorithme proposé dans [Champion, 2003] a été en particulier détaillé. Celui-ci repose sur une modélisation des conflits au centre des carrefours. L'aspect comportemental de la tâche de conduite se trouve traité au niveau de la perception des priorités. L'obtention d'un trafic réaliste

passé donc par une reconnaissance pertinente de la situation et du contexte. Une analyse de cet algorithme a permis d'exhiber deux faiblesses. D'une part, l'insertion dans un flux prioritaire manque de réalisme. D'autre part le mécanisme ne permet pas de traiter les éventuels interblocages pouvant apparaître dans le cas de situations de trafic complexes.

En amont de la coordination, il nous a donc semblé intéressant de se focaliser sur la manière dont un agent approchant d'un carrefour va percevoir la situation et va anticiper celle-ci. Nous avons proposé au chapitre 3 un modèle de comportement dit opportuniste et anticipatif. Notre approche opportuniste se définit comme un comportement privilégiant l'intérêt momentané d'un agent aux respects des règles et normes définies pour le système. Transposé au cadre du trafic routier, cela revient à proposer un comportement pouvant transgresser certaines règles du Code de la Route de manière à faciliter le déplacement de conducteurs simulés en carrefour. Le modèle proposé pour ARCHISIM consiste donc en un ensemble de règles d'agrégation permettant aux agents d'évaluer les relations de priorités qu'ils entretiennent avec les autres conducteurs du carrefour. Ces règles prennent en compte des priorités informelles comme celles relatives à la vitesse ou à l'impatience et viennent surcharger les règles du Code de la Route.

Ce type de comportement opportuniste a pour avantage de faciliter l'insertion des véhicules non prioritaires en carrefour, ce qui représentait une faiblesse des précédents travaux [Champion, 2003]. Mais ce comportement augmente également les risques d'interblocages en carrefour en favorisant par exemple le stockage au centre des carrefours. Nous avons donc complété notre modèle décisionnel par une couche anticipative permettant aux agents d'éviter ces interblocages.

Notre approche de l'anticipation est basée sur les travaux de Rosen qui propose une classe particulière : l'anticipation préventive. Elle consiste à partager la représentation de l'environnement d'un agent en deux régions : les états désirés et non désirés. Chaque agent essaie de prédire les effets de ses actions et modifie son comportement à l'instant présent de manière à éviter d'être dans un état non désiré. Cette approche a été reprise par quelques travaux mais aucun n'aborde la manière dont les agents peuvent inférer les effets d'une action et déduire les états non désirés qui peuvent en résulter.

La formalisation que nous proposons repose sur l'utilisation de réseaux de contraintes comme représentation mentale de l'environnement d'un agent. Nous y distinguons deux types d'effets pour une action : les effets directs et indirects. Les effets directs d'une action sont modélisés sous la forme d'un ajout de contraintes dans la représentation de l'environnement. Les effets indirects sont quant à eux obtenus par propagation de ces nouvelles contraintes ajoutées. Les états non désirés que les agents cherchent à éviter apparaissent ainsi comme des inconsistances au sein du réseau de contraintes.

Le chapitre 4 a été consacré à l'implémentation de notre contribution. Nous avons en particulier détaillé les différents choix relatifs à la mise en œuvre de notre modèle d'anticipation dans le cadre d'ARCHISIM. Nous avons montré qu'en pratique la complexité de l'algorithme d'anticipation proposé est raisonnable et n'engendre pas des coûts de calculs pénalisant pour la simulation.

Nous avons exposé les différentes expérimentations réalisées dans le cadre de cette thèse au chapitre 5. Une première partie de notre validation s'est intéressée à une évaluation visuelle des comportements obtenus au sein du simulateur de conduite. Divers scénarii ont été testés de manière à évaluer les capacités de notre algorithme d'anticipation à réduire les phénomènes d'interblocages dans les situations suivantes : forte densité de véhicules en entrée du carrefour, pourcentage de tourne-à-gauche élevé, présence de poids-lourds au centre du carrefour, etc.

Une seconde partie de la validation a consisté à simuler un carrefour existant. Ce travail est le fruit d'une collaboration avec l'Université de Reggio Calabria qui nous a fourni des données statistiques

relatives aux débits et mouvements tournants mesurés sur un carrefour réel. La comparaison entre les débits réels mesurés sur le terrain et ceux obtenus en simulation indique que le trafic obtenu est valide d'un point de vue statistique. En particulier, l'utilisation d'un indicateur, l'EQMR (écart moyen quadratique), montre qu'il est possible de simuler des flux de trafic en carrefour avec un taux d'erreur inférieur à 6% par rapport à la réalité.

La fin du chapitre a porté sur une discussion autour des différentes perspectives envisageables à notre travail. Nous avons étudié la possibilité d'étendre notre modèle à la gestion de l'accélération latérale dans le cadre d'ARCHISIM, puis nous avons terminé par des perspectives de recherche plus générales. Une première perspective consiste à étendre notre modèle d'anticipation aux relations n-aires, autrement dit, aux relations dans lesquelles seront impliquées plus de deux agents. Une seconde perspective vise à formaliser, à l'aide des réseaux de contraintes distribués, une anticipation collective dans laquelle des groupes d'agents anticipent ensemble pour éviter des états non désirés qui leur sont communs.

Bibliographie

- [Aknine *et al.*, 2000] AKNINE, S., PINSON, S. et SHAKUN, M. F. (2000). New coalition formation methods for multi-agent coordination. *Journal of Autonomous Agents and Multi-Agent Systems*.
- [Aldewereld *et al.*, 2005] ALDEWERELD, H., GROSSI, D., VAZQUEZ-SALCEDA, J. et DIGNUM, F. (2005). Designing normative behaviour by the use of landmarks. *In Workshop of Agents, Norms and Institutions for Regulated Multiagent Systems (AAMAS'05)*.
- [Allen, 1984] ALLEN, J. (1984). Towards a general theory of action and time. *Artificial Intelligence*.
- [Auberlet *et al.*, 2003] AUBERLET, J.-M., TRIPODI, A. et GATTUSO, D. (2003). The use of innovative technologies in the case of the space reduction in urban environment : study of a lane keeping system. *In Summer Computer Simulation Conference*, pages 795–800, Montréal, Québec, Canada.
- [Aubin, 1984] AUBIN, J.-P. (1984). *L'analyse non linéaire et ses motivations économiques*, chapitre Jeux à deux personnes. Concepts fondamentaux et exemples. Masson.
- [Barbuceanu *et al.*, 1998] BARBUCEANU, M., GARAY, T. et MANKOVSKI, S. (1998). Coordinating with obligations. *In Proceedings of 2nd International Conference of Autonomous Agents (Agents'98)*.
- [Bazzan, 2005] BAZZAN, A. L. C. (2005). A distributed approach for coordination of traffic signal agents. *Journal of Autonomous Agents and Multi-Agent Systems*, (10):131–164.
- [Bessière, 1992] BESSIÈRE, C. (1992). *Systèmes à contraintes évolutives en intelligence artificielle*. Thèse de doctorat, Université de Montpellier II.
- [Bessière, 1994] BESSIÈRE, C. (1994). Arc-consistency and arc-consistency again. *Artificial Intelligence*.
- [Bessière *et al.*, 1995] BESSIÈRE, C., FREUDER, E. et REGIN, J. (1995). Using inference to reduce arc consistency computation. *In Proceedings of the International Conference on Artificial Intelligence (IJCAI'95)*.
- [Bjørnskau et Elvik, 1992] BJØRNSKAU, T. et ELVIK, R. (1992). Can road traffic law enforcement permanently reduce the number of accidents ? *Accident Analysis and Prevention*, 5(24):507–520.
- [Björklund et Åberg, 2005] BJÖRKLUND, G. M. et ÅBERG, L. (2005). Driver behaviour in intersections : Formal and informal traffic rules. *Transportation Research Part F*, 8:239–253.
- [Blank *et al.*, 2005] BLANK, D. S., LEWIS, J. M. et MARSHALL, J. B. (2005). The multiple roles of anticipation in developmental robotics. *In AAAI 2005 Fall Symposium : From Reactive to Anticipatory Cognitive Embodied Systems*, pages 8–15, Arlington, USA.
- [Boella et Lesmo, 2002] BOELLA, G. et LESMO, L. (2002). A game theoretic approach to norms. *Cognitive Science Quarterly* 2, 2(3-4):492–512.

- [Boella et van der Torre, 2002] BOELLA, G. et van der TORRE, L. (2002). Rational norm creation : attributing mental attitudes to normative systems, part 2. *In Proceedings of 9th International Conference on Artificial Intelligence and Law*.
- [Bonabeau et al., 1999] BONABEAU, E., DORIGO, M. et THERAULAZ, G. (1999). *Swarm Intelligence : From Natural to Artificial Systems*. Addison Wesley.
- [Bonakdarian et al., 1998] BONAKDARIAN, E., CREMER, J., KEARNEY, J. et WILLEMSEN, P. (1998). Generation of ambient traffic for real-time driving simulation. *In Proceeding of IMAGE Conference*, Arizona, USA.
- [Bonte et al., 2006] BONTE, L., ESPIÉ, S. et MATHIEU, P. (2006). Modélisation et simulation des usagers deux-roues motorisés dans ARCHISIM. *In Journées Francophones sur les Systèmes Multi-Agents (JFSMA 2006)*. à paraître.
- [Bou et al., 2006] BOU, E., LÓPEZ-SÁNCHEZ, M. et RODRÍGUEZ-AGUILAR, J. A. (2006). Self-configuration in autonomic electronic institutions. *In COIN (Coordination, Organization, Institutions and Norms in Agent Systems) Workshop at ECAI Conference*, pages 1–9, Riva del Garda, Italie.
- [Bousquet et al., 1998] BOUSQUET, F., BAKAM, I., PROTON, H. et LE PAGE, C. (1998). Cormas : common-pool resources and multi-agent systems. *In Lecture Notes in Artificial Intelligence*, numéro 1416, pages 826–836. Springer-Verlag.
- [Briot et Demazeau, 2001] BRIOT, J.-P. et DEMAZEAU, Y. (2001). *Principes et architectures des systèmes multi-agents*. Hermes Lavoisier.
- [Brooks, 1986] BROOKS, R. A. (1986). A robust layered control system for a mobile robot. *Journal of Robotics and Automation*, pages 14–23.
- [Brooks, 1991] BROOKS, R. A. (1991). Intelligence without reason. *Artificial Intelligence*, 47(1-3):139–159.
- [Butz et al., 2003] BUTZ, M. V., SIGAUD, O. et GÉRARD, P. (2003). *Anticipatory Behavior in Adaptive Learning Systems : Foundations, Theories, and Systems*, chapitre Internal Models and Anticipation in Adaptive Learning Systems. Springer.
- [Carabelea et Boissier, 2005] CARABELEA, C. et BOISSIER, O. (2005). Coordinating agent in organizations using social commitments. *In Proceedings of the 1st International Workshop on Coordination and Organisation*, Namur, Belgique.
- [Castelfranchi et al., 2006] CASTELFRANCHI, C., FALCONE, R. et PIUNTI, M. (2006). Agents with anticipatory behaviors : To be cautious in a risky environment. *In Proceedings of European Conference of Artificial Intelligence 2006 (ECAI06)*.
- [Champion, 2003] CHAMPION, A. (2003). *Mécanisme de coordination multi-agent fondé sur les jeux : application à la simulation comportementale de trafic routier en situation de carrefour*. Thèse de doctorat, Université de Valenciennes.
- [Champion et al., 2002] CHAMPION, A., ZHANG, M.-Y., AUBERLET, J.-M. et ESPIÉ, S. (2002). Behavioural simulation : towards high-density network traffic studies. *In WANG, Xia, N. e. Y., éditeur : Proceedings of the Third International Conference on Traffic and Transportation Studies (ICTTS 2002)*.
- [Cohen et Levesque, 1990] COHEN, P. et LEVESQUE, H. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42:213–261.

-
- [Cohen, 1993] COHEN, S. (1993). *Ingénierie du trafic routier. Eléments de Théorie du Trafic et Applications*. Presses de l'École Nationale des Ponts et Chaussées.
- [Cohen et Semaan, 1990] COHEN, S. et SEMAAN, R. (1990). Un logiciel de simulation comparative de stratégies de fonctionnement de carrefours à feux isolés. *Recherche Transports Sécurité*, (26): 49–54.
- [Condotta, 2000] CONDOTTA, J.-F. (2000). *Problèmes de satisfaction de contraintes : algorithmes et complexité*. Thèse de doctorat, Université Paul Sabatier (Toulouse III).
- [Corkill, 1991] CORKILL, D. D. (1991). Blackboard systems. *AI Expert*, 6(9):40–47.
- [Davidsson, 2003] DAVIDSSON, P. (2003). *Anticipatory Behavior in Adaptive Learning Systems*, chapitre A Framework for Preventive State Anticipation. Springer.
- [Debruyne, 1995] DEBRUYNE, R. (1995). Les algorithmes d'arc-consistance dans les CSP dynamiques. *Revue d'intelligence artificielle*.
- [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.
- [Desner et Stone, 2004] DESNER, K. et STONE, P. (2004). Multiagent traffic management : A reservation-based intersection control mechanism. *In Proceeding of The Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pages 530–537, New-York, USA.
- [Deville et al., 2004] DEVIGNE, D., MATHIEU, P. et ROUTIER, J.-C. (2004). Planning for spatially situated agents. *In International Conference on Intelligent Agent Technology (IAT'04)*, pages 385–388.
- [Deville et van Hentenryck, 1991] DEVILLE, Y. et van HENTENRYCK, P. (1991). An efficient arc consistency algorithm for a class of csp problems. *In Proceedings of the International Conference on Artificial Intelligence (IJCAI'91)*.
- [Dignum, 1997] DIGNUM, F. (1997). Autonomous agents with norms. *Artificial Intelligence and Law*, 7:69–79.
- [Dignum et al., 2000] DIGNUM, F., MORLEY, D., SONENBERG, L. et CAVEDON, L. (2000). Towards socially sophisticated BDI agents. *In Proceedings of ICMAS'2000*.
- [Doniec et al., 2005a] DONIEC, A., ESPIÉ, S., MANDIAU, R. et PIECHOWIAK, S. (2005a). Dealing with multi-agent coordination by anticipation : Application to the traffic simulation at junctions. *In Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS'05)*, Bruxelles.
- [Doniec et al., 2005b] DONIEC, A., ESPIÉ, S., MANDIAU, R. et PIECHOWIAK, S. (2005b). Towards anticipation in traffic simulation at junctions. *In Proceedings of the International Workshop on Modelling Driver Behaviour in Automotive Environments*, pages 242–250, Ispra.
- [Doniec et al., 2006a] DONIEC, A., ESPIÉ, S., MANDIAU, R. et PIECHOWIAK, S. (2006a). Simulation de trafic routier en carrefour : gestion des comportements et des priorités. *In International Workshop Logistique & Transport 2006*, Tunisie.
- [Doniec et al., 2006b] DONIEC, A., MANDIAU, R., PIECHOWIAK, S. et ESPIÉ, S. (2006b). Coordination anticipative multi-agent pour la simulation de trafic. *In 7ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'06)*, pages 401–415.

- [Doniec *et al.*, 2006c] DONIEC, A., MANDIAU, R., PIECHOWIAK, S. et ESPIÉ, S. (2006c). L'anticipation comme modèle d'interaction : application à la coordination multi-agent en simulation. In *15ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA'06)*, Tours.
- [Doniec *et al.*, 2005c] DONIEC, A., PIECHOWIAK, S. et MANDIAU, R. (2005c). A DisCSP solving algorithm based on sessions. In *Recent Advances in Artificial Intelligence : Proceedings of the 18th International Florida Artificial Intelligence Research Society Conference (FLAIRS'05)*, pages 666–670, Clearwater, USA.
- [Doran, 1998] DORAN, J. (1998). Social simulation, agents and artificial societies. In *ICMAS'98 : Proceedings of the Third International Conference on Multi-Agent Systems*.
- [Drogoul, 1993] DROGOUL, A. (1993). *De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi-agents*. Thèse de doctorat, Université de Paris VI.
- [Duboz *et al.*, 2003] DUBOZ, R., RAMAT, E. et PREUX, P. (2003). Scale transfer modelling : Using emergent computation for coupling an ordinary differential equation system with a reactive agent model. *Systems Analysis Modelling & Simulation*, 43(6):793–814.
- [Durfee *et al.*, 1987] DURFEE, E. H., LESSER, V. R. et CORKILL, D. D. (1987). Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, C36(11):1275–1291.
- [Ekdahl, 1999] EKDAHL, B. (1999). Anticipatory systems as linguistic systems. In *Proceedings of CASYS'99 : Computing Anticipatory Systems*, pages 678–689, Liège, Belgique.
- [Ekdahl, 2000] EKDAHL, B. (2000). Agents as anticipatory systems. In *Proceedings of the 4th World Multiconference on Systemic, Cybernetics and Informatics (SCI 2000)*.
- [El Fallah-Seghrouchni *et al.*, 2004] EL FALLAH-SEGHROUCHNI, A., DEGIRMENCIYAN-CARLAULT, I. et MARC, F. (2004). Modelling, control and validation of multi-agent plans in highly dynamic context. In *Proceedings of The Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, New-York, USA.
- [El Hadouaj, 2004] EL HADOUAJ, S. (2004). *Conception de comportements de résolution de conflits et de coordination : Application à une simulation multi-agent du trafic routier*. Thèse de doctorat, Université Paris VI.
- [El Hadouaj *et al.*, 2000] EL HADOUAJ, S., DROGOUL, A. et ESPIÉ, S. (2000). How to combine reactivity and anticipation : the case of conflicts resolution in a simulated road traffic. In MOSS, S. et DAVIDSSON, P., éditeurs : *Multi-Agent-Based Simulation, Second International Workshop*, pages 82–96, Boston, USA. Springer.
- [Erman *et al.*, 1980] ERMAN, L. D., HAYES-ROTH, F., LESSER, V. R. et REDDY, D. R. (1980). The Hearsay-II speech-understanding system : Integrating knowledge to resolve uncertainty. *Computing Surveys*, 2(12):213–253.
- [Espié, 1995] ESPIÉ, S. (1995). Archisim, multi-actor parallel architecture for traffic simulation. In *Proceedings of the Second World Congress on Intelligent Transport Systems*, Yokohama, Japan.
- [Espié *et al.*, 2002] ESPIÉ, S., AUBERLET, J.-M. et ZHANG, M. (2002). Integrated approach for the study of new road profiles. In *DSC'02 : Proceedings of Driving Simulation Conference*, Paris, France.
- [Espié et Saad, 2000] ESPIÉ, S. et SAAD, F. (2000). Driver behaviour modelling and traffic simulation. In *International Ergonomics Association Congress*, San Diego, Californie, USA.

-
- [Esteva *et al.*, 2001] ESTEVA, M., RODRÍGUEZ-AGUILAR, J. A., SIERRA, C., GARCIA, P. et ARCOS, J. L. (2001). On the formal specifications of electronic institutions. *In AgentLink*, pages 126–147.
- [Ferber, 1995] FERBER, J. (1995). *Les systèmes multi-agents : vers une intelligence collective*. InterEditions.
- [Ferber et Michel, 2000] FERBER, J. et MICHEL, F. (2000). Madkit : Une expérience d'architecture de plate-forme multi-agent générique. *In JFIADSMA'00 : 8ème Journée Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents*, pages 223–236.
- [Flipo, 2000] FLIPO, A. (2000). Trust : le simulateur de conduite poids-lourd pour la formation. *In DSC'00 : Proceedings of Driving Simulation Conference*, pages 208–200.
- [Fox, 1987] FOX, M. S. (1987). Constraint-directed search : A case study of job shop scheduling. Rapport technique, San Matco : California.
- [Freska, 1992] FRESKA, C. (1992). Using orientation information for qualitative spatial reasoning. *In Lectures Notes in Computer Science*, volume 639, pages 162–178.
- [Gasser, 1992] GASSER, L. (1992). *Distributed Artificial intelligence : theory and practice*, chapitre DAI approaches to Coordination. Kluwer.
- [Gazis, 1974] GAZIS, D. C. (1974). *Traffic science*, page 293. J. Wiley & Sons, New York.
- [Georgeff, 1983] GEORGEFF, M. P. (1983). Communication and interaction in multi-agents planning. *In Proceedings of AAAI'83*, pages 125–129.
- [Georgé, 2004] GEORGÉ, J.-P. (2004). *Résolution de problèmes par émergence : étude d'un environnement de programmation émergente*. Thèse de doctorat, Université de Toulouse III.
- [Gerard, 2002] GERARD, P. (2002). *Systèmes de classeurs : étude de l'apprentissage latent*. Thèse de doctorat, Université de Paris VI.
- [Gibson, 1981] GIBSON, D. (1981). Available computer models for traffic operations analysis. *In* of SCIENCE, N. A., éditeur : *The Application of Traffic Simulation Models, TRB Special Report 194*.
- [Gintis, 2000] GINTIS, H. (2000). *Game Theory Evolving*. Princeton University Press.
- [Glize, 2000] GLIZE, P. (2000). *L'adaptation des Systèmes à Fonctionnalité Emergente par Auto-Organisation Coopérative* Habilitation à diriger les recherches, Université paul sabatier toulouse.
- [Greenshields, 1935] GREENSHIELDS, B. D. (1935). A study of traffic capacity. *In Proceedings of the Highway Research Board*, volume 14, pages 448–477.
- [Grislin-Le Strugeon *et al.*, 2004] GRISLIN-LE STRUGEON, E., HANON, D. et MANDIAU, R. (2004). Behavioral self-control of agent-based virtual pedestrians. *In Advanced Distributed Systems : 5th International School and Symposium, Lecture Notes in Computer Science*, pages 529–537. Springer.
- [Hachette, 2006] HACHETTE, éditeur (2006). *Dictionnaire français Edition 2006*.
- [Hamadi, 1999] HAMADI, Y. (1999). Optimal distributed arc-consistency. *In Proceedings of Constraint Programming (CP'99)*, pages 219–223.
- [Hanon *et al.*, 2006] HANON, D., GRISLIN-LE STRUGEON, E. et MANDIAU, R. (2006). Un modèle décisionnel orienté comportement utilisant le vote, application à la navigation autonome en environnement simulé. *In 15ème Congrès Francophone Reconnaissance des Formes et Intelligence Artificielle (RFIA'06)*, Tours.

- [Haralick et Elliot, 1980] HARALICK, R. et ELLIOT, G. (1980). Increasing tree search efficiency for constraint satisfaction problem. *Artificial Intelligence*, (14):263–313.
- [Hayes-Roth, 1993] HAYES-ROTH, B. (1993). Opportunistic control of action in intelligent agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6):1575–1587.
- [Helleboogh *et al.*, 2005] HELLEBOOGH, A., HOLVOET, T. et BERBERS, Y. (2005). Simulating actions in dynamic environments. In *Proceedings of Conceptual Modeling and Simulation Conference 2005*, pages 123–128, Marseille, France.
- [Hoffmann, 1993] HOFFMANN, J. (1993). Anticipation and cognition. Gettingen, Germany.
- [Hogeweg et Hesper, 1990] HOGEWEG, P. et HESPER, B. (1990). Individual-oriented modelling in ecology. *Mathematical Computational Modelling*.
- [Hutson *et al.*, 1988] HUTSON, M. A., DEANGELIS, D. L. et POST, W. M. (1988). New computer models unify ecological theory. *Bioscience*.
- [Jennings, 1996] JENNINGS, N. R. (1996). Coordination techniques for distributed artificial intelligence. In *Foundations of Distributed Artificial Intelligence*, pages 187–210. Wiley.
- [Jennings, 1998] JENNINGS, N. R. (1998). An roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38.
- [Kaplan, 1998] KAPLAN, F. (1998). Rôle de la simulation multi-agent pour comprendre l’origine et l’évolution du langage. In HERMES, éditeur : *Systèmes multi-agents : de l’interaction à la socialité (JFIADSMA’98)*, pages 51–64, Paris.
- [Kita *et al.*, 2001] KITA, H., TANIMO, K. et FUKUYAMA, K. (2001). An inverse analysis of merging-giveaway game : focusing on the equilibrium selection. In *Proceedings of the Second Berkeley-Tottori Joint Seminar on Evolution Processes of Transportation Systems*.
- [Kok *et al.*, 2005] KOK, J. R., SPAAN, M. T. et VLASSIS, N. (2005). Non-communicative multi-robot coordination in dynamic environments. *Robotics and Autonomous Systems*, (50):99–114.
- [Koppa, 1997] KOPPA, R. J. (1997). *Traffic flow theory*, chapitre Human factors. Oak Ridge National Laboratory.
- [Koren et Borenstein, 1991] KOREN, Y. et BORENSTEIN, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1398–1404.
- [Kumar, 1992] KUMAR, V. (1992). Algorithms for constraint-satisfaction problems : A survey. *Ai Magazine*.
- [Laird, 2001] LAIRD, J. E. (2001). It knows what you’re going to do : Adding anticipation to a quakebot. In *Proceeding of Autonomous Agent 2001*, Montreal, Canada.
- [Le Petit Robert, 2006] LE PETIT ROBERT, éditeur (2006). *Dictionnaire, Edition 2006*.
- [Leclercq, 2002] LECLERCQ, L. (2002). *Modélisation dynamique du trafic et applications à l’estimation du bruit routier*. Thèse de doctorat, Institut National des Sciences Appliquées de Lyon.
- [Leglise, 1971] LEGLISE, M. (1971). Étude et réalisation d’un modèle de simulation de trafic urbain. *Revue d’informatique et de recherche opérationnelle*, 1:61–85.
- [Lesser et Corkill, 1981] LESSER, V. R. et CORKILL, D. D. (1981). Functionally accurate, cooperative distributed systems. In *IEEE Transactions on Systems, Man and Cybernetics*, pages 81–96.
- [Ligozat, 1998] LIGOZAT, G. (1998). Reasoning about cardinal directions. *Journal of Visual Languages and Computing*, 1(9):23–44.

-
- [Mackworth, 1977a] MACKWORTH, A. (1977a). Consistency in networks of relations. *Artificial Intelligence*, 8:99–118.
- [Mackworth, 1977b] MACKWORTH, A. (1977b). On reading sketch maps. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'77)*, pages 598–606, Cambridge MA.
- [Malone, 1998] MALONE, T. (1998). What is coordination theory. In *National Science Foundation Coordination Theory Workchop*.
- [Malone, 1987] MALONE, T. W. (1987). Modeling coordination in organizations and markets. *Management Sciences*, 33(10):1317–1332.
- [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, 2000] MANDIAU, R. (2000). *Modélisation et évaluation d'organisations multi-agents*. Habilitation à diriger les recherches, Université de Valenciennes et du Hainaut-Cambrésis.
- [Mandiau et al., 2000] MANDIAU, R., LE STRUGEON, E. et AGIMONT, G. (2000). Study of efficiency of multi-agent organizational structure. In *ICMAS'00 : Proceeding of the 4th International Conference on Multi-Agent Systems*.
- [Marc, 2005] MARC, F. (2005). *Plannification sous Contraintes dans un Contexte Dynamique : applications aux Simulations Aériennes*. Thèse de doctorat, Université Pierre et Marie Curie.
- [Marshall et al., 2004] MARSHALL, J., BLANK, D. et MEEDEN, L. (2004). An emergent framework for selfmotivation in developmental robotics. In *Proceedings of the 3rd International Conference on Development and Learning*, pages 104–111, La Jolla, USA.
- [Mathieu et Picault, 2006] MATHIEU, P. et PICAULT, S. (2006). Vers une représentation des comportements centrée interactions. In *15ème Congrès Francophone Reconnaissance des Formes et Intelligence Artificielle (RFIA'06)*, Tours.
- [Michon, 1985] MICHON, J. A. (1985). A critical view of driver behavior models : What do we know, what should we do ? In *Human Behavior and Traffic Safety*, pages 485–520. New York : Plenum Press.
- [Minsky, 1988] MINSKY, M. (1988). *La société de l'esprit*. InterEditions.
- [Minsky et Ungureanu, 1997] MINSKY, N. H. et UNGUREANU, V. (1997). Regulated coordination in open distributed systems. In *Proceedings of the 2nd International Conference on Coordination Models and Languages*.
- [Mohr et Henderson, 1986] MOHR, R. et HENDERSON, T. (1986). Arc and path consistency revisited. *Artificial Intelligence*.
- [Müller, 2004] MÜLLER, J.-P. (2004). Emergence of collective behaviour and problem solving. In *Engineering Societies in the Agents World IV, 4th International Workshop, ESAW 2003*, Lecture Notes in Computer Science, pages 1–21, Londres, UK.
- [Neveu et Berlandier, 1994] NEVEU, B. et BERLANDIER, P. (1994). Arc-consistency for dynamic constraint satisfaction problems : An RMS free approach. In *Proceeding of the European Conference of Artificial Intelligence (ECAI'94)*.
- [Nwana et al., 1996] NWANA, H. S., LEE, L. C. et JENNINGS, N. R. (1996). Co-ordination in software agent systems. *The British Telecom Technical Journal*, 4(14):79–88.
- [Parsons et Wooldridge, 2000] PARSONS, S. et WOOLDRIDGE, M. (2000). Game theoretic and decision theoretic agents. *The Knowledge Engineering Review*.

- [Paruchuri *et al.*, 2002] PARUCHURI, P., PULLALAREVU, A. R. et KARLPALEM, K. (2002). Multi agent simulation of unorganized traffic. In *Proceeding of The International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, Bologne, Italie.
- [Piaget, 1999] PIAGET, J. (1999). *Psychologie de l'intelligence*. Armand Colin, 7 édition.
- [Picard et Gleizes, 2005] PICARD, G. et GLEIZES, M.-P. (2005). Cooperative self-organization to design robust and adaptive collectives. In *Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS'05)*, Bruxelles.
- [Picard et Gleizes, 2006] PICARD, G. et GLEIZES, M.-P. (2006). Auto-organisation coopérative pour la conception de collectifs adaptatifs et robustes. In *7ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'06)*.
- [Piechowiak, 2001] PIECHOWIAK, S. (2001). *Raisonnement à base de contraintes : Application au diagnostic*, Habilitation à diriger les recherches, Université de Valenciennes et du Hainaut-Cambrésis.
- [Piechowiak et Hamadi, 2002] PIECHOWIAK, S. et HAMADI, Y. (2002). *Organisation et applications des SMA*, chapitre Problèmes de satisfaction de contraintes et systèmes multi-agents, pages 169–205. Hermes.
- [Piroird, 1997] PIROIRD, J. (1997). Toll system design with a driving simulator. In *DSC'97 : Proceedings of Driving Simulation Conference*.
- [Ramat *et al.*, 1998] RAMAT, E., PREUX, P., SEURONT, L. et LAGADEUX, Y. (1998). Modélisation et simulation multi-agents en biologie marine - étude du comportement du copépode. In *SMAGET'98 : Systèmes Multi-Agents pour la Gestion de l'Environnement et des Territoires*, Clermont-Ferrand.
- [Reece et Shafer, 1993] REECE, D. A. et SHAFER, S. A. (1993). A computational model of driving for autonomous vehicles. *Transportation Research*, 27(1):23–50.
- [Reynolds, 1987] REYNOLDS, C. (1987). Flocks, herds, and schools : A distributed behavior model. In *Proceedings of ACM SIGGRAPH*, pages 25–34.
- [Riegler, 2001] RIEGLER, A. (2001). The role of anticipation in cognition. In *Computing Anticipatory Systems. Proceedings of the American Institute of Physics 573*, pages 534–541.
- [Ronald *et al.*, 1999] RONALD, E., SIPPER, M. et CAPCARRÈRE, M. S. (1999). Design, observation, surprise ! a test of emergence. *Artificial Life*, (5):225–239.
- [Rosen, 1974] ROSEN, R. (1974). Planning, management, policies and strategies : Four fuzzy concepts. *International Journal of General Systems*.
- [Rosen, 1985] ROSEN, R. (1985). *Anticipatory Systems - Philosophical, Mathematical and Methodological Foundations*. Pergamon Press.
- [Rosenschein, 1985] ROSENSCHEIN, J. S. (1985). *Rational interaction : cooperation among intelligent agents*. Thèse de doctorat, Stanford University.
- [Rosenschein *et al.*, 1986] ROSENSCHEIN, J. S., GINSBERG, M. et GENESERETH, M. R. (1986). Cooperation without communication. *Proceedings of the Fifth National Conference on Artificial Intelligence, AAAI'86*.
- [Rothrock et Keefer, 1957] ROTHROCK, C. A. et KEEFER, L. A. (1957). Measurement of urban traffic congestion. *Highway Research Board Bulletin*, 156:1–13.

-
- [Ruskin et Wang, 2002] RUSKIN, H. J. et WANG, R. (2002). Modelling traffic flow at an urban unsignalized intersection. *In Proceedings of International Conference on Computational Science*, pages 381–390.
- [Russell et Norvig, 1995] RUSSELL, S. et NORVIG, P. (1995). *Artificial intelligence : A Modern Approach*. Prentice-Hall.
- [Saad, 1992] SAAD, F. (1992). In-depth analysis of interactions between drivers and the road environment : contribution of on-board observations and subsequent verbal report. *In Proceedings of the 4th Workshop of ICTCT*, University of Lund.
- [Saad et al., 1989] SAAD, F., ELSANDE, P. V., DELHOMME, P., LEPESANT, C. et GAUCHÉ, T. (1989). Analyse des comportements en situations de conduite critique : le franchissement d’intersections. Rapport technique, Conversion DSCR-INRETS.
- [Sanella et al., 1992] SANELLA, M., FREEMAN-BENSON, B., MALONEY, K. et BORNIG, A. (1992). Multi-way versus one-way constraints in user interfaces : experience with deltablue algorithm. Rapport technique, Université de Washington.
- [Schelling, 1978] SCHELLING, T. C. (1978). *Micromotives and Macrobehavior*. W. W. Norton & Company.
- [Seward, 1949] SEWARD, J. P. (1949). An experimental analysis of latent learning. *Journal of Experimental Psychology*.
- [Shoham et Tennenholtz, 1995] SHOHAM, Y. et TENNENHOLTZ, M. (1995). Computational social systems : offline design. *Journal of Artificial Intelligence*, 1-2(73).
- [Simonin, 2001] SIMONIN, O. (2001). *Le modèle satisfaction-altruisme : coopération et résolution de conflits entre agents situés réactifs, application à la robotique*. Thèse de doctorat, Université de Montpellier II.
- [Smith, 1980] SMITH, R. G. (1980). The contract net protocol. *IEEE Transactions on Computers*, C-29(2).
- [Stolzmann, 1998] STOLZMANN, W. (1998). Anticipatory classifier systems. *In Genetic Programming 1998 : Proceedings of the Third Annual Conference*, San Francisco, USA.
- [Suksthankar et al., 1996] SUKSTHANKAR, R., HANCOCK, J., BALUJA, S., POMERLEAU, D. et THORPE, C. (1996). Adaptive intelligent vehicle modules for tactical driving. *In Proceedings of AAAI-1996 Workshop on Intelligent Adaptive Agents*.
- [Summala, 2005] SUMMALA, H. (2005). Towards understanding driving behaviour and safety efforts. *In Proceedings of the International Workshop on Modelling Driver Behaviour in Automotive Environments*, pages 105–111, Ispra, Italie.
- [Sycara, 1989] SYCARA, K. P. (1989). Multi-agent compromise via negotiation. *Distributed Artificial Intelligence*, 2:119–138.
- [Sycara et al., 1991] SYCARA, K. P., s. ROTH, SADEH, N. et FOX, M. (1991). Distributed constrained heuristic search. *In IEEE Transactions on Systems, Man and Cybernetics*, pages 1446–1461.
- [Tanenbaum, 2000] TANENBAUM, A. (2000). *Réseaux*. Dunod / Prentice Hall, 4 édition.
- [Tolman, 1932] TOLMAN, E. (1932). *Purposive behavior in animals and men*. New-York : century.
- [Tranois et al., 1998] TRANOIS, H., LEBRUN, A. et DELEAGE, J.-L. (1998). Utilisation de l’intelligence artificielle distribuée pour la simulation microscopique d’un carrefour. *Recherche Transport Sécurité*, pages 33–42.

- [Tricot, 2005] TRICOT, N. (2005). *Conception et évaluation de systèmes coopératifs avancés d'aide à la conduite automobile*. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis.
- [Tricot et al., 2006] TRICOT, N., RAJAONAH, B., POPIEUL, J. et MILLOT, P. (2006). Design and evaluation of an advanced driver assistance system : The case of an auto adaptive cruise control. *Le Travail Humain*, (69):129–152.
- [Tsang, 1993] TSANG, E. (1993). *Foundations of Constraint Satisfaction*. Academic Press Limited.
- [Van Dyke Parunak et al., 1998] VAN DYKE PARUNAK, H., SAVIT, R. et RIOLO, R. L. (1998). *Multi-Agent Systems and Agent-Based Simulation*, chapitre Agent-Based Modeling vs. Equation-Based Modeling : A Case Study and User's Guide, pages 10–26. Springer Verlag.
- [Vanbergue et Drogoul, 2002] VANBERGUE, D. et DROGOUL, A. (2002). Approche multi-agent pour la simulation urbaine. *In Actes des journées CASSINI*, pages 95–112, Brest, France.
- [Verrons, 2004] VERRONS, M.-H. (2004). *La négociation entre agents virtuels*. Thèse de doctorat, Université des Sciences et Technologies de Lille.
- [Waltz, 1975] WALTZ, D. (1975). Understanding line drawings of scenes with shadows. *In The Psychology of Computer Vision*.
- [Wang et al., 2004] WANG, H., KEARNEY, J. K., CREMER, J. et WILLEMSSEN, P. (2004). Steering autonomous driving agents through intersections in virtual urban environments. *In MSV'04 : Proceedings of International Conference on Modeling, Simulation and Visualization Methods, Las Vegas, Nevada, USA*. CSREA Press.
- [Wang et Ruskin, 2006] WANG, R. et RUSKIN, H. J. (2006). Modelling traffic flow at multi-lane urban roundabouts. *International Journal of Modern Physics C (IJMPC)*, 17(3).
- [Yokoo, 2001] YOKOO, M. (2001). *Distributed Constraint Satisfaction : Foundations of Cooperation in Multi-agent Systems*. Springer Verlag.
- [Zeghal, 1993] ZEGHAL, K. (1993). Un modèle de coordination d'actions réactive appliqué au trafic aérien. *In Actes des Journées Francophones d'Intelligence Artificielle Distribuée et Systèmes Multi-Agent : JFIADSMA'93*.
- [Zeigler, 1990] ZEIGLER, B. P. (1990). *Intelligent Agents and Endomorphic Systems*, chapitre Object-Oriented Simulation with Hierarchical Modular Models. Academic Press.

Résumé

Les systèmes multi-agents permettent la simulation de phénomènes complexes difficilement descriptibles de manière analytique. Cette approche repose souvent sur la coordination d'agents dont l'ensemble des actions et interactions entraînent l'émergence du phénomène à simuler. Lorsqu'il s'agit de reproduire le comportement humain, il faut parfois composer avec une coordination compétitive et être capable de reproduire des comportements anticipatifs. Dans cette thèse, nous nous sommes intéressés à la problématique de reconnaissance et d'anticipation du contexte de coordination. Nos travaux se basent sur une approche de l'anticipation proposée par Rosen : "l'anticipation préventive" qui consiste à adapter son comportement courant de manière à éviter un certain nombre d'états non désirés. Nous proposons une formalisation de ce concept basée sur la manipulation et le traitement de réseaux de contraintes. Le modèle décisionnel sous-jacent utilise des techniques de propagation de contraintes permettant à chaque agent d'inférer les effets d'une action et de détecter ainsi un certain nombre d'états non désirés. L'algorithme proposé est générique et permet de filtrer, en amont de la coordination, toutes actions conduisant à l'un des états non désirés du système. Implémentée dans le cadre d'une simulation de trafic développée à l'INRETS : ARCHISIM, notre proposition permet d'introduire des comportements opportunistes plus réalistes sans risque d'apparition d'interblocage au centre des carrefours. Les différentes évaluations menées ont permis de valider l'émergence des phénomènes de trafic obtenus confirmant ainsi la pertinence de l'approche.

Mots clés : systèmes multi-agents, anticipation, coordination, simulation, trafic routier

Abstract

Multi-agent systems allow the simulation of complex phenomena which are not easily describable in an analytical way. This approach is often based on the coordination of agents whose actions and interactions involve the emergence of the phenomenon to be simulated. When the aim is to mimic the human behaviour, it is sometimes necessary to use a competitive coordination and to be able to reproduce anticipatory behaviours. This thesis addresses the issue of recognition and anticipation of coordination context. Our work is based on an approach of the anticipation suggested by Rosen : "preventive anticipation" which consists in adapting its actual behaviour to avoid some undesired states. We propose a formalization of this concept based on constraints networks processing. The underlying decisional model uses constraints propagation techniques which allow each agent to gather the effects of an action and thus to detect some undesired states. The proposed algorithm is generic and filters out, before any coordination, all actions leading to one of the non-desired states of the system. Implemented within the framework of a simulation of traffic developed at INRETS and called ARCHISIM, our proposal enable to introduce opportunistic behaviours without any risk of having deadlock in the center of the intersection. The various evaluations validate the emergence of traffic phenomena and confirm the relevance of the approach.

Keywords : multi-agent systems, anticipation, coordination, simulation, road traffic

