



HAL
open science

Coordination locale et optimisation distribuée du trafic de véhicules autonomes dans un réseau routier

Mohamed Tlig

► **To cite this version:**

Mohamed Tlig. Coordination locale et optimisation distribuée du trafic de véhicules autonomes dans un réseau routier. Intelligence artificielle [cs.AI]. Université de Lorraine, 2015. Français. NNT : 2015LORR0049 . tel-01251168

HAL Id: tel-01251168

<https://inria.hal.science/tel-01251168v1>

Submitted on 5 Jan 2016

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.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



UNIVERSITÉ
DE LORRAINE

Département de formation doctorale en informatique
UFR STMIA

École doctorale IAEM Lorraine

Coordination locale et optimisation distribuée du trafic de véhicules autonomes dans un réseau routier

THÈSE

présentée et soutenue publiquement le 26 mars 2015

pour l'obtention du

Doctorat de l'université de Lorraine
(spécialité informatique)

par

Mohamed TLIG

Composition du jury

<i>Président :</i>	Abder Koukam	Professeur, Université de Technologie de Belfort-Montbéliard
<i>Rapporteurs :</i>	René Mandiau	Professeur, Université de Valenciennes et du Hainaut-Cambrésis
	Philippe Mathieu	Professeur, Université Lille 1
<i>Examineurs :</i>	Rochdi Merzouki	Professeur, École Polytech-Lille , Université Lille 1
	Ammar Oulamara	Professeur, Université de Lorraine
<i>Directeur de thèse :</i>	Olivier Simonin	Professeur, INSA de Lyon, Université de Lyon
<i>Co-Directeur de thèse :</i>	Olivier Buffet	Chargé de recherche, INRIA Nancy Grand Est

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503



Mis en page avec la classe thloria.

Remerciements

Je tiens à exprimer mes sincères remerciements et ma reconnaissance à mes directeurs de thèse, qui m'ont accompagné et soutenu tout au long de ce travail de thèse. Leurs grandes qualités humaines, leurs conseils scientifiques, et leurs critiques constructives ont rendu ce travail particulièrement enrichissant. Je tiens à les remercier pour m'avoir communiqué leur passion pour la recherche scientifique.

Je voudrais remercier tout d'abord Olivier Simonin pour m'avoir transmis sa vision de l'aventure scientifique. Je suis très reconnaissant pour la confiance et la liberté qu'il m'a accordé, mais aussi pour tous les conseils précieux qu'il a su me communiquer. Je remercie Olivier Buffet pour sa disponibilité quotidienne et son implication. Ses commentaires et ses idées ont été d'une aide précieuse tout au long de ce travail. Je n'oublierai pas non plus sa rigueur scientifique et son impact sur ce travail.

Je tiens à vous dire que j'ai apprécié tout au long de ces années les échanges, scientifiques et amicaux, qui ont caractérisé cette collaboration.

Je remercie vivement René Mandiau et Philippe Mathieu d'avoir accepté d'être mes rapporteurs et d'avoir bien voulu évaluer ce travail de thèse. Je suis d'autant plus reconnaissant de la qualité de leurs lectures, de leurs remarques, et de leurs nombreuses suggestions, qu'elles me permettent d'envisager de plus riches perspectives.

Je remercie également Abder Koukam, Rochdi Merzouki et Ammar Oulamara pour leur participation au jury de cette thèse et pour le temps qu'ils ont bien voulu consacrer pour commenter mes travaux. Merci pour cette dernière discussion riche par vos remarques, et qui aura sans nul doute une influence sur ma manière d'aborder mes travaux par la suite.

Je pense aussi à notre chef d'équipe François Charpillet qui m'a accueilli pendant tout ce temps dans l'équipe MAIA, une équipe dynamique et dans laquelle j'ai eu la chance de trouver une ambiance sympathique et motivante.

Mes remerciements vont également aux membres de l'équipe MAIA et de l'équipe LARSEN avec qui c'était un bonheur de travailler. Je les remercie infiniment pour leurs amitiés et leurs apports scientifiques à mes travaux.

Enfin, je voudrais exprimer ma reconnaissance et ma gratitude à toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce travail de thèse.

Pour terminer, je tiens à remercier mon père, ma mère et mon frère. Leur affection et leur soutien m'ont été d'un grand secours tout au long de ma vie professionnelle et personnelle. Je vous remercie du fond du cœur d'être toujours à mes côtés et de me donner cette force pour continuer.

Je vous dédie ce travail en signe de reconnaissance.

Table des matières

Table des figures	xi
Liste des tableaux	xv
Liste des Algorithmes	xvii

Chapitre 1 Introduction
--

1.1 Le projet InTraDE	1
1.2 Objectifs de la thèse	3
1.3 Approche suivie dans la thèse	4
1.4 Plan de la thèse	5

Partie I Partage d'une voie par deux files de véhicules autonomes

Introduction

Chapitre 2

État de l'art sur le problème du partage d'une voie

2.1	Scénario du partage d'une voie	14
2.2	Modélisation sans contrôle du trafic	15
2.2.1	Modélisation en général	15
2.2.1.1	Les modèles macroscopiques	15
2.2.1.2	Les modèles microscopiques	15
2.2.2	Modélisation du trafic interrompu par un accident	16
2.2.3	Analyse et estimation du retard	17
2.3	Modélisation et contrôle du trafic	19
2.3.1	Modélisation pour la simulation	19
2.3.2	Contrôleur pour minimiser les retards	21
2.4	Bilan	21

Chapitre 3

Proposition de solutions par coordination réactive

3.1	Formalisation du problème	24
3.1.1	Modèle du réseau de transport	24
3.1.2	Modèle des agents	25
3.1.3	Critères d'optimisation	27
3.2	Comportements de coordination	29
3.2.1	Approche par alternance	29
3.2.2	Optimisation locale à court terme (OLC)	31
3.3	Résultats expérimentaux	32
3.3.1	Simulation	32
3.3.2	Libération des deux voies	33

3.3.3	Régulation d'un trafic continu	34
3.4	Simulation avec SCANerstudio	36
3.5	Conclusion	36

Partie II Optimisation d'un réseau d'intersection pour véhicules autonomes

Introduction

Chapitre 4

Approches existantes pour la gestion d'intersections

4.1	Le code de la route	45
4.1.1	Règles de priorité	45
4.1.2	Les panneaux de signalisation	46
4.1.3	Les feux de circulation tricolores	47
4.1.3.1	Définition d'un flux, d'un cycle et d'une phase	47
4.2	Quel est le problème ?	48
4.3	Quatre approches de référence	49
4.3.1	TRANSYT	49
4.3.2	SCOOT	50
4.3.3	SCATS	51
4.3.4	PRODYN	51
4.3.5	Limites des approches de référence	52

4.4	Synthèse des approches	53
4.4.1	Perceptions de l'environnement	53
4.4.2	Variables de contrôle	54
4.4.3	Espace discret vs continu	54
4.4.4	Critères d'optimisation	54
4.4.5	Centralisé vs décentralisé	55
4.4.5.1	Contrôle	55
4.4.5.2	Optimisation	55
4.4.6	Adaptation instantanée vs adaptation moyenne	56
4.4.7	Algorithmes d'optimisation	56
4.4.7.1	Algorithmes heuristiques	56
4.4.7.2	Optimisation boîte noire	58
4.4.7.3	Algorithmes de contrôle	60
4.4.8	Les limites des approches	64
4.5	Approche pour véhicules autonomes (IAV)	66
4.5.1	Approche reposant sur l'historique des attentes des véhicules	66
4.5.2	Négociation véhicule à véhicule	66
4.5.3	Approche par réservations	67
4.6	Bilan	69

<p>Chapitre 5 Modélisation et simulation</p>

5.1	Choix de modélisation	74
5.1.1	Modélisation du réseau	74
5.1.1.1	Réseau et trafic	75
5.1.1.2	Les véhicules	76

5.1.1.3	Les intersections	77
5.1.2	Autonomie, intelligence et communication	78
5.2	Critères d'optimisation	79
5.2.1	L'énergie	79
5.2.2	Le temps	79
5.3	Simulateur	80
5.3.1	Les simulateurs existants	80
5.3.2	Pourquoi un nouveau simulateur?	81
5.3.3	Le simulateur développé	82
5.4	Bilan	84

Chapitre 6

Synchronisation de flux de véhicules dans une intersection

6.1	Synchronisation d'un croisement	86
6.1.1	Approche par synchronisation locale	86
6.1.2	Période minimale de croisement	87
6.1.2.1	Cas $\theta = 90^\circ$	87
6.1.2.2	Cas $\theta < 90^\circ$	88
6.1.2.3	Cas $\theta > 90^\circ$	89
6.1.3	Synchronisation temporelle	90
6.1.4	Routes à deux voies opposées	92
6.1.5	Adaptation des vitesses	94
6.1.6	Rayons de la zone de contrôle	96
6.2	Expérimentations	97
6.2.1	Simulations	98
6.2.2	Comparaison des stratégies	98

6.2.3	Effet de variation du paramètre R (rayon d'action)	100
6.2.4	Effet de variation du paramètre D (débit)	100
6.3	Synchronisation n -véhicules/ p -véhicules	100
6.4	Conclusion	102

<p>Chapitre 7 Synchronisation des intersections d'un réseau</p>
--

7.1	Notations et hypothèses	104
7.1.1	Contrainte de périodicité	106
7.1.2	Contrôle des phases	106
7.2	Critères d'optimisation séparables	107
7.3	Coordination et optimisation des phases des intersections	110
7.3.1	Hill-Climbing (HC)	110
7.3.2	Paysages des fonctions à optimiser	111
7.3.2.1	Périodicité	111
7.3.2.2	Temps	111
7.3.2.3	Énergie	112
7.3.2.4	Similarité entre les deux critères	113
7.3.3	Algorithme de recherche stochastique distribuée (DSA)	114
7.3.3.1	Principe de la recherche stochastique distribuée	114
7.3.3.2	Besoin du saut $T/2$	114
7.3.3.3	Algorithme proposé	115
7.3.4	De l'optimisation hors ligne à l'optimisation en ligne	115
7.3.4.1	L'optimisation hors ligne	116
7.3.4.2	L'optimisation en ligne	116
7.4	Stratégie premier arrivé premier servi (FCFS)	117

7.5	Expérimentations	118
7.5.1	Paramètres de simulation	118
7.5.2	Courbes d'optimisation obtenues	119
7.5.3	Optimisation hors ligne	121
7.5.4	Optimisation en ligne	122
7.6	Conclusion	123

Chapitre 8

Conclusion de la thèse

8.1	Partage d'une voie par deux files de véhicules	125
8.1.1	Synthèse des contributions	125
8.1.2	Perspectives	126
8.2	Optimisation d'un réseau d'intersections	126
8.2.1	Synthèse des contributions	126
8.2.2	Perspectives	127

Annexe A

Détails de calcul de la synchronisation de flux de véhicules dans une intersection

A.1	Adaptation des vitesses	131
A.2	Rayon minimum r_0	133

Annexe B

Exemple d'un fichier définissant un réseau de transport
--

Table des figures

1.1	Véhicule intelligent autonome InTraDE	3
2.1	Deux flux de véhicules bloqués par un obstacle	14
2.2	Trafic interrompu par un accident [Baykal-Gürsoy <i>et al.</i> , 2009a]	17
2.3	Problème d'interférence entre deux files	18
2.4	Route composée de plusieurs voies [Hidas, 2002]	19
3.1	Représentation du réseau considéré	25
3.2	Le modèle action-décision	26
3.3	Un scénario particulier illustrant les temps à additionner pour estimer la durée du parcours restant pour V_1	28
3.4	Première approche : alterner les véhicules	29
3.5	Illustration de la deuxième approche : optimisation locale à court terme	31
3.6	Le simulateur	33
3.7	Observation de l'évacuation de 100 véhicules	34
3.8	Comparaison des moyennes et des écarts types des temps des traversées (T=10-10)	35
3.9	Observation d'une simulation de chaque critère, date d'entrée et durée de traversée des véhicules à (T=10-10)	36
3.10	Simulation avec SCANeRstudio de l'approche par Alternance (figures 1 et 2), et de l'approche OLC (figures 3 et 4)	37
4.1	Plan d'un cycle composé de 4 phases	48
4.2	Boucles magnétiques en amont des carrefours (SCOOT)	50
4.3	Illustration du modèle de France et Ghorbani [2003] avec une architecture hiérarchique composée de six agents AL et un agent AC	57

4.4	Présentation du modèle (automates cellulaires) de Brockfeld <i>et al.</i> [2001] : les routes ont deux directions (Nord et Est) et sont composées d'une seule voie.	59
4.5	Illustration du modèle de Rochner <i>et al.</i> [2006]	64
4.6	Coordination entre intersections	65
4.7	Modélisation de la géométrie des intersections	67
4.8	Discrétisation de la zone de croisement par l'agent manager	68
5.1	Croisements à 2, 4 flux et angle variable	75
5.2	Dimension d'un véhicule	76
5.3	Rayon d'action d'une intersection (R) et zone de croisement (r_0)	78
5.4	Le simulateur développé avec un réseau de 12 intersections	82
5.5	Architecture du simulateur (modules et dépendances)	83
6.1	Zone de croisement de deux flux avec un angle $\theta = 90^\circ$	87
6.2	Zone de partage	88
6.3	Croisement de deux flux avec un angle $\theta < 90^\circ$	89
6.4	Croisement de deux flux avec un angle $\theta > 90^\circ$	90
6.5	Principe de synchronisation d'une intersection	90
6.6	Zone de croisement de deux routes avec 4 flux	92
6.7	Animation de croisement de quatre véhicules	93
6.8	Temps nécessaire pour passer de la figure 6.7-4 à la figure 6.7-1	94
6.9	Profil d'accélération	95
6.10	Un véhicule en direction de l'intersection entre R et r_0	97
6.11	Illustration du simulateur	98
6.12	Synchronisation vs. Feux de circulation à une intersection	99
6.13	Mesure du retard moyen des véhicules à la variation du débit	101
6.14	Croisement de deux flux $\theta = 90^\circ$, $n = 3$, $p = 2$	101
7.1	Réseau composé de 5 intersections ; synchroniser les intersections implique le contrôle des déphasages de chacune d'elles	104
7.2	L'intersection i est le résultat du croisement de deux routes à deux voies	105
7.3	Représentation des temps de passage vus comme des signaux périodiques selon les 4 directions de circulation	105

7.4	La phase optimale ϕ_i de l'intersection i dépend des phases de ses voisins	107
7.5	Mesurer la performance d'un véhicule sur le segment $i \rightarrow j$ requière la décomposition de ce segment en trois parties : (1) entre i et la zone de contrôle de j de rayon R , (2) la phase d'adaptation dans la zone de contrôle entre R et r_0 , et (3) la zone de croisement de rayon r_0	109
7.6	Le critère temps en fonction de la phase : a - critère temps pour les deux segments entrant et sortant du flux nord, b - critère temps pour une intersection (la somme des 4 flux de 2 segments).	111
7.7	Optimiser la voie orientée vers le nord et composée de deux segments (l'entrant et le sortant) revient à optimiser ϕ_i pour le flux f_{Nord}^{entree} et f_{Nord}^{sortie}	112
7.8	Le critère énergie en fonction de la phase : a - critère énergie pour les deux segments entrant et sortant du flux nord, b - critère énergie pour une intersection	112
7.9	Comparaison entre les courbes d'optimisation obtenues en fonction du critère énergie (a, b, c) et le critère temps (d, e, f) pour les mêmes trois configurations	113
7.10	Différence entre l'espace de recherche des algorithmes HC et DSA	115
7.11	Principe de fonctionnement de l'approche First Come First Served	118
7.12	Réseau de simulation utilisé avec 12 intersections	119
7.13	Comparaison entre l'évolution des critères locaux (énergie) en fonction des itérations dans HC (a, b, c) et dans DSA (d, e, f)	120
A.1	Profil d'accélération	131
A.2	Le rayon minimum r_0 (pour deux routes à une seule voie chacune)	134
A.3	r_0 en fonction de θ pour deux routes à deux voies chacune et d'une largeur de $10m$	135

Liste des tableaux

3.1	Résumé des moyennes et des écarts types	35
4.1	Résumé des solutions citées	70
4.2	Présentation de toutes les solutions citées selon les différentes dimensions identifiées	71
6.1	Moyennes et Écarts-types de l'énergie consommée en fonction du rayon d'action $R = 50m$, $R = 100m$ et $R = 200m$	100
7.1	Energie cinétique moyenne consommée par véhicule, utilisant FCFS dans ses versions asynchrone, synchrone ou optimisé HC, et Alt dans ses versions asynchrone, synchrone, optim HC et optim DSA ; Les injections sont mesurées en véhicules/minute	121
7.2	Moyenne des temps de traversées par véhicule, utilisant FCFS dans ses versions asynchrone, synchrone ou optimisé HC, et Alt dans ses versions asynchrone, synchrone, optim HC et optim DSA ; Les injections sont mesurées en véhicules/minute	122
7.3	Moyenne de l'énergie consommée et des temps de traversées par véhicule utilisant Alt avec des phases optimisées hors ligne, Alt avec des phases optimisées en ligne, et FCFS avec des phases aléatoires	123

Liste des Algorithmes

3.1	Règles pour le passage alterné d'un véhicule	30
3.2	Règles de passage de OLC pour le véhicule β	32
6.1	Contrôle d'un flux par l'agent de contrôle	91
7.1	Évaluation de la phase actuelle pour l'intersection i	109
7.2	Hill-Climbing-iteration($i, t, \vec{\phi}_t$)	110
7.3	DSA($i, t, \vec{\phi}_t$)	115

Chapitre 1

Introduction

Durant les dernières décennies, les progrès réalisés dans les domaines de l'intelligence artificielle ont conduit l'homme à traiter des problèmes de plus en plus complexe. Parmi ces problèmes, nous nous intéressons dans cette thèse aux problèmes des transports. Les modèles multi-agents sont aujourd'hui considérés comme un moyen prometteur pour déployer des applications qui, souvent, doivent traiter des problèmes complexes et interopérer avec d'autres systèmes hétérogènes. Ils sont utilisés lorsque le système étudié est de grande taille et repose sur l'interaction de différentes entités. Dès lors, ils permettent d'assister l'homme en résolvant des problèmes d'abstraction élevée comme dans les systèmes d'aide à la décision. Ils offrent des abstractions logicielles de haut niveau nécessaires pour gérer des applications comme dans le domaine des transports.

Nous nous intéressons, dans cette thèse, aux comportements réactifs et aux systèmes multi-agents réactifs [Ferber, 1999] dans le cadre de la résolution de conflits et de la régulation du trafic de véhicules autonomes. Les comportements réactifs, même s'ils ne présentent pas de capacités de raisonnement, permettent d'agir en temps réel. Les agents réactifs sont aussi connus pour leur limite calculatoire ainsi que pour leur lenteur à s'auto-organiser. Conscients de ces limites, nous examinons dans cette thèse comment les approches réactives peuvent être combinées avec les approches fondées sur l'optimisation. Tout au long de cette thèse, nous essayons de rapprocher ces deux thématiques tout en traitant des problèmes de complexité croissante.

1.1 Le projet InTraDE

Cette thèse s'inscrit dans le cadre du projet européen InTraDE¹ (**I**ntelligent **T**ransportation in **D**ynamic **E**nvironment). Dans cette section, nous présentons brièvement le projet et ses objectifs.

Le commerce maritime s'est développé d'une manière considérable durant la dernière décennie. Cela est dû à la mondialisation ainsi qu'au développement continu des pays

1. www.InTraDE-nwe.eu

émergents. Cette croissance mondiale a influencé l'aménagement des ports maritimes ainsi que les terminaux portuaires. Au sein de la région du Nord-Ouest de l'Europe (NWE), peu de ports possèdent la capacité de gérer cette croissance, malgré l'importance de cette zone côtière qui s'étend de l'Irlande aux Pays-Bas.

Le problème majeur lié au développement des terminaux portuaires de la région NWE dépend de la gestion interne du trafic et de l'optimisation au sein de leurs espaces confinés. Une solution a été proposée pour certains ports importants, tels que les ports de Rotterdam, Düsseldorf et Hambourg, pour automatiser la manutention de marchandise en conteneurs en utilisant un système de Véhicules Guidés Automatisés² (AGV). Cette solution a permis de résoudre certains problèmes relatifs au trafic interne. En revanche, elle a permis de souligner un certain nombre de limites :

- L'infrastructure portuaire doit s'adapter à l'utilisation du système AGV, ce qui rend difficile toute tentative de généralisation sur d'autres ports européens.
- Le système AGV est alimenté par un moteur à combustion interne, ce qui augmente les niveaux de pollution dans l'environnement immédiat des espaces d'activité portuaires.
- Les échecs de gestion d'un tel système de transport ne sont pas pris en compte. Par exemple, lorsqu'un véhicule est en état de panne/faute, il bloque le parcours des véhicules suivants.

Le projet InTraDE est un projet européen qui a pour but d'étudier et d'améliorer le transport des conteneurs au sein des ports maritimes de petites tailles. Il a débuté en 2010 pour une période de trois ans et demi, et il regroupe les partenaires académiques européens suivants : le Laboratoire d'Automatique Génie Informatique et Signal (LAGIS) à Lille, l'Institut National de Recherche en Automatique et en Informatique (INRIA) à Nancy, le Centre Régional d'Innovation et de Transfert de Technologie dans les transports et la logistique (CRITT) au Havre, le laboratoire LOOM de l'université de Liverpool John Moores, et l'Institut de Technologie de Dublin (DIT).

Il compte aussi des partenaires industriels tels que le port d'Oostende en Belgique, lequel fournit un environnement de test réel, l'entreprise RoboSoft, qui est en charge de la production du prototype de véhicule intelligent autonome illustré par la figure 1.1, et l'entreprise Oktal, qui est en charge de fournir un logiciel de simulation en réalité virtuelle des véhicules.

L'objectif est d'améliorer la gestion du trafic ainsi que l'optimisation des espaces dans les zones confinées en développant un système de transport intelligent et écologique offrant une meilleure sécurité.

Ce système doit être capable de s'adapter aux exigences spécifiques de l'environnement, et pourrait être transféré à des ports et terminaux quelles que soient leurs dimensions.

Le système de transport suggéré fonctionne en conjonction avec un logiciel de simulation, permettant ainsi (1) une supervision robuste (en temps réel) de l'opération de manutention des conteneurs et (2) l'utilisation de l'infrastructure existante des ports sans aucun investissement supplémentaire.

2. est un robot qui se déplace sans l'intervention humaine, en utilisant le filoguidage, le laserguidage et l'optoguidage [AGV].

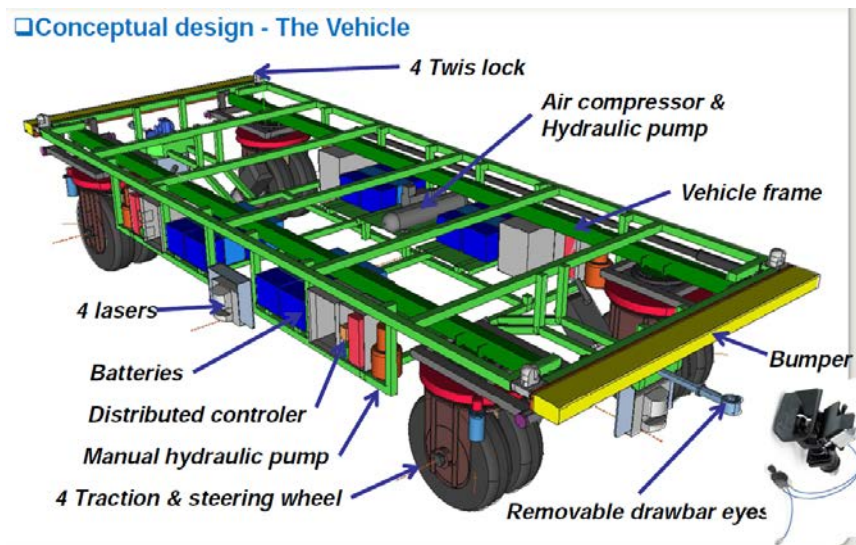


FIGURE 1.1 – Véhicule intelligent autonome InTraDE

1.2 Objectifs de la thèse

Dans le contexte d'InTraDE, la thèse se focalise sur le problème de la gestion et de la régulation du trafic des véhicules autonomes transportant des conteneurs entre bateaux et dépôts.

Nous supposons qu'un superviseur planifie en-ligne les trajets de nombreux véhicules dans une zone limitée (en espace). Ces véhicules, ou convois de véhicules, peuvent être autonomes ou conduits par un pilote. Dans un tel système complexe de grande taille, c'est à dire avec de nombreux véhicules (agents), de nombreux événements inattendus peuvent avoir lieu : une panne d'un véhicule, une erreur humaine pendant la conduite, un obstacle sur une route, des replanifications locales, etc. Dans un système de transport, de tels événements peuvent générer des blocages locaux mais aussi des phénomènes globaux tels que des embouteillages, des ralentissements, voire même des blocages du système.

De plus, les systèmes qui obligent les véhicules à s'arrêter fréquemment, tels que les signalisations classiques pour la gestion du trafic, engendrent un retard considérable. Nous pouvons citer pour exemple les feux de circulation ou les panneaux "STOP". Par conséquent, tout gain de temps ou tout évitement de congestion sont recherchés.

Enfin, les systèmes de transport sont connus pour leur complexité, essentiellement due au caractère dynamique du trafic, lequel croît en fonction du débit des véhicules. Le traitement de ces problèmes nécessite donc des approches **rapides et efficaces**.

Ainsi, dans cette thèse, nous considérons les techniques de coordination réactive et d'optimisation comme un moyen de gestion des conflits et des congestions du trafic dans un système de transport semi-planifié de grande taille afin d'améliorer la circulation et de réduire les risques d'embouteillages.

1.3 Approche suivie dans la thèse

La nature distribuée de notre problème, à travers *les agents* (véhicules autonomes) qui le composent, nous amène à le considérer comme un système multi-agent. Nous cherchons ici à éviter les conflits entre véhicules, c'est-à-dire tout ralentissement du système, en explorant des approches fondées sur des règles de coopération entre agents. Il s'agit de prévenir et gérer les conflits au niveau local, c'est-à-dire lorsqu'ils apparaissent entre les agents, afin de les traiter rapidement (en temps réel) sans avoir à replanifier les tâches des agents (véhicules) impliqués. En effet, la re-planification [Nebel et Koehler, 1995] n'est pas adaptée aux systèmes multi-agents de grandes tailles, du fait de l'explosion combinatoire que cela implique.

Par conséquent, l'approche que nous adoptons dans cette thèse repose sur la recherche de comportements de coordination locale permettant de fluidifier et d'assurer le bon fonctionnement du système de transport, c'est-à-dire des véhicules autonomes. Dans cette thèse, nous utilisons le principe de la coordination d'action au sens de la définition de Ferber [1995] :

“Articulation des actions individuelles accomplies par chacun des agents de manière à ce que l'ensemble aboutisse à un tout cohérent et performant.”

La coordination réactive

Le besoin de rapidité et d'efficacité pour traiter des problèmes de trafic, ainsi que la capacité limitée de perception des véhicules autonomes, nous orientent naturellement vers des comportements réactifs coopératifs.

La coordination réactive est obtenue à partir d'interactions simples entre agents voisins, reposant sur des réactions à leurs perceptions locales. De telles hypothèses permettent aux agents de réagir en temps réel aux situations conflictuelles. Parmi les modèles connus de coordination réactive nous pouvons citer [Reynolds, 1987] pour la navigation collective (flocking), [Zeghal, 1998] pour l'évitement entre robots mobiles ou avions et [Simonin et Ferber, 2000; Simonin, 2001] pour la résolution de conflits de navigation multi-robots. Notons que cette approche permet des communications directes entre agents voisins ou via le marquage de l'environnement [Simonin, 2010].

Toutefois ces règles réactives ne garantissent pas l'optimalité des solutions, justement à cause de la localité des traitements et des limites des architectures réactives.

Optimisation

Les limites des systèmes réactifs (localité de traitement) et la complexité de la régulation du trafic nous amènent à considérer les problèmes abordés dans cette thèse comme des problèmes d'optimisation, et à définir systématiquement des critères d'optimisation.

En outre, les critères d'optimisation permettent, d'une part, d'évaluer les solutions proposées expérimentalement (mesures de performance), et d'autre part, de comparer nos solutions avec d'autres approches de la littérature.

Le couplage des techniques de coordination réactive et d'optimisation constitue une démarche originale dans le cadre de la régulation du trafic de véhicules autonomes. Elle permet de ne pas se limiter à des mécanismes purement réactifs mais à intégrer des processus d'optimisation tout en gardant l'hypothèse de localité.

Synthèse de notre approche

Dans cette thèse, nous cherchons à développer des traitements locaux avec des communications limitées pour améliorer le trafic des véhicules autonomes. Ces traitements sont obtenus en intégrant une optimisation locale (spatialement) aux systèmes multi-agents réactifs. Cette démarche nous conduit à distinguer deux types d'optimisation en fonction de la nature du problème traité : (i) l'optimisation de la prise de décision des véhicules, et (ii) l'optimisation des paramètres du trafic des véhicules autonomes. Afin d'observer et de comprendre le fonctionnement des algorithmes proposés dans un tel système complexe, nous développons des scénarios de trafic et évaluons empiriquement les performances des algorithmes selon différents critères d'optimisation. Enfin, notre démarche est d'étudier des problèmes de complexité croissante tout au long la thèse. Ils correspondent à trois niveaux différents d'interaction : la coordination locale entre véhicules voisins, le croisement de véhicules au niveau d'une intersection, et la régulation d'intersections dans un réseau.

1.4 Plan de la thèse

Le présent mémoire s'articule en deux grandes parties. La première partie s'intéresse au problème du partage d'une voie par deux flux opposés de véhicules autonomes. Puis, dans la deuxième partie, nous étendons notre étude à des cas plus génériques et complexes de gestion d'intersections dans un réseau de transport.

Partage d'une voie par deux flux de véhicules

Dans la première partie de cette thèse, nous commençons par étudier un sous-problème du trafic. Nous considérons une route à double sens où, soudainement, une des voies est bloquée, par exemple par un véhicule tombé en panne, obligeant alors les véhicules des deux sens à partager la seule voie libre. Cette situation est posée comme le problème générique du partage d'un espace commun entre deux files d'agents mobiles de tailles finies ou infinies. Notons que ce problème de partage d'une route entre deux files de véhicules est équivalent à celui de la gestion d'un carrefour, mais sans feux tricolores. Ces problèmes de gestion des intersections sont traditionnellement étudiés en recherche opérationnelle et en théorie des files d'attente.

Dans le chapitre 2, nous présentons une étude bibliographique sur les travaux proposant des comportements permettant d'assurer le trafic dans ce type de situation. Cette étude montre qu'il n'y a pas beaucoup de travaux (surtout dans le cadre de l'intelligence artificielle) proposant des comportements de véhicules permettant d'assurer le trafic, sur une même route, de files de véhicules opposées.

Dans le chapitre 3, nous proposons deux approches reposant sur des règles de coordination simples, et qui requièrent une communication limitée entre les véhicules en tête des deux files. L'avantage de ces approches est la décentralisation complète du modèle car il n'existe pas de mécanisme centralisé qui gère l'ensemble des véhicules. Nous cherchons, ici, à assurer le trafic simultané des deux files par le partage d'une voie, tout en minimisant le retard des véhicules par rapport à leur plan initial.

Pour décrire le comportement des agents de ces stratégies réactives, nous proposons un modèle de l'environnement et un modèle d'agents. Nous avons implémenté ces stratégies dans un simulateur permettant leur analyse et leur comparaison. Différents scénarios de trafic sont analysés et l'étude expérimentale montre la capacité de l'une des stratégies à réduire les retards.

Optimisation de la circulation dans un réseau de transport

Dans les systèmes de transport réels, les congestions sont générées aux intersections des routes, c'est à dire dans les parties de l'espace qui doivent être partagées par les véhicules. La plupart des approches proposées forcent l'arrêt des véhicules à ces intersections.

Dans la deuxième partie de cette thèse, nous explorons une nouvelle façon d'aborder ce problème. Nous adoptons une démarche inverse à l'existant en visant le passage des véhicules aux intersections sans les arrêter.

Dans le chapitre 4, nous présentons une étude bibliographique sur les systèmes de gestion d'intersections dans les réseaux de transport. Nous mettons en avant les grandes dimensions selon lesquelles ces systèmes se distinguent en insistant sur les différents types d'algorithmes utilisés.

Dans le chapitre 5, nous proposons une modélisation du problème et un outil de simulation que nous utilisons dans le cadre de cette deuxième partie pour analyser les approches développées.

Dans le chapitre 6, nous proposons une approche originale pour la gestion d'une intersection, en permettant aux flux de véhicules de passer sans s'arrêter. L'idée développée consiste à faire passer alternativement, et sans interruption, les véhicules de chaque route avec une inter-distance suffisante de manière à éviter les collisions.

Plus précisément, nous définissons une approche introduisant un agent contrôleur à chaque intersection qui synchronise le passage des flux de véhicules entrants à une période pré-calculée. Cette durée dépend de la géométrie de l'intersection (dont l'angle formé par les deux routes). Le rôle de cet agent contrôleur est de modifier la vitesse des véhicules, à l'entrée de chaque intersection, pour leur permettre de se croiser.

Dans le chapitre 7, nous étudions la façon d’optimiser le trafic globalement, c’est-à-dire au niveau d’un réseau d’intersections. Une approche commune consiste à utiliser les feux de signalisation classiques synchronisés, à savoir, des feux de circulation qui partagent tous la même longueur de cycle (temps nécessaire pour un cycle de feux de circulation), et dont les déphasages (phases de ces signaux périodiques) sont choisis de manière à favoriser les ondes vertes. Ils permettent à la plupart des véhicules d’éviter l’arrêt s’ils respectent une certaine vitesse. Le problème de ce type d’approche est qu’elle n’est pas adaptative, ni conçue pour des véhicules autonomes.

Nous développons ici une nouvelle approche pour les véhicules autonomes et nous l’évaluons empiriquement en tenant compte de divers paramètres entrant en jeu, tels que le débit de véhicules ou le rayon d’action des agents contrôleurs. Les résultats montrent que notre approche, dans laquelle les véhicules n’ont pas besoin de s’arrêter aux intersections, améliore considérablement la circulation en utilisant simplement des intersections indépendantes.

Enfin, nous dressons dans le dernier chapitre un bilan des travaux présentés et leurs perspectives.

Première partie

Partage d'une voie par deux files de véhicules autonomes

Introduction

Dans cette première partie de la thèse, nous nous attaquons au problème général du partage d'un même espace par un ensemble de véhicules autonomes, tels que ceux envisagés dans des environnements de transport automatiques (plate-forme portuaire³, etc.).

Dans de tels systèmes les véhicules reçoivent des plans de route à suivre pour transporter des marchandises. Ces systèmes sont ainsi très sensibles aux retards/conflicts locaux car ils peuvent impacter tous les véhicules ayant une route passant par ce conflit localisé. Ainsi, nous avons considéré comme cas d'étude une route à double sens où, soudainement, une des voies est bloquée, par exemple par un véhicule tombé en panne, obligeant alors les véhicules des deux sens à partager la seule voie libre. Cette situation généralise le problème du partage d'un espace commun entre deux files d'agents mobiles de tailles finies ou infinies.

Dans le premier chapitre de cette thèse, nous présentons une étude bibliographique généraliste sur le problème du partage d'un espace commun entre deux files de véhicules autonomes. Nous notons que ce problème n'a pas été beaucoup étudié dans la littérature.

Dans le deuxième chapitre, nous proposons deux approches reposant sur des règles de coordination simples, lesquelles requièrent une communication limitée entre les véhicules de tête des deux files. Nous visons, ici, à assurer le trafic simultané des deux files par le partage d'une voie, tout en minimisant le retard des véhicules par rapport à leur plan initial.

3. <http://www.intrade-nwe.eu/>

Chapitre 2

État de l'art sur le problème du partage d'une voie

Sommaire

2.1	Scénario du partage d'une voie	14
2.2	Modélisation sans contrôle du trafic	15
2.2.1	Modélisation en général	15
2.2.1.1	Les modèles macroscopiques	15
2.2.1.2	Les modèles microscopiques	15
2.2.2	Modélisation du trafic interrompu par un accident	16
2.2.3	Analyse et estimation du retard	17
2.3	Modélisation et contrôle du trafic	19
2.3.1	Modélisation pour la simulation	19
2.3.2	Contrôleur pour minimiser les retards	21
2.4	Bilan	21

Ce premier chapitre commence, en section 2.1, par une présentation détaillée du problème de partage d'une voie. Suivra ensuite en section 2.2 une première partie bibliographique, qui décrit les différentes approches pour la modélisation du trafic. Enfin, avant de conclure en section 2.4, nous consacrons la section 2.3 à quelques travaux pertinents sur le contrôle des véhicules, et qu'il nous semble important de citer.

2.1 Scénario du partage d'une voie

Nous nous concentrons dans cette étude sur des véhicules autonomes ayant pour mission de transporter des ressources d'une origine à une destination. Comme illustré sur la figure 2.1, nous considérons une route à deux sens, où le trafic a été interrompu par un obstacle au temps $t = 0$ sur l'une des voies.

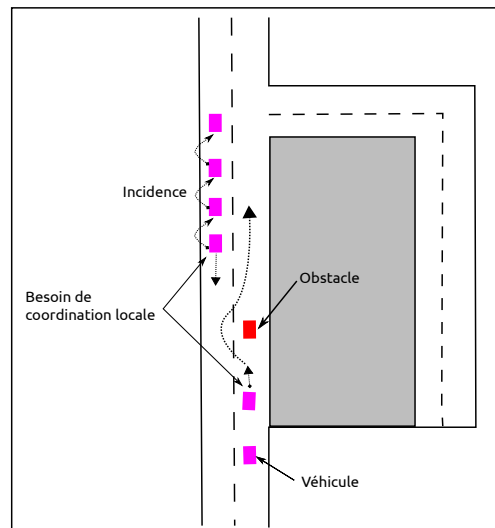


FIGURE 2.1 – Deux flux de véhicules bloqués par un obstacle

Ce problème de partage d'une route entre deux files de véhicules est équivalent à celui de la gestion d'un carrefour, mais sans feux tricolores. Dans ce cas, si le trafic est fluide, la solution la plus simple consiste à faire passer les premiers véhicules arrivés au niveau de l'obstacle (premier arrivé premier servi). De cette manière, on réduit la taille des deux files d'attentes en fonction des arrivées des véhicules. Cependant, cette solution n'est plus optimale si le trafic est dense ou si on a des véhicules plus prioritaires que d'autres.

Ces problèmes de gestion des intersections sont traditionnellement étudiés en recherche opérationnelle et en théorie des files d'attente. A notre connaissance, il n'y a pas de travaux proposant des comportements pour des véhicules autonomes permettant d'assurer le trafic, sur une même route, de files de véhicules opposées. Par contre, des problématiques proches ont été étudiées à d'autres fins telles que l'estimation des retards des véhicules [Tanner, 1953], la modélisation du flux du trafic [Baykal-Gürsoy *et al.*, 2009b], ou la résolution de conflits spatiaux entre robots autonomes [Gaillard *et al.*, 2011, 2012].

Dans les sections suivantes, nous présentons une étude bibliographique de ce problème de partage d'une voie. Nous avons décidé de classer les travaux existants dans la littérature en deux grandes classes distinctes : d'une part, les travaux qui s'intéressent uniquement à la modélisation du trafic et ne proposant pas des solutions de contrôle du trafic, et d'autre part, les travaux qui amènent à considérer des problèmes plus ou moins similaires au nôtre comme un problème de contrôle de véhicules.

2.2 Modélisation sans contrôle du trafic

Les premières études du trafic urbain remontent au milieu du 20^{ème} siècle avec l'apparition des feux de circulation. Depuis, plusieurs travaux ont permis de dégager des lois de comportement, en particulier pour la modélisation et la simulation du trafic.

2.2.1 Modélisation en général

Comme dans la littérature, nous considérons les modèles de trafic selon le niveau de détail qu'ils proposent. Dans l'histoire des modèles de trafic, il existe deux grandes classes : les modèles macroscopiques (niveau agrégé du trafic) et les modèles microscopiques (niveau des véhicules) [Mammar, 2007]. Le domaine de la modélisation du trafic est un domaine très vaste et il existe aussi un troisième type de modèles, appelé le modèle mésoscopique, et qui se situe entre le modèle macroscopique et le modèle microscopique, en modélisant une partie du système macroscopiquement et l'autre partie microscopiquement. Nous nous contentons ici de ces deux derniers qui suffisent à notre étude.

2.2.1.1 Les modèles macroscopiques

Les modèles macroscopiques considèrent le trafic comme un flux dans un graphe. Ce sont les premiers modèles à voir le jour et ce sont les plus anciens [Greenshields, 1935]. Au début, les moyens de calculs –essentiellement humains– étaient insuffisants pour pouvoir modéliser le trafic avec précision. La démarche consistait à mesurer la densité et le débit du trafic par l'intermédiaire de caméras ou de rampes de passage. Puis, à l'aide de modèles analytiques, fondés par exemple sur une analogie avec la mécanique des fluides, il était possible de décrire le comportement du trafic avec les lois de circulation des fluides (les véhicules sur les routes sont considérés comme le fluide circulant dans les canalisations) [Greenshields, 1935; Lighthill et Whitham, 1955; Richards, 1956]. Ces modèles macroscopiques ne décrivent pas les comportements individuels, mais offrent un modèle de haut niveau (agrégé) du trafic.

2.2.1.2 Les modèles microscopiques

L'échelle microscopique se place au niveau des véhicules. On dit que se sont des modèles individu-centrés. Ils décrivent le mouvement de chaque véhicule et leurs interactions [Si-

mon et Nagel, 1998; Kosonen et Pursula, 1990; Meignan *et al.*, 2007]. Ces modèles sont fondés sur de nouvelles lois propres au trafic telles que les lois de poursuite (un conducteur suit toujours la voiture qui le précède) ou les lois de changement de voie (lors d'un dépassement sur une autoroute par exemple). Certains de ces travaux se situent dans le cadre des systèmes multi-agents.

Ces modèles sont par nature très fins et peuvent être coûteux à implémenter et à exécuter. En effet, ils nécessitent de traiter de nombreuses données (par exemple les paramètres de chaque véhicule tels que la vitesse, la direction, l'accélération, etc...), ce qui en fait leur principale limite. En plus, le comportement humain est imprévisible, ce qui amène beaucoup d'auteurs, tels que [Hancock, 1999], à critiquer les modèles microscopiques. En effet, ils n'ont pas la même exactitude de description du trafic (l'évolution des flux dans les réseaux) que les modèles macroscopiques. Malgré cela, la plupart des recherches scientifiques des quinze dernières années délaissent les modèles macroscopiques et tendent à étudier les modèles microscopiques. Ceci est expliqué par la puissance de calcul des nouveaux ordinateurs qui augmente exponentiellement d'année en année.

2.2.2 Modélisation du trafic interrompu par un accident

Il existe dans la littérature quelques travaux qui se sont intéressés à la modélisation du trafic interrompu par des événements irréguliers tels que les accidents de la circulation, les véhicules en panne, les conditions météorologiques défavorables ou la présence de matières dangereuses. Cependant, la plupart de ces travaux ne se focalisent que sur la modélisation du comportement des véhicules et essaient de valider les modèles développés en les comparant à des données collectées dans la réalité sans donner de solutions pour améliorer le trafic (contrôle). Les travaux réalisés sont généralement inspirés de la théorie des files d'attente, par exemple pour modéliser le scénario décrit dans la section 2.1. La route qui va faire passer les deux files d'attente est modélisée comme un serveur qui ne traite qu'une seule tâche à la fois et les véhicules présents sur les deux routes sont représentés par des tâches en attente d'exécution.

La théorie des files d'attente est un domaine très vaste. Nous rappelons ici quelques principes de base pour aider le lecteur à bien suivre notre exposé.

Rappel : Les systèmes de files d'attente sont décrits selon la notation de Kendall [Kendall, 1953; Alec Miller, 1966] par 6 symboles $a/s/C/K/m/Z$:⁴

- a : indique la loi de probabilité des instants d'arrivées, par exemple GI pour la loi générale indépendante et M pour la loi exponentielle ;
- s : indique la loi de probabilité de la durée du service (au guichet) ; on utilise les mêmes symboles que précédemment ;
- C : indique le nombre de serveurs (nombre de guichets) ;
- K : c'est la capacité totale du système, c'est-à-dire le nombre de serveurs (C) + le nombre de places en attente ;
- m : indique la population totale de clients (par exemple : nombre d'inscrits sur une liste électorale dans le cas d'une file d'attente à un bureau de vote) ;

4. http://fr.wikipedia.org/wiki/Notation_de_Kendall

- Z : indique la politique de service, par exemple *first in, first out* (FIFO, alias *PAPS*, premier arrivé, premier servi).

Très souvent, les trois derniers symboles de la notation sont omis avec, par défaut, K infini, m infini et Z en *PAPS*.

A titre d'exemple, [Baykal-Gürsoy et al. \[2009a,b\]](#) ont présenté un modèle de files d'attente pour décrire le flux du trafic sur une route sujette à des accidents comme le montre la figure 2.2.

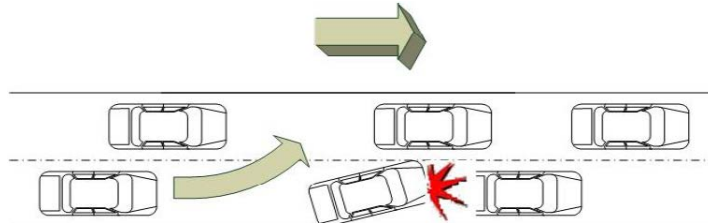


FIGURE 2.2 – Trafic interrompu par un accident [[Baykal-Gürsoy et al., 2009a](#)]

Ils considèrent une route avec plusieurs voies où toutes les voies vont dans le même sens et la modélisent comme étant un système de files d'attente $M/M/c$ à l'état d'équilibre, où c est le nombre de voies. Ils supposent qu'elles ont une capacité infinie de stockage des véhicules, lesquels arrivent suivant une distribution de Poisson sur les routes. Quand un incident se produit (suivant aussi une distribution de Poisson), tout ou une partie des voies peuvent être fermées à la circulation. Le trafic se détériore de telle sorte que le nombre de voies et le taux de service de toutes les voies diminuent. L'impact négatif de l'incident implique à la fois la congestion et la réduction de la capacité routière.

Ils analysent ce système à l'état d'équilibre et proposent un modèle de files d'attente ainsi qu'une solution visant à estimer le nombre stationnaire de véhicules sur les voies. Cependant, l'un des problèmes constaté par cette modélisation est que la complexité du modèle augmente exponentiellement quand le nombre de voies augmente. Ils proposent alors une solution pour approximer leur modèle $M/M/c$ et présentent des résultats de simulation qui montrent la validité de leur modèle de files d'attente par rapport à des données réelles en fonction des durées moyennes de traversée.

2.2.3 Analyse et estimation du retard

Après avoir discuté des différents modèles de trafic urbain, nous étudions, dans cette partie, les travaux qui s'attaquent aux problèmes du retard des véhicules. Plusieurs contributions existent pour évaluer les retards dans les réseaux routiers, mais nous préférons présenter, ici, quelques travaux qui s'intéressent à l'analyse et l'estimation des retards sur les routes interrompues par des incidents (notre scénario).

Une autre approche proposée par [Baykal-Gürsoy et al. \[2006\]](#) (appliquée sur le même modèle présenté dans la figure 2.2) s'appuie sur la théorie des files d'attente pour estimer le retard du trafic causé par les incidents. Ils présentent un modèle analytique pour approximer le temps moyen de traversée sur chaque voie, et peuvent ainsi estimer le retard

causé par la fermeture d'une voie. Pour valider leur modèle, ils le comparent avec des résultats obtenus en simulation microscopique.

Un problème très proche du nôtre a été traité par le passé par [Tanner \[1953\]](#). C'est le seul article à notre connaissance qui considère notre cas d'étude. Il définit un modèle théorique mathématique des retards estimés lorsque deux files de véhicules opposées tentent de passer simultanément par la même voie. Tanner considère une route AB avec deux voies. Une partie au milieu de cette route (une arête) est assez large pour faire passer un véhicule comme le montre la figure 2.3. Les véhicules arrivent avec une distribution stochastique des deux côtés de la route.

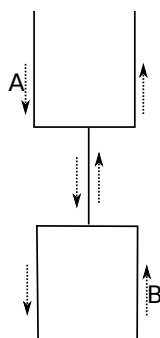


FIGURE 2.3 – Problème d'interférence entre deux files

Dans ce modèle, tous les véhicules se déplacent à la même vitesse constante, et leurs temps de démarrage et d'arrêt sont considérés comme négligeables. Les véhicules V_a et V_b sont les véhicules qui circulent respectivement sur les voies A et B . Un véhicule V_a ne peut s'engager sur l'arête que (1) s'il n'y a pas un véhicule V_b sur l'arête (dans le sens opposé), et (2) s'il n'y a pas eu un autre véhicule V_a qui est passé pendant une période β_1 . De même pour V_b , il ne peut s'engager sur l'arête que (1) s'il n'y a pas un véhicule V_a sur l'arête (dans le sens opposé), et (2) s'il n'y a pas eu un autre véhicule V_b qui est passé pendant une période β_2 . Pour traverser l'arête, il faut pour V_a et V_b respectivement une durée de α_1 et α_2 .

Tanner étudie, dans ce travail, les retards moyens pour les véhicules en provenance de la route A et en provenance de la route B lorsqu'une solution stationnaire existe, c'est-à-dire quand la vitesse à laquelle les véhicules passent à travers l'arête est supérieure à la vitesse à laquelle ils arrivent, de sorte que les files d'attente des véhicules ne se remplissent pas. Il distingue trois cas d'étude :

- (i) quand $\alpha_1 > \beta_1$, $\alpha_2 > \beta_2$;
- (ii) quand $\alpha_1 > \beta_1$, $\alpha_2 < \beta_2$ (ou $\alpha_1 < \beta_1$, $\alpha_2 > \beta_2$) ;
- (iii) et quand $\alpha_1 < \beta_1$, $\alpha_2 < \beta_2$.

Il estime que les cas (ii) et (iii) sont complexes à étudier et présentent des difficultés mathématiques. A la fin de l'article, Tanner présente l'expression analytique pour l'estimation des retards des deux flux A et B en fonction du débit d'injection, de la périodicité (β_1 et β_2) et des temps de traversée (α_1 et α_2).

Cependant, ce travail de modélisation permet d'évaluer les retards, ce qui diffère de notre objectif principal qui vise à définir des comportements permettant de réduire efficacement ces retards.

On notera que dans le même contexte, [Tanner \[1962\]](#) s'est intéressé aussi à étudier le retard moyen des véhicules sur une route secondaire d'un carrefour où la circulation sur la route principale a la priorité absolue. Ce travail présente aussi un modèle mathématique théorique des retards estimés.

2.3 Modélisation et contrôle du trafic

Dans cette section, nous présentons les travaux qui s'intéressent au contrôle du trafic urbain dans le contexte d'une interruption durable d'une voie. A notre connaissance, il n'y a pas de travaux qui soient réellement proches de notre scénario. Nous présentons ici les travaux qui nous semblent les plus pertinents et qui se rapprochent le plus des modèles de trafics interrompus par des incidents.

2.3.1 Modélisation pour la simulation

Différentes approches ont été proposées pour modéliser et analyser l'évolution du trafic lorsqu'il est interrompu par un incident et en particulier pour simuler le comportement du trafic (et des conducteurs). L'objectif est de pouvoir étudier et évaluer les performances des systèmes de réseau de transport au niveau opérationnel.

[Hidas \[2002\]](#) a proposé une simulation microscopique multi-agent d'un réseau de transport et présenté des modèles de changements de voie (voulus, forcés et coopératifs) pour éviter les accidents. Les modèles en question concernent les flux de véhicules qui sont sur une route composée de plusieurs voies comme illustré dans la figure 2.4.

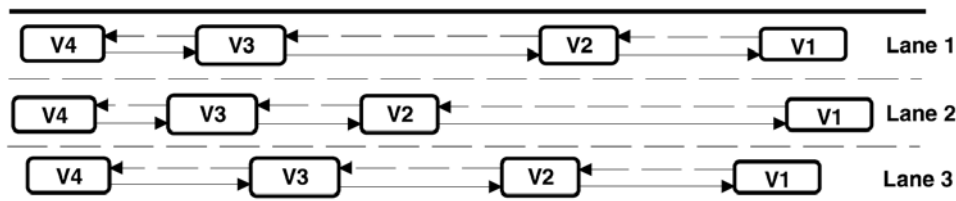


FIGURE 2.4 – Route composée de plusieurs voies [[Hidas, 2002](#)]

A un instant donné, une ou plusieurs voies peuvent être fermées à la circulation. L'objectif de ce travail est d'utiliser la simulation comme un moyen d'analyse des incidents de trafic, comme les accidents ou les pannes sur les routes. Pour cela, Hidas étudie le comportement humain concernant les changements de voie sans interférence avec les voies de destination, et définit ainsi un certain nombre de règles qui régissent ce comportement. Il commence par évaluer si le changement de voie est essentiel, désirable ou pas

nécessaire selon des critères choisis à l'avance (tels que la vitesse, la présence d'un incident sur la route, etc). Ensuite, il cherche une voie de destination appropriée tout en étudiant la faisabilité du changement de voie en fonction des paramètres du véhicule (tels que l'accélération et la décélération). Et enfin, il change de voie. Il implémente son modèle microscopique dans un simulateur (SITRAS), et ses résultats montrent que seuls les comportements forcés et coopératifs reproduisent de façon réaliste les interactions entre flux de véhicules aux situations de congestion.

Dans le même contexte, [Sheu \[2013\]](#); [Chevallier et Leclercq \[2009\]](#) présentent aussi des modèles microscopiques, avec moins de paramètres, pour les voies de circulation afin de caractériser le comportement des conducteurs induit par l'arrivée d'un incident. Dans ces modèles, ils introduisent des règles moins strictes qui régissent le contrôle longitudinal des véhicules suiveurs et proposent de nouveaux algorithmes de décision d'insertion. Ces règles sont généralement exprimées comme un problème de choix binaire à un seul niveau : le résultat du processus de décision d'insertion est évalué qu'une seule fois par pas de temps, indépendamment de ce qui s'est passé ou va se passer après l'insertion. Dans ces travaux, les résultats obtenus en simulation sont comparés à des vidéos réelles de circulations en situation de congestion afin d'évaluer les modèles. Ces approches comparées montrent qu'elles répartissent avec précision les débits observés dans des conditions de congestion.

[Bonte et al. \[2006, 2007\]](#) introduisent un modèle permettant la simulation des utilisateurs de deux-roues motorisés. L'objectif de ce travail est de mieux comprendre les comportements de ces conducteurs considérés comme étant vulnérables suite à leur manque de protection et de visibilité. Pour remédier à ce problème, les auteurs proposent un concept de voies virtuelles ; un espace, structuré, dynamique et privé, dédié aux usagers des deux roues motorisés. Ils combinent leur approche dans un modèle de trafic existant du simulateur ARCHISIM [\[Espié, 1995\]](#) et comparent les résultats obtenus avec des données réelles du trafic.

[Lacroix et Mathieu \[2012\]](#); [Lacroix et al. \[2013\]](#) proposent un formalisme de génération d'une population consistante et variée d'agents (véhicules) dans le simulateur SCANNER⁵. L'objectif est de générer des simulations multi-agents réalistes. Leur approche est fondée sur un modèle de différenciation des comportements en utilisant des normes [\[Lacroix et al., 2008\]](#) et un modèle de construction automatique de comportements par apprentissage (réseaux de Kohonen) à partir de données réelles. Les résultats présentés par les auteurs montrent que leur approche augmente la variété des agents et produit des comportements représentatifs.

Nous remarquons, dans cette section, que les outils de simulation (environnements de test) sont très importants pour la validation des modèles proposés. Pour répondre à ce besoin, quelques travaux proposent des simulateurs expérimentaux de trafic microscopique. Par exemple, [\[Bonhomme et al., 2014\]](#) présentent leur simulateur TrafficGen comme un outil permettant aujourd'hui de simuler des flux de véhicules avec des vitesses différentes, et donc différentes techniques de dépassement et de changement de voie.

5. <http://www.oktal.fr/fr/automobile/gammes-de-simulateurs/produits-logiciels-scane-r>

2.3.2 Contrôleur pour minimiser les retards

En théorie des files d'attente, la première solution proposée pour le contrôle des retards dans les systèmes $M/M/1$ fut celle de Cohen [1987]. Le modèle de file d'attente étudié se compose d'un serveur et deux files d'attente. Chaque file d'attente a sa propre distribution du temps de service qui suit une loi de Poisson. Après l'achèvement de chaque service, le serveur procède au traitement de la tâche qui appartient à la file d'attente la plus longue. Si les deux files d'attente ont le même nombre de tâches, le serveur choisit entre elles avec une certaine probabilité. Son modèle a un intérêt pratique et théorique, car l'équation fonctionnelle à résoudre déterminée par Cohen n'a pas encore été étudiée dans la littérature des files d'attente. Il montre que ce problème peut être réduit à un problème de limitation de valeur de Riemann–Hilbert (une classe de problème pour l'étude des équations différentielles dans les plans complexes [Clancey et Gohberg, 1981]). Il présente quelques résultats numériques, et constate que la stratégie favorisant la file d'attente la plus longue a une influence importante sur la probabilité d'équité entre les deux files d'attente.

Ahadar *et al.* [2003] présentent pour leur part un contrôleur (une commande Bang-Bang) inspiré de la théorie des files d'attente. L'objectif de cette commande est de minimiser l'attente des véhicules à une intersection isolée. Dans le problème traité, ils ne considèrent qu'une intersection à deux phases (uniquement deux voies entrantes) sans l'influence des intersections adjacentes. A partir des relations qui gèrent le retard pendant la saturation d'un carrefour, ils déterminent un modèle discret de files d'attente en fonction des cycles. Ensuite, ils définissent des critères d'optimisation et ils appliquent le principe du Maximum pour obtenir une stratégie de commande Bang-Bang. L'objectif de ce travail est de se rapprocher le plus fidèlement possible d'un fonctionnement réel d'un carrefour. Ils étudient la formulation qui prend en compte les contraintes aussi bien sur la commande que sur l'état. Ils appliquent les principes de la commande optimale en temps réel car le problème (formalisé) à optimiser est linéaire. Leurs résultats confirment la symétrie existante entre les deux phases de l'intersection. En effet, ils montrent que la commande a un caractère sinusoïdal inversé des deux flux de véhicules (quand un flux sature l'autre devient fluide, et vice versa) et illustre l'absence de blocage en fonction des débits d'injection.

2.4 Bilan

Nous avons commencé, dans ce chapitre, par présenter le scénario du partage d'une voie que nous souhaitons étudier. Ensuite, nous avons présenté plusieurs travaux qui ont une relation proche ou éloignée avec notre scénario de travail.

Nous constatons que le problème de partage de voies par deux files d'attente a peu été étudié d'un point de vue contrôle dans la littérature. Les travaux de Tanner sont les plus proches de notre cas d'étude et ne définissent qu'un modèle théorique mathématique des retards estimés pour le passage simultané. Ceci nous a poussé à étudier des solutions à ce problème.

Le chapitre suivant va proposer une nouvelle modélisation du scénario et présenter deux approches fondées sur la coordination réactive pour la résolution de ce problème de partage d'espace.

Chapitre 3

Proposition de solutions par coordination réactive

Sommaire

3.1	Formalisation du problème	24
3.1.1	Modèle du réseau de transport	24
3.1.2	Modèle des agents	25
3.1.3	Critères d'optimisation	27
3.2	Comportements de coordination	29
3.2.1	Approche par alternance	29
3.2.2	Optimisation locale à court terme (OLC)	31
3.3	Résultats expérimentaux	32
3.3.1	Simulation	32
3.3.2	Libération des deux voies	33
3.3.3	Régulation d'un trafic continu	34
3.4	Simulation avec SCANeRstudio	36
3.5	Conclusion	36

Ce chapitre présente deux approches reposant sur des règles de coordination simples, qui requièrent une communication limitée entre les véhicules de tête des deux files. Nous voulons assurer le trafic simultané des deux files par le partage d'une voie, tout en minimisant le retard des véhicules par rapport à leur plan. Pour décrire le comportement des agents de ces stratégies réactives, nous utilisons un formalisme inspiré du langage STRIPS [Fikes et Nilsson, 1971]. Nous implémentons ces stratégies dans un simulateur permettant leur analyse et leur comparaison.

En section 3.1 nous proposons une formalisation du problème et du modèle multi-agent, permettant de représenter les actions possibles des agents et leurs règles de décision (comportements). En section 3.2 nous présentons deux types de règles qui produisent des stratégies différentes. La section 3.3 est consacrée à l'étude expérimentale en simulation des différentes stratégies, en considérant des scénarios déterministes et non-déterministes de l'arrivée des véhicules sur la zone de conflit. Enfin, nous discutons ces résultats et la poursuite de ce travail dans la conclusion.

3.1 Formalisation du problème

Pour la résolution de ce problème de partage de ressource, nous visons à étudier les comportements locaux au niveau de chaque véhicule autonome, pour aboutir à des règles de coordination. Ceci nous amène à considérer des modèles microscopiques.

Dans cette étude, nous discrétisons le temps et l'espace de manière à simplifier la définition de ce modèle :

- l'espace est discrétisé selon une unité de longueur l correspondant au déplacement réalisé en une seconde,
- tous les véhicules ont la même vitesse, et
- l'accélération et la décélération sont instantanés.

3.1.1 Modèle du réseau de transport

Le réseau de transport (les routes) est modélisé ici par un ensemble d'arcs discrets de taille $n \cdot l$. Ces arcs sont connectés par des nœuds. Chaque flux de véhicules suit un chemin particulier dans le réseau. Le trafic est défini par l'ensemble des flux de véhicules.

Le réseau que nous considérons est modélisé par l'ensemble des arcs représentés en figure 3.1.

Deux flux traversent ce réseau. Le premier traverse les arcs A_1, A_2, A_3 et le second B_1, B_2, B_3 . Il contient une section particulière, composée des arcs A_2 (pour les véhicules venant de la source A) et B_2 (pour les véhicules venant de la source B), où les véhicules traversent dans les deux sens. C'est donc cette arête *conflit* qui doit être partagée par les deux flux de véhicules.

Chaque arc est caractérisé par un ensemble de paramètres :

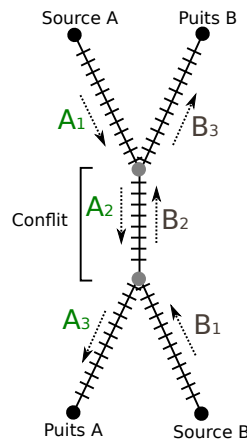


FIGURE 3.1 – Représentation du réseau considéré

- sa longueur en mètres,
- et le nombre de véhicules sur l'arc, lequel s'incrémente et se décrémente quand les véhicules entrent ou sortent.

Chaque arc est aussi divisé en cellules de taille l qui permettent de localiser les agents, comme illustré sur la figure 3.1. Chaque cellule peut être occupée seulement par un véhicule à la fois.

3.1.2 Modèle des agents

L'objectif de cette section est de définir le modèle des agents et de leurs interactions. Un agent/véhicule perçoit des informations sur son environnement et effectue des actions en conséquence. Nous considérons des comportements réactifs car nous souhaitons agir localement et en temps réel, par opposition à un système centralisé. Ainsi chaque agent utilise seulement les perceptions issues de ses capteurs embarqués.

Pour décrire le comportement des agents, nous distinguons le *modèle d'action* des *règles de décision*, comme cela est schématisé sur la figure 3.2. Le *modèle d'action* décrit une liste des actions que l'agent peut réaliser dans son environnement. L'objectif principal de l'agent est de déterminer quelle action choisir afin de satisfaire au mieux ses objectifs. Chaque action est formée de pré-conditions, requises pour son exécution, et d'effets, qui sont les conséquences sur le système. Les *règles de décision* sont dans notre cas les comportements réactifs et de coordination des agents. Elles doivent permettre le déclenchement des actions adéquates.

Chaque agent se déplaçant dans le réseau contient trois variables internes :

1. $T_{objectif}$, la date au delà de laquelle l'agent sera considéré en retard,
2. Arc , l'arc sur lequel se trouve l'agent, et
3. Abs , la position de l'agent sur son arc courant, laquelle s'incrémente au fur et à mesure qu'il se déplace.

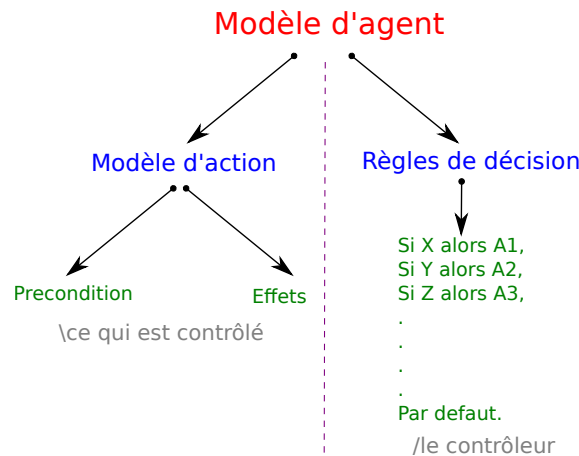


FIGURE 3.2 – Le modèle action-décision

Modèle des actions

Il n'existe pas de représentation normalisée des systèmes multi-agents réactifs. Dans le but de décrire l'état de l'environnement et ses transformations, nous utilisons un formalisme **inspiré** de STRIPS (STanford Research Institute Problem Solver) comme cela est fait par Ferber dans [Ferber, 1999]. STRIPS a été proposé par Fikes et Nilsson pour traiter de problèmes de planification en intelligence artificielle [Fikes et Nilsson, 1971]. C'est un formalisme simple, suffisamment expressif pour décrire de nombreux problèmes et suffisamment restrictif pour être utilisable avec des algorithmes efficaces.

En STRIPS l'état de l'environnement est décrit par un ensemble de clauses composées de littéraux. Leur conjonction est supposée garantir la validité de l'état. Un opérateur STRIPS, qui caractérise une action, est composé :

- d'une précondition qui est un ensemble de prédicats qui doivent être vrais pour que l'action soit exécutable,
- d'une liste de prédicats à supprimer (*delete list*) après l'exécution de l'action, et
- d'une liste de prédicats à ajouter (*add list*) après l'exécution de l'action.

Afin de simplifier l'écriture, nous considérons à la place de la *delete list* et de la *add list*, une post-condition qui regroupe simplement ces deux listes (les prédicats effacés sont précédés par un *non* noté \neg). Ainsi, chaque opérateur est décrit de la façon suivante :

$$\begin{aligned}
 Ope. = \langle & nom : Action()..., \\
 & pre-cond. : A(t), B(t)..., \\
 & post-cond. : C(t + 1), \neg D(t + 1)...\rangle,
 \end{aligned}$$

où :

- t est le pas de temps courant ; et
- $t + 1$ est le prochain pas de temps.

Dans notre cas, les opérateurs manipulent des prédicats à plusieurs variables, comme illustré par les trois actions suivantes :

- *Forward*, qui définit le déplacement d'une unité de distance sur l'arc courant ;

- *DoNothing*, qui consiste à ne rien faire pendant un pas de temps ;
- *ChangeArc*, qui réalise le passage de l'arc courant à l'arc suivant.

La première action, *Forward*, décrit le déplacement :

$$\begin{aligned} \langle name : & \quad Forward(), \\ pre : & \quad \neg Last_t(Abs_t, Arc_t), Free_t(Abs_t + 1, Arc_t), \\ post : & \quad Free_{t+1}(Abs_t, Arc_t), \\ & \quad Abs_{t+1} = Abs_t + 1, \\ & \quad \neg Free_{t+1}(Abs_t + 1, Arc_t) \rangle, \end{aligned}$$

où :

- $Last(Abs, Arc)$ est vrai si et seulement si la position Abs est la dernière position de l'arc Arc ;
- $Free(Abs, Arc)$ est vrai si et seulement si la position Abs de l'arc Arc est libre.

L'action *DoNothing* permet à l'agent de ne pas avancer quand il est en train d'attendre ou quand il est précédé par un véhicule :

$$\begin{aligned} DoNothing = \langle name : & \quad DoNothing(), \\ & \quad pre : \quad , \\ & \quad post : \quad \rangle. \end{aligned}$$

L'action *ChangeArc* décrit le passage de l'agent de son arc courant au suivant :

$$\begin{aligned} \langle name : & \quad ChangeArc(), \\ pre : & \quad Last_t(Abs_t, Arc_t), Free_t(1, NextArc_t), \\ post : & \quad Free_{t+1}(Abs_t, Arc_t), \\ & \quad Arc_{t+1} = NextArc_t, \\ & \quad Abs_{t+1} = 1, \\ & \quad \neg Free_{t+1}(1, NextArc_t) \rangle, \end{aligned}$$

où :

- $NextArc$ indique l'arc suivant de l'agent dans le réseau.

Cet action exprime le fait que, si l'agent veut se déplacer sur l'arc suivant, il doit se trouver en dernière position de son arc courant et la première position de l'arc suivant doit être libre.

3.1.3 Critères d'optimisation

Pour estimer le retard qu'aura un véhicule en sortant du réseau, il faut calculer les temps restants pour atteindre la sortie. Considérons la situation générique où deux véhicules V_1 et V_2 sont présents dans le réseau, comme illustré sur la figure 3.3.

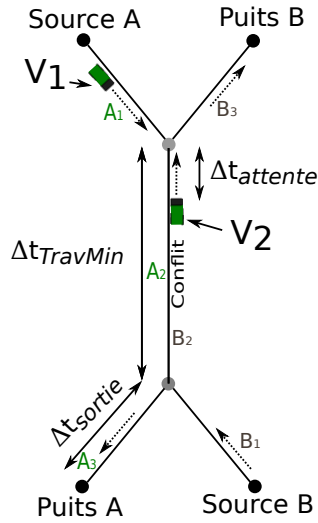


FIGURE 3.3 – Un scénario particulier illustrant les temps à additionner pour estimer la durée du parcours restant pour V_1

V_1 veut entrer sur l'arête conflit. Pour cela, il doit d'abord attendre le passage de V_2 , impliquant un temps d'attente initial ($\Delta t_{attente}$). Quand V_2 quitte l'arête conflit pour passer sur son dernier arc (B_3), V_1 peut passer à son tour (sur A_2). Enfin, pour sortir du réseau, V_1 doit traverser son dernier arc A_3 .

Ici, pour V_1 , la date de sortie du réseau estimée $T_{estimé}$ est la suivante :

$$T_{estimé} = T_{réel} + \Delta t_{attente} + \Delta t_{TravMin} + \Delta t_{sortie}, \quad (3.1)$$

où :

- $T_{réel}$ est la date actuelle ;
- $\Delta t_{attente}$ est le temps requis pour que le véhicule présent à contre-sens quitte l'arête conflit ;
- $\Delta t_{TravMin}$ est le temps nécessaire à la traversée de l'arc conflit ; et
- Δt_{sortie} est le temps requis pour traverser le dernier arc avant de sortir.

Nous appellerons retard d'un véhicule le temps perdu entre la date de sortie du réseau initialement planifiée et la date réelle de sortie. Nous posons ainsi le retard D (pour l'anglais *Delay*) comme étant :

$$D = \max(0, T_{estimé} - T_{objectif}). \quad (3.2)$$

Considérons maintenant N véhicules $V_1, V_2, V_3, \dots, V_N$ présents dans le réseau et une politique d'ordonnancement π . Pour chaque véhicule V_i le retard associé est noté D_i^π . Nous pouvons choisir d'exprimer notre objectif de minimisation des retards typiquement

par l'une des trois formules suivantes :

$$f_{Sum}(\pi) = \min_{\pi} E \left[\sum_{i=1}^N \frac{D_i^{\pi}}{N} \right], \quad (3.3)$$

$$f_{Max}(\pi) = \min_{\pi} E \left[\max_{i \in \{1..N\}} (D_i^{\pi}) \right], \quad (3.4)$$

$$f_{Sum^2}(\pi) = \min_{\pi} E \left[\sum_{i=1}^N \frac{(D_i^{\pi})^2}{N} \right], \quad (3.5)$$

où π est la politique appliquée et E l'espérance mathématique.

La première formule 3.3 vise à minimiser la moyenne des retards de l'ensemble des véhicules. Son défaut est d'ignorer la grande variabilité possible des retards d'un véhicule à l'autre. La deuxième formule 3.4, au contraire, vise à minimiser le pire retard parmi tous les véhicules, mais elle ne tient pas compte des autres véhicules. C'est pourquoi nous introduisons une troisième formule 3.5 visant à un compromis entre les deux premières en minimisant la moyenne des carrés des retards. Dans les trois cas, nous cherchons à obtenir un comportement global du système qui permette le partage des retards entre agents.

3.2 Comportements de coordination

Nous proposons deux types de stratégies suivies par les agents, fondées sur un principe de coordination réactive entre les véhicules en tête de files.

3.2.1 Approche par alternance

Le premier comportement est simple et s'inspire du comportement civique qu'adoptent la plupart des conducteurs pour l'insertion de deux files de véhicules sur une seule voie. Il s'agit de donner l'accès à la voie en alternant systématiquement entre les deux files de véhicules. Dans notre cas, l'arête conflit va être utilisée alternativement par un véhicule de chaque arc entrant (s'il y a des véhicules présents), cf. illustration figure 3.4.

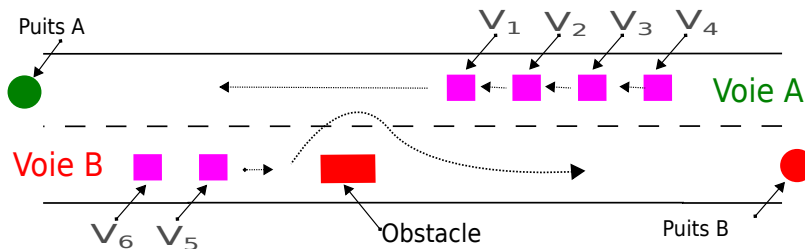


FIGURE 3.4 – Première approche : alterner les véhicules

Sur l'exemple de la figure 3.4, nous avons quatre véhicules (V_1, V_2, V_3 et V_4) sur la voie A et deux véhicules (V_5 et V_6) sur la voie B. L'ordre dans lequel vont passer les véhicules

par la voie libre (A) est $V_5 V_1 V_6 V_2 V_3 V_4$ ou bien $V_1 V_5 V_2 V_6 V_3 V_4$, selon le premier véhicule qui s'engage sur la voie, V_1 ou V_5 .

Alterner est un processus simple qui ne requiert pas de communications puisque le sens de passage change systématiquement (quels que soient les retards). Seule la perception d'un véhicule présent sur l'arête conflit ou à son entrée est nécessaire au fonctionnement. Néanmoins, il faut toutefois gérer le cas particulier de l'arrivée simultanée d'un véhicule de chaque côté de l'arête conflit alors que celle-ci ne contient aucun véhicule par un processus de *break – tie*. Pour cela, chaque véhicule émet un signal de déblocage après un délai aléatoire (très court). Dès qu'un véhicule perçoit un tel signal, et s'il n'émet pas lui-même, il s'engage sur la voie. Si les deux émettent simultanément, ils recommencent ce processus.

L'algorithme 3.1 donne l'essentiel du comportement des agents/véhicules en focalisant sur la décision de passage. C'est-à-dire qu'il considère la décision à prendre pour un agent se situant à une entrée de l'arête conflit. Les deux premiers tests en cascade correspondent à (i) la voie est déjà occupée par un véhicule de la file opposée, auquel cas il n'entre pas, sinon (ii) s'il n'y a pas d'agent sur l'entrée opposée, alors il peut s'engager. Le troisième test correspond au cas nominal, c'est-à-dire celui d'un véhicule qui s'engage après la sortie d'un véhicule ayant évolué dans le sens opposé. Reste le cas particulier de deux agents présents sur les deux entrées de l'arête conflit, alors qu'il n'y avait pas d'agent évoluant dessus (ligne 8). Dans ce cas la perception du signal de déblocage permet de s'engager, sinon chacun émet le signal après un délai aléatoire. Le dernier cas (ligne 15) est celui où l'agent doit attendre car l'arête conflit est déjà empruntée et un agent attend à l'entrée opposée.

Algorithme 3.1 : Règles pour le passage alterné d'un véhicule

```

1 Entrée : agent à l'entrée de l'arête conflit
2 si un agent de la direction opposée est sur l'arête conflit alors
3   | DoNothing();
4 sinon si pas d'agent à l'entrée opposée alors
5   | ChangeArc();
6 sinon si il y avait un agent sur l'arête conflit dans la direction opposée alors
7   | ChangeArc();
8 sinon si arrivée simultanée d'un agent à l'entrée opposée alors
9   | break – tie();
10  | si gagnant alors
11  |   | ChangeArc();
12  | sinon
13  |   | DoNothing();
14 sinon
15  | DoNothing();

```

3.2.2 Optimisation locale à court terme (OLC)

Le deuxième comportement que nous proposons essaie d'optimiser le passage en favorisant les véhicules qui sont plus en retard que les autres. Pour rester sur une décision locale comme dans la première approche, la comparaison des retards se fait en utilisant la communication uniquement entre les deux agents prêt à entrer sur l'arête conflit en même temps.

Prenons comme exemple la figure 3.5, où nous avons noté entre parenthèses le retard des véhicules en tête de chaque file d'attente (par exemple, le véhicule V_1 a 20 secondes de retard).

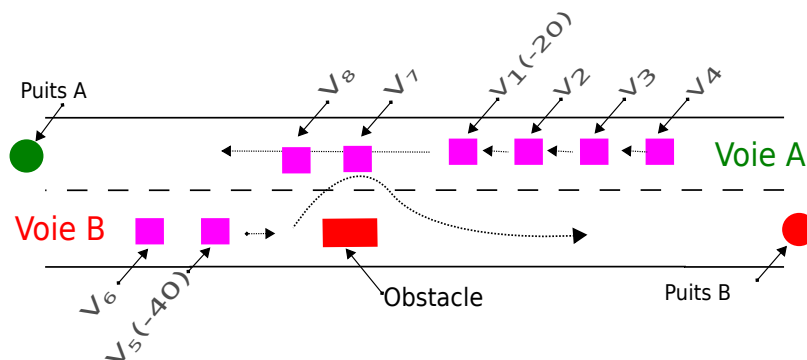


FIGURE 3.5 – Illustration de la deuxième approche : optimisation locale à court terme

Si nous ne considérons que ces retards, nous les faisons passer dans l'ordre suivant : V_5 puis V_1 . Mais si, comme dans la figure 3.5, nous avons des véhicules déjà engagés et devons tenir compte des retards, le choix de l'ordre de passage n'est plus trivial.

Pour remédier à ce problème de choix, on utilise les critères d'optimisation que nous avons présenté en section 3.1.3. Notons α et β les véhicules en tête des files d'attente de A et B respectivement (supposées non vides). Ayant choisi de ne considérer que ces deux véhicules, nous allons restreindre l'évaluation du critère choisi à ceux-ci, et n'aurons que deux stratégies à comparer, faire passer α avant β ($\alpha \rightarrow \beta$) ou β avant α ($\beta \rightarrow \alpha$). Chacun des deux véhicules peut donc dans un premier temps calculer ses deux retards possibles : $D_v^{\alpha \rightarrow \beta}$ et $D_v^{\beta \rightarrow \alpha}$ où v est α ou β , puis les transmettre à l'autre véhicule. Dans un deuxième temps, en s'appuyant sur ses propres estimations et celles reçues, chaque agent compare les deux cas de passage selon le critère d'optimisation utilisé. Par exemple, si le critère utilisé est celui de la formule 3.4, l'ordre de passage sera α puis β si $f_{Max}^{\alpha \rightarrow \beta} = (D_\beta^{\alpha \rightarrow \beta} + D_\alpha^{\alpha \rightarrow \beta})$ est plus grand que $f_{Max}^{\beta \rightarrow \alpha} = (D_\beta^{\beta \rightarrow \alpha} + D_\alpha^{\beta \rightarrow \alpha})$, sinon β puis α dans le cas contraire.

L'algorithme 3.2 donne l'essentiel du comportement du véhicule β où $f_*^{\beta \rightarrow \alpha}$ est l'application de l'un des 3 critères ($f_{Sum}^{\beta \rightarrow \alpha}$, $f_{Max}^{\beta \rightarrow \alpha}$ ou $f_{Sum2}^{\beta \rightarrow \alpha}$) si β passe avant α et $f_*^{\alpha \rightarrow \beta}$ le cas

contraire.

Algorithme 3.2 : Règles de passage de OLC pour le véhicule β

```

1 si (un agent est en attente de l'autre côté) alors
2   |   envoyer/recevoir retard();
3   |   calculer  $f_*^{\beta \rightarrow \alpha}$  et  $f_*^{\alpha \rightarrow \beta}$ ();
4   |   si ( $f_*^{\beta \rightarrow \alpha} > f_*^{\alpha \rightarrow \beta}$ ) alors
5   |   |   si (un agent de la direction opposée est sur l'arête conflit) alors
6   |   |   |   attendre jusqu'à libération de l'arête conflit;
7   |   |   |   ChangeArc();
8   |   |   sinon
9   |   |   |   DoNothing();
10  |   sinon
11  |   |   si (un agent de la direction opposée est sur l'arête conflit) alors
12  |   |   |   DoNothing();
13  |   |   sinon
14  |   |   |   ChangeArc();

```

3.3 Résultats expérimentaux

3.3.1 Simulation

Pour étudier le modèle qu'on vient de décrire, nous avons implémenté un simulateur en utilisant la plate-forme JADE⁶ (Java Agent Development Framework), laquelle offre un middleware JAVA très utile pour développer des applications multi-agents. Les actions et les décisions possibles sont les mêmes pour tous les agents. La figure 3.6 montre un exemple du simulateur développé.

Nous avons reproduit le même réseau que celui présenté dans les figures 3.1 et 3.3, avec les paramètres suivants :

- la vitesse de chaque véhicule est de 10 mètres par seconde (36 km/h) donc $l = 10\text{m}$;
- la longueur de chaque arc est de 300 mètres ($30 \cdot l$) ;
- à chacune des deux entrées du réseau, nous avons installé une source qui génère des véhicules.

Chaque source injecte les véhicules en suivant un processus de Bernoulli avec un paramètre λ sachant que :

$$\lambda = \frac{1}{T}, \quad (3.6)$$

où T est le temps moyen, en secondes, entre deux véhicules consécutifs.

6. <http://JADE.tilab.com>

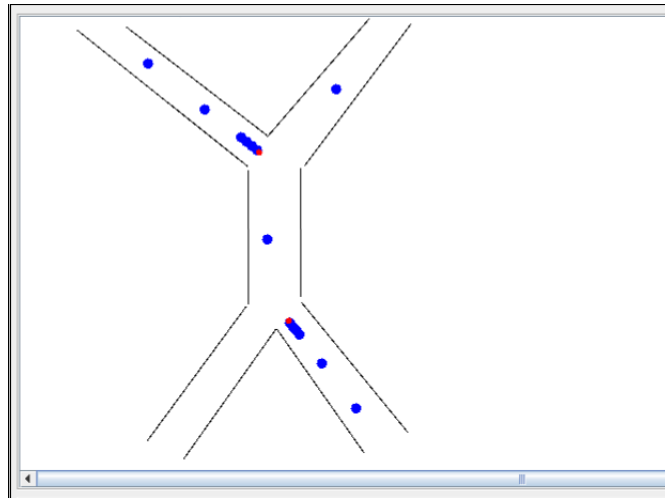


FIGURE 3.6 – Le simulateur

3.3.2 Libération des deux voies

Dans toutes nos simulations, nous avons vérifié qu'on ne rencontre pas le cas pathologique qui consiste à faire toujours passer une seule file aux dépens de l'autre. Dans la suite du chapitre, on appellera *Alt* la stratégie Alternier, et *Sum*, *Max* et *Sum*² les critères de la stratégie OLC présentés respectivement par les formules 3.3, 3.4 et 3.5.

Dans ce scénario, nous commençons toujours par la stratégie Alternier, puis, après l'injection de 50 véhicules dans le réseau, soit on continue avec la même stratégie (*Alt*), soit on choisit l'un des trois critères d'optimisation de la deuxième stratégie (*Sum*, *Max* ou *Sum*²). Après l'injection de 100 véhicules, on arrête les insertions et on attend que le réseau se vide.

La figure 3.7 montre les résultats de simulation obtenus par la stratégie Alternier et les 3 critères de l'optimisation locale (OLC). Nous avons utilisé des fréquences d'injection élevées – donc un trafic dense – en employant le processus de Bernoulli avec un paramètre $T = 10s$ pour chaque file.

Les courbes tracées, dans la figure 3.7, sont les moyennes de 100 simulations. L'axe des abscisses représente le temps en secondes et l'axe des ordonnées représente le nombre de véhicules qui sont dans le réseau à l'instant t .

On remarque que la stratégie *Alt* – *Alt* n'est pas bonne et met beaucoup de temps avant de vider le réseau. En moyenne elle dure 600 secondes. On constate aussi que, dès le déclenchement des trois critères de la stratégie OLC, le nombre de véhicules pour les stratégies *Alt* – *Sum*², *Alt* – *Max* et *Alt* – *Sum* tend à décroître pour, à long terme, voir les deux voies finalement se vider.

L'évacuation la plus rapide est donnée par la courbe *Alt* – *Sum*². D'autre part, pour le critère *Alt* – *Sum*, il existe deux phases. Dès le déclenchement de la stratégie, la courbe fait un plateau, puis on observe la plus grande pente moyenne, toutes courbes confondues. Ceci est dû à la particularité du critère *Sum* qui évite les changements de

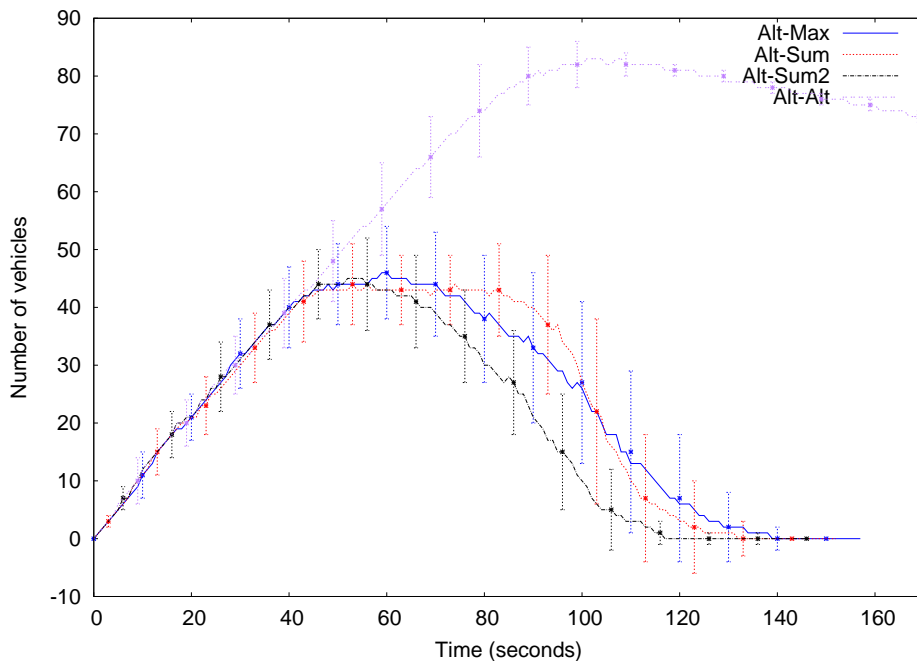


FIGURE 3.7 – Observation de l'évacuation de 100 véhicules

file. Quand il y a encore des injections, pendant que l'une des files circule, l'autre sature (les générations sont alors interdites, et donc le processus de Bernoulli n'est pas respecté). Cette saturation est un effet de bord limitant l'augmentation du nombre de véhicules. Elle fait que l'instant où l'on atteint les 100 véhicules injectés est d'autant plus retardé. Une fois les injections finies, *Alt - Sum* vide ses files le plus vite en évitant les pertes de temps liées à des changements de files. On voit que *Alt - Max* et *Alt - Sum²*, eux, ont un nombre maximum de véhicules légèrement plus élevé que *Alt - Sum*, mais pas de plateau aussi marqué.

3.3.3 Régulation d'un trafic continu

Dans ce scénario, nous calculons la moyenne des temps des traversées de chaque stratégie pour des injections avec le paramètre $T = 10s$. Nous arrêtons les simulations dès que 100 véhicules sont sortis mais nous n'arrêtons pas les injections de véhicules comme dans les simulations précédentes.

La figure 3.8, illustre une comparaison des moyennes et des écarts-types des temps de traversées. Sur l'axe des ordonnées nous avons le temps de traversée en secondes, et sur l'axe des abscisses le nombre de véhicules ayant quitté le réseau. Les courbes tracées sont les moyennes de 100 simulations.

La stratégie *Alt* est la pire encore une fois. Le critère *Sum* s'avère nettement moins bon que *Sum²* et *Max*, ne favorisant pas les changements de file et créant ainsi des accumulations de véhicules de l'un ou l'autre côté.

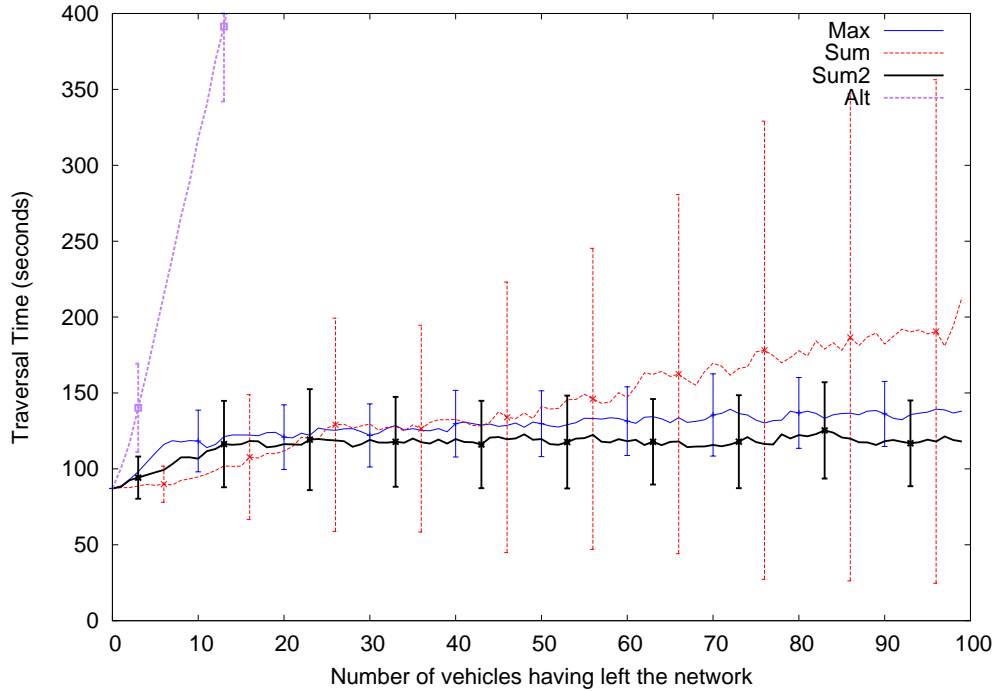


FIGURE 3.8 – Comparaison des moyennes et des écarts types des temps des traversées (T=10–10)

On remarque que les meilleurs critères sont Sum^2 et Max avec les moyennes les plus basses, en notant que le temps de traversée minimum est de 87 secondes. Max , en se préoccupant du pire retard, favorise le changement de file (voir illustration plus loin) et donc réduit la variance, alors que Sum^2 travaille plus sur la réduction de la moyenne des temps de traversées entre les véhicules.

Le tableau 3.1 donne un résumé des temps de traversées mesurées avec les écarts-types pour chaque stratégie et pour des injections de flux de véhicule $T = 10s$ et $T = 30s$. Quand on passe à des injections plus espacées dans le temps ($T = 30s$), on constate que Sum^2 reste le meilleur critère et que les trois critères de la stratégie OLC se rapprochent en moyenne et en variance.

TABLE 3.1 – Résumé des moyennes et des écarts types

Moyenne	Alt	Sum	Max	Sum^2
10 – 10	1260 ± 86	142 ± 94	127 ± 22	115 ± 28
30 – 30	782 ± 173	104 ± 28	105 ± 19	102 ± 19

Pour mieux comprendre la stratégie OLC, nous présentons dans la figure 3.9 le résultat d’une simulation prise au hasard pour chaque critère avec des injections $T = 10s$. Chaque courbe représente la séquence des 100 véhicules dans leur ordre de sortie, chacun ayant pour coordonnées sa date d’injection dans le réseau en abscisse, et son temps de traversée

en ordonnée. Les points, dans la figure 3.9, représentent les véhicules sortis, indépendamment de leur file d'appartenance (en rouge la première file et en bleu la deuxième), et permettent de voir les changements de file.

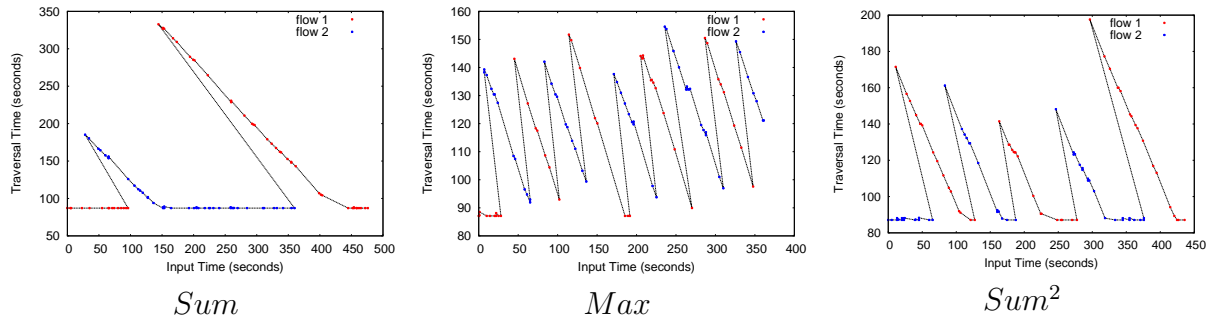


FIGURE 3.9 – Observation d’une simulation de chaque critère, date d’entrée et durée de traversée des véhicules à ($T=10-10$)

La première chose qui nous interpelle est que le critère *Sum* met beaucoup de temps avant de changer de file contrairement à *Max* qui change très souvent. D’après les figures, le critère *Sum²* apparaît comme un compromis entre les deux. Changer souvent de file fait perdre beaucoup de temps, et si on ne change pas de file on accumule les retards sur la file en attente.

3.4 Simulation avec SCANeRstudio

Dans le cadre du projet InTraDE, nous disposons de l’outil de simulation 3D SCANeRstudio⁷ qui est développé par l’entreprise OKTAL, l’un des partenaires du projet. Les deux approches ont été aussi simulées avec ce simulateur, la figure 3.10 présente des captures d’écran des deux approches.

L’obstacle est simulé par le véhicule noir. Deux trains de véhicules composés chacun de trois véhicules autonomes sont simulés, trois avec conteneur pour une voie et trois sans conteneur pour l’autre. Les sous-figures 1 (situation avant le partage de la voie) et 2 (alternance entre deux véhicules autonomes, un avec conteneur et l’autre sans conteneur) illustrent la stratégie par alternance. Les sous-figures 3 (situation avant le partage de la voie) et 4 (passage prioritaire des trois véhicules avec conteneurs) illustrent la stratégie OLC.

3.5 Conclusion

Dans ce chapitre nous avons abordé la résolution des conflits de partage de l’espace entre files de véhicules ou plus généralement entre agents mobiles (par exemple des robots).

7. <http://www.scanersimulation.com/software/software-research.html>

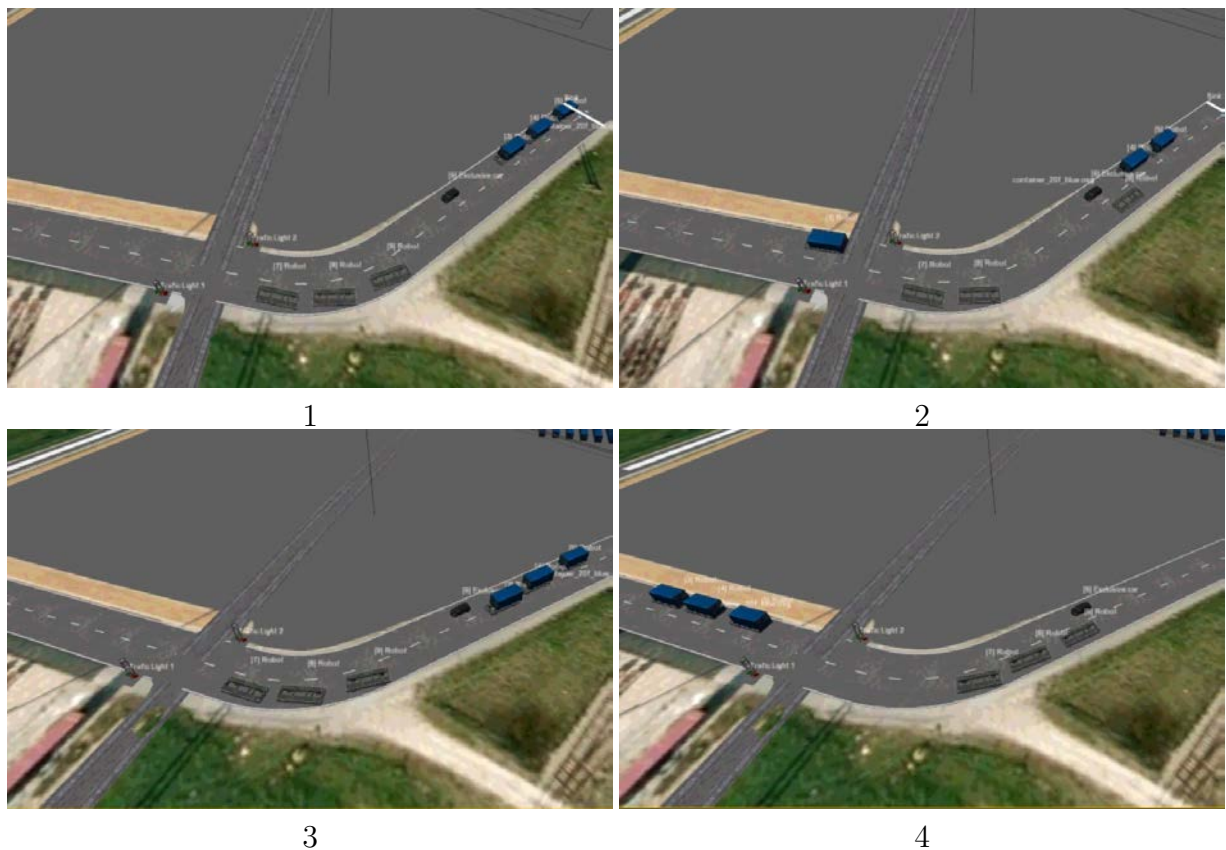


FIGURE 3.10 – Simulation avec SCANeRstudio de l’approche par Alternance (figures 1 et 2), et de l’approche OLC (figures 3 et 4)

Pour cela, nous avons exploré des modèles multi-agents reposant sur des comportements de coordination réactive. Nous avons tout d’abord proposé une stratégie utilisant uniquement les perceptions locales (Alterner) puis une autre intégrant des communications entre les véhicules en tête de files.

L’étude expérimentale a permis de montrer la capacité de régulation des conflits (des congestions) de ces comportements, engendrés dans différents scénarios de simulation du trafic. Les phénomènes d’embouteillage qui sont des phénomènes émergents indésirables, sont ici traités localement, donc indépendamment de tout système de planification extérieur, et en temps réel. L’introduction de communications simples des retards permet d’améliorer fortement une stratégie alternance utilisée couramment par les conducteurs. Plus globalement, l’ensemble des mesures révèle qu’une approche centrée sur des règles de coordination locale s’avère performante pour réguler le trafic autour des conflits générés par le problème du partage de l’espace.

Nous poursuivons cette étude, dans les chapitres suivants, en généralisant les modèles proposés au partage d’espaces plus complexes que sont les intersections de routes (carrefours).

Deuxième partie

Optimisation d'un réseau d'intersection pour véhicules autonomes

Introduction

La gestion du trafic dans les grands réseaux routiers reste un défi important dans les systèmes de transport. Une approche efficace consiste à utiliser les infrastructures existantes pour maîtriser les flux croissants des véhicules. Dans ce cadre, la régulation par systèmes intelligents, fondés sur l’intelligence artificielle, et les véhicules autonomes sont considérés comme des approches prometteuses pour traiter le contrôle du trafic.

Dans de nombreux systèmes de transport réels, les congestions sont générés au niveau des intersections entre les routes [Dresner et Stone, 2004], plus précisément, au niveau des parties de l’espace qui doivent être partagées par les véhicules. Afin d’améliorer les temps de passage, il existe plusieurs solutions pour la gestion des intersections. Les plus simples favorisent généralement un flux par rapport à un autre comme pour les feux de signalisation et les panneaux “STOP”. De tels événements sont générateurs de retards pour les véhicules car ils les obligent à des arrêts fréquents et de durées variables. Si les flux de véhicules sont importants, ces retards locaux peuvent mener à l’émergence d’embouteillages.

C’est pour cette raison que nous nous sommes intéressés dans la deuxième partie de cette thèse au problème du croisement de flux de véhicules autonomes aux intersections, c’est-à-dire lorsqu’ils doivent emprunter le même lieu au même moment.

Le chapitre 4 propose une présentation des approches existantes pour la gestion des intersections, en discutant leurs avantages et limites. Dans le chapitre 5, nous introduisons le modèle sur lequel nous nous sommes basés afin d’étudier de nouvelles approches. Ensuite, dans le chapitre 6, nous proposons une approche décentralisé qui permet aux véhicules autonomes de franchir les intersections du réseau sans s’arrêter. Cette approche, que nous appelons approche par synchronisation, permet d’améliorer le trafic en termes de retard comparée à une approche classique par feux de circulation. Enfin, au chapitre 7, nous présentons une optimisation de cette approche fondée sur différents algorithmes de recherche locale, pour passer d’une synchronisation en chaque croisement de routes à une synchronisation d’un réseau d’intersections.

Chapitre 4

Approches existantes pour la gestion d'intersections

Sommaire

4.1	Le code de la route	45
4.1.1	Règles de priorité	45
4.1.2	Les panneaux de signalisation	46
4.1.3	Les feux de circulation tricolores	47
4.1.3.1	Définition d'un flux, d'un cycle et d'une phase	47
4.2	Quel est le problème ?	48
4.3	Quatre approches de référence	49
4.3.1	TRANSYT	49
4.3.2	SCOOT	50
4.3.3	SCATS	51
4.3.4	PRODYN	51
4.3.5	Limites des approches de référence	52
4.4	Synthèse des approches	53
4.4.1	Perceptions de l'environnement	53
4.4.2	Variables de contrôle	54
4.4.3	Espace discret vs continu	54
4.4.4	Critères d'optimisation	54
4.4.5	Centralisé vs décentralisé	55
4.4.5.1	Contrôle	55
4.4.5.2	Optimisation	55
4.4.6	Adaptation instantanée vs adaptation moyenne	56
4.4.7	Algorithmes d'optimisation	56
4.4.7.1	Algorithmes heuristiques	56
4.4.7.2	Optimisation boîte noire	58
4.4.7.3	Algorithmes de contrôle	60

4.4.8	Les limites des approches	64
4.5	Approche pour véhicules autonomes (IAV)	66
4.5.1	Approche reposant sur l'historique des attentes des véhicules	66
4.5.2	Négociation véhicule à véhicule	66
4.5.3	Approche par réservations	67
4.6	Bilan	69

Ce chapitre est consacré à l'étude bibliographique des approches existantes pour les problèmes de gestion du trafic. Nous nous sommes intéressés en particulier au travail de Bazzan [2009] qui présente un aperçu des méthodes et des approches dans les domaines de l'ingénierie du trafic pour surmonter les problèmes de congestion aux intersections. Dans ce travail, Bazzan [2009] choisit une classification historique, à savoir trois grandes classes : *les approches classiques*, les plus anciennes, fondées sur les règles de coordination entre intersections ; *les approches avancées*, fondées sur la physique, l'optimisation, l'intelligence artificielle (systèmes multi-agents) et l'apprentissage ; et enfin *les approches pour les véhicules autonomes*, les plus récentes, généralement utilisées dans un environnement fermé et uniquement pour les véhicules autonomes. Cependant, nous avons choisi de présenter les travaux différemment. Il est plus simple et clair de mettre plutôt en avant les grandes dimensions selon lesquelles ces travaux se distinguent puisqu'il existe plusieurs classifications possibles. Nous proposons, ici, de présenter ces grandes dimensions en insistant sur les différents types d'algorithmes utilisés.

En section 4.1, nous dressons un aperçu des méthodes utilisées par le code de la route pour gérer les intersections et garantir la sécurité des usagers de la route. Ensuite, dans la section 4.2, nous présentons le problème de gestion d'intersections comme un problème de satisfaction des usagers et d'optimisation. Dans la section 4.3, nous détaillons quatre approches de référence qui correspondent aux systèmes les plus déployés actuellement. En section 4.4, nous mettons en avant les dimensions identifiées en fonction de plusieurs critères, à savoir les technologies utilisées, la réactivité au trafic (adaptabilité aux variations de flux de trafic), le mode de déploiement (centralisé ou pas), les algorithmes d'optimisation utilisés, et les variables observées ainsi que de contrôle. Nous avons consacré la section 4.5 à l'étude des approches spécialement conçues pour les véhicules intelligents autonomes (IAV). Une synthèse de toutes les approches est aussi présentée en section 4.6, sous forme de tableaux, permettant de dresser un bilan.

4.1 Le code de la route

Dans cette partie, nous considérons par défaut le code de la route français. Les systèmes de gestion du trafic (feux de signalisation) ne sont pas installés partout, car les autorités ne peuvent pas gérer toutes les intersections par des systèmes coûteux. Ainsi, il existe des intersections gérées par de simples panneaux et parfois des intersections dépourvues de toute signalisation. C'est pourquoi le législateur a défini des solutions simples qui doivent être respectées par les conducteurs dans ces situations.

4.1.1 Règles de priorité

Par exemple le code de la route français, dans son article *R.415 – 5*⁸, stipule : *Lorsque deux conducteurs abordent une intersection par des routes différentes, le conducteur venant*

8. <http://www.legifrance.gouv.fr>

par la gauche est tenu de céder le passage à l'autre conducteur, sauf dispositions différentes prévues. Ceci implique une règle simple à respecter à chaque intersection sans signalisation souvent appelée *priorité à droite*.

Nous trouvons aussi dans l'article *R.415 – 10* : *Tout conducteur abordant un carrefour à sens giratoire est tenu, quel que soit le classement de la route qu'il s'apprête à quitter, de céder le passage aux usagers circulant sur la chaussée qui ceinture le carrefour à sens giratoire*. Cet article annule donc l'effet de la première règle de priorité à droite et définit une nouvelle règle de priorité à gauche appliquée uniquement pour les carrefours à sens giratoire souvent appelés *rond-points*.

Ces règles ne sont pas communes à tous les pays. Elles diffèrent d'un pays à un autre et surtout dans les pays où les véhicules circulent à gauche sur la chaussée. Par exemple, au Royaume-Uni, la priorité aux rond-points est pour les véhicules qui viennent de droite et il n'existe pas de règle générale pour les intersections, la priorité étant généralement indiquée. Notons que les règles de priorité du code de la route sont une forme simple de coordination réactive.

4.1.2 Les panneaux de signalisation

En France, il existe des panneaux pour la gestion des intersections. Par exemple, parmi les panneaux les plus répandus nous trouvons les panneaux de signalisation de céder le passage. En application de l'article *R.415 – 7* du code de la route français : *A certaines intersections indiquées par une signalisation dite "cédez le passage", tout conducteur doit céder le passage aux véhicules circulant sur l'autre ou les autres routes et ne s'y engager qu'après s'être assuré qu'il peut le faire sans danger*. Il est souvent utilisé pour définir une route principale par rapport à une route secondaire et n'oblige pas les usagers de la route secondaire à l'arrêt.

Il existe aussi un autre panneau très répandu dans presque tous les pays du monde : le panneau "STOP". Si un conducteur trouve ce panneau implanté à une intersection sur son chemin, il doit obligatoirement marquer l'arrêt avant de continuer. Il indique que chaque véhicule sur cette voie n'est jamais prioritaire quelle que soit la direction qu'il souhaite prendre. Ainsi nous trouvons dans l'article *R.415 – 6* du code de la route français : *A certaines intersections indiquées par une signalisation dite stop, tout conducteur doit marquer un temps d'arrêt à la limite de la chaussée abordée. Il doit ensuite céder le passage aux véhicules circulant sur l'autre ou les autres routes et ne s'y engager qu'après s'être assuré qu'il peut le faire sans danger*. Ce panneau est généralement utilisé pour contrôler le trafic aux intersections dangereuses, où le trafic est dense, mais pas suffisamment encombrées pour installer un système de feux de signalisation.

Ces panneaux que nous venons de citer sont très utilisés dans les villes et sont connus pour leur fiabilité. Ils sont toujours présents et ne risquent pas de mal fonctionner. C'est pour cette raison qu'ils peuvent aussi être présents à côté des feux de signalisation afin d'assurer le trafic en cas de défaillance des systèmes électroniques.

4.1.3 Les feux de circulation tricolores

Dans les systèmes de gestion du trafic, le système le plus connu et le plus utilisé dans le monde est le système des feux de signalisation. En consultant l'encyclopédie libre en-ligne Wikipedia, on peut trouver cette définition des feux de signalisation⁹ :

Un feu de circulation routière est un dispositif permettant la régulation du trafic routier entre les usagers de la route, les véhicules et les piétons. Les feux destinés aux véhicules à moteurs sont généralement de type tricolore, auxquels peuvent s'ajouter des flèches directionnelles. Ceux destinés aux piétons sont bicolores et se distinguent souvent par la reproduction d'une silhouette de piéton. Les feux tricolores pour cyclistes se distinguent par la reproduction d'une bicyclette.

Généralement, un feu tricolore est composé d'un système électronique commandé. Il est composé de trois couleurs principales. La couleur rouge indique l'obligation d'arrêt aux véhicules. La couleur orange qui ne dure que quelques secondes signale le passage du rouge au vert. La couleur verte indique aux véhicules qu'ils ont la priorité exclusive pour passer. Ces couleurs ont été choisies parce qu'elles ont l'avantage d'être très distinctes.

Le système des feux de signalisation est le système le plus efficace pour la gestion du trafic car il évite tout malentendu entre les différents conducteurs au moment du passage. *Cependant, c'est aussi le système qui génère le plus de retard puisqu'il favorise à chaque instant un ou deux flux (qui ne se croisent pas) et oblige systématiquement l'arrêt de tous les autres flux entrants.*

4.1.3.1 Définition d'un flux, d'un cycle et d'une phase

Dans cette section, nous introduisons les notions de *flux*, de *cycle* et de *phase*, lesquelles vont nous suivre dans cette thèse. A une intersection, *un flux* de véhicules est l'ensemble des véhicules entrant par une voie donnée et ressortant par une autre. Le trafic dans une intersection est constitué d'un ensemble de flux de véhicules, chacun provenant d'une source. Deux flux sont cohérents s'ils peuvent évacuer l'intersection simultanément. Une intersection gérée par des feux de signalisation est composée de plusieurs feux tricolores, implantés sur les différentes voies entrantes dans l'intersection.

Une phase d'un feu est une période durant laquelle un ou plusieurs flux cohérents sont admis dans le carrefour. *Le cycle* d'un feu représente la durée qui sépare deux phases identiques de l'intersection. Il est défini par une séquence de phases. La figure 4.1 présente un cycle de feux composé de quatre phases.

Ainsi, la durée d'un cycle d'un feu sera d'autant plus élevée que le nombre de phases sera grand. Le "temps de dégagement" est un temps perdu qui est interposé entre deux phases consécutives. On l'appelle aussi le rouge intégral. Ce rouge est la durée nécessaire pour que tous les véhicules qui sont dans le carrefour puissent évacuer la zone de conflit avant l'admission d'une autre phase. On remarque donc que l'augmentation de la durée

9. <http://www.wikipedia.org>

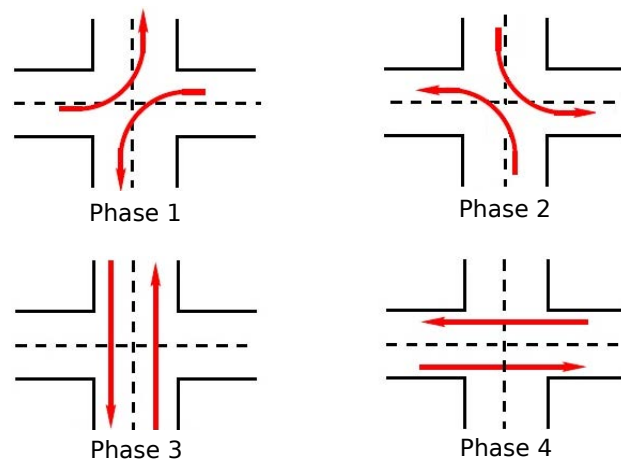


FIGURE 4.1 – Plan d'un cycle composé de 4 phases

du cycle permet d'augmenter la capacité du carrefour, ceci est lié à la minimisation de la part du temps de dégagement. Néanmoins, une durée d'évacuation trop grande favorise l'apparition de blocages au niveau des carrefours suivants. Un autre concept qui est le *déphasage* entre les carrefours adjacents est très important dans la modélisation et le contrôle du trafic géré par les feux de signalisation. Ce paramètre permet de contrôler la création de feux verts successifs pour une voie donnée appelés aussi *ondes vertes*. Il est calculé à partir de la distance entre deux feux consécutifs et d'une vitesse moyenne des véhicules [Mammar, 2007].

4.2 Quel est le problème ?

La demande croissante de mobilité dans nos sociétés a engendré de nouveaux problèmes. Plus précisément, à partir de la deuxième moitié du siècle dernier, le phénomène de congestion du trafic est apparu, plus connu sous le nom d'embouteillage. Ceci est dû au fait que de nombreux véhicules veulent utiliser les mêmes infrastructures routières au même moment.

Réguler le trafic dans un réseau routier est au moins un problème de *satisfaction* puisqu'il faut satisfaire les besoins des usagers de la route, c'est-à-dire qu'ils puissent atteindre leur destination. Une façon triviale de résoudre le problème de la gestion d'intersections est d'arrêter tous les flux pour n'en laisser passer qu'un sous-ensemble, comme c'est le cas dans la gestion actuelle des intersections dans les villes (phase d'un feu de signalisation). Or il a été montré que cette stratégie est coûteuse en termes de temps [Fok et al., 2012] puisqu'elle oblige l'arrêt systématique des flux. Donc, la régulation du trafic est aussi un problème d'optimisation puisqu'il s'agit de trouver les règles de fonctionnement du réseau (au sens large) qui induisent la meilleure qualité de trafic possible, par exemple le fluidifier et ainsi réduire les temps d'attentes de chaque véhicule.

La régulation du trafic est réalisée en définissant les règles de comportement des véhicules (leurs conducteurs) et des systèmes de signalisation placés dans l'environnement (mar-

quages au sol, panneaux stop, panneaux priorité, feux de circulation...). Le code de la route définit ainsi le comportement que doit suivre un conducteur en fonction des actions des autres véhicules et de la signalisation environnante. Ce code, s'il est respecté, assure la sécurité de chacun, mais pas nécessairement la satisfaction de chacun (arrivée à destination), ni l'optimisation du trafic. Pour améliorer le trafic sur un réseau, on peut jouer :

- sur les besoins des utilisateurs (en répartissant mieux les zones d'habitation, de commerces, d'industrie...),
- sur le réseau, en ajoutant des routes ou des voies sur certaines routes, mais avec des implications fortes en termes de coûts financiers, environnementaux (bruit, pollution, gêne visuelle),
- sur la disposition de la signalisation,
- sur le contrôle (les règles de fonctionnement) des signalisations actives/dynamiques (feux de circulation), et
- sur le contrôle (le comportement) des véhicules.

Ici, nous supposons fixes les besoins, le réseau, et la disposition de la signalisation. Comme nous allons le voir, la plupart des travaux rencontrés jouent sur le contrôle des signalisations, mais quelques uns s'intéressent aussi au contrôle des véhicules.

4.3 Quatre approches de référence

Les approches de référence correspondent aux systèmes les plus déployés, car ils sont les plus anciens, mais aussi parce qu'ils sont simples et ne nécessitent pas beaucoup d'infrastructure comparés à d'autres. A partir de 1960, plusieurs algorithmes ont été proposés pour améliorer le passage des flux de véhicules. Quatre systèmes différents, TRANSYT, SCOOT, SCATS et PRODYN, sont décrits pour montrer la diversité de ces approches classiques. Ils s'intéressent tous à la gestion du trafic dans les réseaux sans être centralisés (à l'exécution) pour autant. Nous nous sommes appuyés dans cette section sur le livre [Mammar, 2007] qui présente une synthèse de ces différents systèmes de gestion de trafic.

4.3.1 TRANSYT

TRANSYT (*TRAffic Network StudY Tool*) [Robertson, 1969; Mammar, 2007] est l'un des premiers systèmes proposés. Il repose sur une optimisation hors-ligne qui génère des plans de coordination optimaux entre les feux de signalisation d'un réseau pour une période donnée. TRANSYT exige beaucoup de paramètres d'entrées, comme par exemple la géométrie des artères des intersections, le débit des véhicules, le taux de véhicules sur chaque voie sortante de chaque intersection (fixé à l'avance), le temps de feu vert minimal, des plans de feux initiaux, et des valeurs initiales pour les durées des cycles et les déphasages. A partir de ces paramètres, un modèle mathématique (calculé à partir de

la dispersion des trains de véhicules) est simulé macroscopiquement et ses performances sont évaluées. Enfin vient l'étape de l'optimisation à l'aide d'un algorithme d'optimisation Hill-Climbing, c'est-à-dire améliorant progressivement la solution en modifiant légèrement la durée des feux vert et des décalages entre intersections adjacentes. Les plans récupérés à la fin sont des plans calculés pour une situation statique dans le temps qui dépend des paramètres d'entrée.

Pour résumer, TRANSYT est un système centralisé et n'est pas adaptatif au trafic. Une application réussie de ce modèle nécessite un calibrage approprié des paramètres du modèle. Plus précisément, Guebert et Sparks [1990] ont montré qu'un étalonnage précis des paramètres est essentiel pour le développement de plans de synchronisation efficaces.

4.3.2 SCOOT

SCOOT (*Split Cycle and Offset Optimisation Technique*) [Hunt et al., 1981; Robertson et Bretherton, 1991; Mammari, 2007] est un système décentralisé et complètement adaptatif à la situation du trafic. Il optimise :

- les durées de feu vert pour chaque intersection indépendamment,
- les décalages entre intersections voisines, et
- les cycles des feux entre les zones d'intersection.

Il collecte les données (nombre de véhicules par intervalle de temps) à partir des détecteurs installés sur les routes, boucles magnétiques en amont du carrefour (voir figure 4.2).

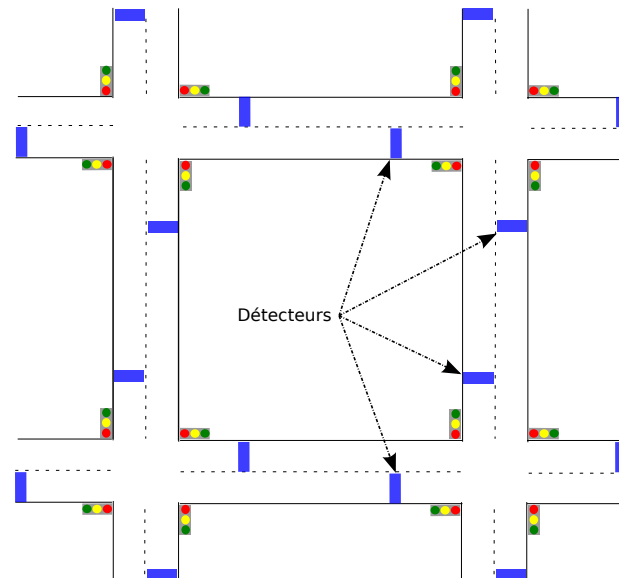


FIGURE 4.2 – Boucles magnétiques en amont des carrefours (SCOOT)

Très similaire à TRANSYT, mais fonctionnant en temps réel, il compare différentes options (par exemple pour les durées des cycles : reproduire les mêmes plans, ajouter ou retrancher quelques secondes) et choisit celle qui le rapproche le plus de son objectif d'optimisation en utilisant l'algorithme Hill-Climbing. Il modifie graduellement les plans

de feux de signalisation en cours d'exécution en fonction des données collectées par les capteurs installés sur les routes. Ces modifications consistent en de petites variations de la durée des temps de cycle (0, 4 ou 8 secondes), des temps de feux vert (0 ou 4 secondes) et des décalages entre intersections (0 ou 4 secondes).

Ce système est installé dans beaucoup de villes dans le monde et principalement en Grande-Bretagne. Néanmoins, SCOOT reste un système faiblement adaptatif comparé à d'autres, vu ses faibles variations graduelles des phases à chaque cycle.

4.3.3 SCATS

SCATS (*Sydney Co-ordinated Adaptative Traffic System*) [Lowrie, 1982; Mammar, 2007] est un système partiellement décentralisé et adaptatif à la situation du trafic. SCATS utilise aussi les données des capteurs installés sur les routes.

Pour contrôler le trafic, SCATS s'appuie sur une notion de sous-systèmes qui regroupent les intersections (de 1 à 10 intersections par sous-système) et où une seule intersection est jugée critique (définie par l'utilisateur du système). Chaque sous-système s'optimise indépendamment des autres sous-systèmes. L'approche de SCATS consiste à utiliser des bibliothèques prédéfinies de décalages et de durées de feu vert avec un algorithme temps réel de reconstruction de plan de feux pour intersection. Les bibliothèques (fournis par l'utilisateur du système) stockent généralement 10 ensembles de décalages et 4 ensembles de durées de feu vert. A partir de ces ensembles, l'algorithme reconstruit les plans des feux une fois par cycle sachant que la variation ne peut excéder les 6 secondes. L'algorithme compare plusieurs solutions et les données des capteurs avant d'appliquer la solution qui minimise la saturation des routes. Par ailleurs, dans SCATS il existe deux types de décalages : à l'intérieur d'un sous-système et entre deux sous-systèmes voisins. Les décalages sont sélectionnés aussi à partir des bibliothèques et en fonction des axes les plus saturés.

Comme SCOOT, SCATS est installé dans beaucoup de villes dans le monde et principalement en Australie (dans les villes contenant plus de 10 millions d'habitants). La différence avec SCOOT est que SCATS est un système hiérarchique qui dépend fortement du choix des bibliothèques des durées de feu vert et des décalages.

4.3.4 PRODYN

PRODYN (*PROgrammation DYNamique*) [Henry et al., 1984; Mammar, 2007] est un système décentralisé et adaptatif au trafic développé par le CERT (Centre d'Etude et de Recherche de Toulouse) en France.

L'objectif de PRODYN est d'optimiser la circulation en ligne en minimisant les retards aux intersections sur un horizon futur de 80 secondes. La fréquence de modification des phases et des durées des feux est fixée à 5 secondes (pas de discrétisation temporelle). Il utilise la programmation dynamique pour minimiser les retards en s'appuyant sur un

modèle d'écoulement du trafic sur tout l'horizon. Ce dernier lui permet d'exprimer l'évolution des files d'attente en fonction des arrivées et des départs sur les tronçons. Pour les départs, comme SCATS et SCOOT, il utilise les données des capteurs installés sur les routes. Cependant, pour les arrivées, il lui faudrait idéalement les données des départs des intersections en amont, eux mêmes en cours d'optimisation, ce qui est impossible.

La solution choisie consiste à utiliser des prédictions établies au pas d'optimisation précédent tout en faisant l'hypothèse que la situation du trafic n'a pas beaucoup évolué. Etant donnée la complexité calculatoire exponentielle en fonction du nombre d'intersections, PRODYN optimise chaque intersection séparément en fonction des données reçues de ses voisines. Il est installé dans beaucoup de villes en France et en Belgique, mais il reste un système relativement complexe en termes de calcul et coûteux en communication.

4.3.5 Limites des approches de référence

Plusieurs aspects peuvent être perçus comme des limitations de ces systèmes. Nous présentons ici ceux qui nous paraissent les plus importants.

Hypothèses simplificatrices

Le premier problème est que ces différents systèmes (ces quatre plus d'autres) font des hypothèses simplificatrices (prédiction du trafic pour TRANSYT, variation des durées de feux selon des pas prédéfinis pour SCOOT, bibliothèques prédéfinies de décalages et de durées de feu vert pour SCATS, etc...) qui ne reflètent pas forcément la réalité.

Ces hypothèses sont essentielles pour ces systèmes, sinon les plans des feux de signalisation deviennent complexes à calculer et donc impossibles à exécuter en temps réel.

Approches autonomes pour véhicules non autonomes

Un autre problème de ces différents systèmes est qu'ils ont été conçus pour les véhicules conduits par les humains. En effet, l'utilisation de véhicules classiques (non autonomes) pose surtout problème parce que, si les feux évoluent (optimisation en ligne), il faut "idéalement" prévenir les véhicules pour qu'ils puissent adapter leurs profils d'accélération au mieux.

En plus, nous avons remarqué que les optimisations proposées favorisent le plus souvent les routes principales, généralement les plus congestionnées, et défavorisent les routes secondaires, sachant que dans les grandes villes ces dernières sont devenues de plus en plus importantes dû à la saturation des axes traditionnels.

Enfin, d'autres aspects sont délaissés par ces systèmes. Par exemple, le trafic peut être affecté par des incidents (accidents, blocages des routes, etc...). Il est donc nécessaire d'avoir des approches robustes et flexibles pour une bonne gestion du trafic.

4.4 Synthèse des approches

Dans cette section, nous ne nous limitons pas à la synthèse des quatre approches précédentes, mais détaillons toutes les approches de gestion du trafic [Bazzan, 2009]. Nous décrivons, ici, les principales caractéristiques des approches rencontrées, puis nous donnons un tableau récapitulatif.

4.4.1 Perceptions de l'environnement

En fonctionnement, les feux de signalisation les plus simples ne perçoivent rien. Dans ce cas, ils ne peuvent pas s'adapter aux conditions de circulation réelles. Par exemple, nous pouvons citer le système TRANSYT où les auteurs ne peuvent que pré-régler leurs cycles respectifs, et éventuellement leurs déphasages en fonction des conditions de trafic prédites. Ces réglages peuvent aussi dépendre du jour de la semaine ou de l'heure de la journée.

Dans d'autres approches plus évoluées, telles que les systèmes SCOOT et SCATS, nous pouvons trouver des feux de signalisation reliés à des capteurs (boucles magnétiques sous les voies, caméras, ...) permettant ainsi :

- d'estimer en continu le trafic, donc le débit, sur les différentes voies [Rochner *et al.*, 2006; Bazzan, 2005], ou
- de prédire les arrivées des véhicules à court terme, comme c'est le cas pour PRODYN.

Il existe aussi des approches où les véhicules eux-mêmes préviennent les contrôleurs des intersections de leurs arrivées. Par exemple, Dresner et Stone [2005] utilisent un agent contrôleur à chaque intersection du réseau, et chaque véhicule en approche doit communiquer avec ce dernier afin de réserver un créneau de passage.

Pour mieux faire encore, les phases d'une intersection peuvent communiquer entre elles pour se prévenir mutuellement de leurs activités, et donc des arrivées prévisibles de véhicules. Dans [Kosonen, 2003], l'auteur introduit une nouvelle méthode de contrôle des feux de circulation qui combine la simulation en temps réel, le contrôle multi-agent, et la logique floue. Pour chaque intersection, les phases sont modélisées comme un groupe d'agents. Chaque agent a la possibilité de modifier le cycle du groupe, si les autres agents sont d'accord, ce qui implique un besoin de négociation et d'échange de données locales de trafic.

Dans le même contexte de la communication entre intersections, Sánchez et Aguirre [2007] comparent deux protocoles fondés sur la négociation et les enchères entre les agents afin de définir les cycles des feux de circulation.

En ce qui concerne les véhicules (quand ceux-ci sont contrôlés), ils peuvent éventuellement :

- percevoir, en plus des signalisations routières, les autres véhicules, et
- communiquer avec ceux-ci, comme c'est le cas du contrôleur proposé par Naumann et Rasche [1997] fondé sur les négociations entre véhicules afin de déterminer un ordre de passage à travers l'intersection.

4.4.2 Variables de contrôle

Dans la littérature de la gestion d'intersections, sont contrôlés : les feux de circulation ou les véhicules. En effet, la majorité des travaux considèrent les allumages des feux de signalisation comme variables de contrôle [Brockfeld *et al.*, 2001; Steingrover *et al.*, 2005; France et Ghorbani, 2003]. Plus précisément, ils varient l'ordre des phases, la durée des cycles et les déphasages entre intersections. Cependant, quelques travaux considèrent les véhicules eux-mêmes comme variables de contrôle, bien que leurs parcours ne peuvent être modifiés. Dans ces cas, les auteurs les font ralentir ou accélérer pour qu'ils passent au bon moment à un certain endroit [Dresner et Stone, 2004; Naumann *et al.*, 1997].

4.4.3 Espace discret vs continu

Certaines approches emploient des modèles de l'environnement discrets [Brockfeld *et al.*, 2001; Gershenson, 2004; Wiering, 2000]. Cela a toutefois le défaut de contraindre les déplacements des véhicules à des arrêts et des accélérations brusques et donc peu réalistes. Cependant, cette solution permet de faciliter la simulation.

Le reste des approches, qui sont plus réalistes à notre sens, soit emploient un modèle continu de l'environnement tels que les solutions de Kosonen [2003], Nunes et Oliveira [2004] ou Köhler *et al.* [2005], soit sont indépendantes, c'est-à-dire que l'optimisation fait abstraction de la notion d'espace [Rochner *et al.*, 2006; Balan et Luke, 2006].

4.4.4 Critères d'optimisation

Comme nous l'avons précisé au début de ce chapitre, le problème de gestion d'intersections est en général vu comme un problème d'optimisation. Dans les travaux rencontrés, ce n'est pas toujours explicite, mais souvent les auteurs partent du fait qu'ils sont confrontés à un problème dans lequel il faut maximiser ou minimiser une grandeur. Nous nous focalisons, dans cette section, sur les différents critères d'optimisation rencontrés.

Nous pouvons distinguer deux critères principaux :

- La plupart des auteurs se sont concentrés sur la minimisation du retard moyen des véhicules. C'est-à-dire qu'ils considèrent le retard de tous les véhicules, à chaque instant, en comparant leurs temps de parcours à un temps calculé théoriquement [Mohring *et al.*, 2006; Köhler *et al.*, 2005; Hourri *et al.*, 2011; Hounsell et Shrestha, 2012]. A notre connaissance, il n'existe pas de travaux considérant la minimisation d'un autre type de retard tel que le pire retard par exemple.
- D'autres travaux, tels que dans [Ferreira et Khosla, 2000; Bazzan, 2005; Camponogara et Kraus Jr, 2003], se sont intéressés à la minimisation des temps d'attentes des véhicules au niveau des intersections. Dans le même contexte, Da Silva *et al.* [2006] considèrent la minimisation du nombre total des véhicules à l'arrêt à chaque intersection.

Cependant, il existe aussi d'autres critères dans la littérature, mais qui restent des cas particuliers dus à des contraintes de modélisation ou des contraintes algorithmiques. Par exemple nous pouvons citer les travaux de :

- [Nunes et Oliveira \[2004\]](#) qui considèrent la minimisation du taux d'occupation sur les voies (nombre de véhicules par rapport à la taille du réseau), calculées d'une façon centralisée par l'environnement de simulation, ou
- [Richter et al. \[2007\]](#) qui maximisent le débit en maximisant le nombre de véhicules qui entrent dans les intersections, car leur environnement de simulation est discret.

Nous remarquons que ces critères ne sont pas vraiment différents les uns des autres, c'est-à-dire qu'il tendent tous à fluidifier la circulation des réseaux de transport en laissant quitter les véhicules le plus rapidement possible.

4.4.5 Centralisé vs décentralisé

Les systèmes multi-agents ont fourni un moyen efficace pour décentraliser les approches. Dans cette partie, comme la plupart des auteurs, nous distinguons l'exécution (le contrôle) et l'optimisation.

4.4.5.1 Contrôle

A notre connaissance, la plupart des approches impliquent des contrôleurs complètement décentralisés. En effet, un contrôleur centralisé serait trop complexe et non robuste aux problèmes de communications. Il est généralement suffisant pour un feu de signalisation de connaître des informations liées à son voisinage afin d'agir efficacement. Cependant, dans les travaux rencontrés, il en existe que nous qualifions de "centralisés par zones" [[Robertson, 1969](#); [Mohring et al., 2006](#); [Köhler et al., 2005](#)], c'est-à-dire que, pour un ensemble de feux de circulation, les auteurs affectent un unique contrôleur en charge de réguler cette zone.

4.4.5.2 Optimisation

L'optimisation aussi se fait rarement de manière centralisée, parce que le problème à résoudre serait trop complexe (trop de paramètres à régler simultanément). Par contre, décentraliser/distribuer l'optimisation implique typiquement de définir un critère d'optimisation local à chaque intersection (mesurable en fonction de ce qui se passe dans le voisinage de l'intersection), ce qui pose les difficultés suivantes :

- Il n'est pas évident de pouvoir définir de tels critères locaux tels que l'optimisation de ceux-ci garantisse l'optimisation du critère global initial. De ce fait, plusieurs auteurs se contentent de l'optimisation du critère local [[Ferreira et Khosla, 2000](#); [Da Silva et al., 2006](#)] et n'ont pas de garanties "globales".
- Chaque agent (intersection ou véhicule) poursuit son propre objectif, donc nous sommes dans le cadre d'un problème de théorie des jeux générique, dans lequel les

joueurs ne collaborent pas nécessairement. C'est pour cette raison que [Camponogara et Kraus Jr \[2003\]](#), [Bazzan \[2005\]](#) et [Mandiau et al. \[2008\]](#) présentent le problème comme un problème de la théorie des jeux. Dans ce cas, les algorithmes d'optimisation distribués risquent de ne pouvoir trouver un équilibre, voire de ne jamais se stabiliser sur une solution fixe (les joueurs essayant perpétuellement d'adapter leurs comportements individuels aux comportements des autres).

4.4.6 Adaptation instantanée vs adaptation moyenne

Nous distinguons dans cette section deux types de contrôleurs : les contrôleurs qui s'adaptent à des conditions moyennes de trafic (par exemple les flux moyens des véhicules) et ceux qui s'adaptent à des situations instantanées (par exemple un état particulier du système).

La plupart des travaux rencontrés optimise les phases des intersections en fonction des flux moyens dans les réseaux de transport [[Robertson, 1969](#); [Hunt et al., 1981](#); [Gershenson, 2004](#)].

Néanmoins, nous avons rencontrés aussi des algorithmes tels que ceux fondés sur : l'apprentissage par renforcement [[Wiering, 2000](#); [Steingrover et al., 2005](#)] qui s'adaptent à un état particulier du système, PRODYN [[Henry et al., 1984](#)] qui replanifie en fonction de la situation courante, ou les négociations [[Naumann et Rasche, 1997](#)] et les réservations qui se font directement entre véhicules autonomes.

4.4.7 Algorithmes d'optimisation

Nous avons rencontré dans la littérature diverses approches pour optimiser la gestion d'intersections. Nous avons décidé de classer ces différents algorithmes en trois grandes classes :

- *les algorithmes heuristiques*, souvent fondés sur des règles prédéfinies par ses concepteurs, et qui ne garantissent pas d'optimalité (même locale) ;
- *les algorithmes d'optimisation boîte noire*, qui considèrent le problème comme une fonction à optimiser sans exploiter le fait qu'il s'agit d'un problème de contrôle ;
- *les algorithmes de contrôle*, qui visent à trouver les contrôleurs qui vont satisfaire certains critères d'optimisation.

4.4.7.1 Algorithmes heuristiques

[Gershenson \[2004\]](#) propose un modèle de feux de signalisation fondé sur l'auto-organisation. Il définit trois méthodes pour synchroniser les feux de signalisation sans communication directe entre eux. Ces méthodes sont implémentées dans un simulateur qui ressemble beaucoup à celui de la figure 4.4, faisant de nombreuses hypothèses simplificatrices.

Les méthodes se basent toutes sur le même principe : chaque feu compte le nombre de véhicules i approchant de l'intersection en fonction du pas de simulation t_s quand le

feu est rouge et indépendamment de la vitesse du véhicule (c'est-à-dire qu'il soit en mouvement ou à l'arrêt). Quand $k_i = i * t_s$ atteint un seuil θ , le feu de signalisation vire à l'orange puis au pas de temps suivant au vert, alors que le feu adverse vire à l'orange aussi, puis au rouge et k_i est remis à zéro.

Ce mécanisme simple permet d'obtenir l'auto-organisation de la façon suivante : S'il y a quelques voitures à l'arrêt derrière un feu rouge, elles restent bloquées jusqu'à ce que d'autres arrivent (cela donne du temps aux autres véhicules afin de les rejoindre). Plus les véhicules sont nombreux, moins longtemps ils restent à l'arrêt. Avec un nombre suffisant de voitures, les feux rouges virent au vert avant même qu'ils n'atteignent l'intersection, ce qui induit des ondes vertes dans le réseau. Ainsi, les trains de véhicules mobiles améliorent la fluidité du trafic par rapport à une distribution homogène. Les zones vides entre les trains de véhicules de la même direction sont utilisées pour faire traverser les trains de véhicules de l'autre direction. Cette méthode n'exige pas de phase ni d'horloge interne. Les feux de circulation ne changent que lorsque les conditions ci-dessus sont remplies. S'il n'y a pas assez de voitures qui s'approchent d'un feu rouge, le feu adverse peut rester au vert. Toutefois, en fonction de la valeur de θ , les densités de trafic élevées peuvent déclencher une commutation rapide des feux de signalisation qui ralentira la circulation. La différence entre les variantes réside dans l'ajout de paramètres de contrôle. Dans la deuxième méthode proposée, l'auteur ajoute un paramètre afin d'éviter le changement brusque des feux, et dans la troisième méthode il ajoute un paramètre pour fixer la taille des trains de véhicules.

France et Ghorbani [2003] proposent un système multi-agent hiérarchique pour optimiser le trafic urbain. Au premier niveau, ils définissent un agent de trafic local (AL) à chaque intersection du réseau. Les AL sont en charge de sélectionner les plans des feux de signalisation, parmi des modèles fournis par l'utilisateur du système, en fonction de l'état de la circulation locale (au niveau de chaque intersection). Au deuxième niveau, ils proposent un agent de coordination du trafic (AC) qui est en charge de la supervision d'un certain nombre d'agents AL voisins comme le montre l'exemple de la figure 4.3.

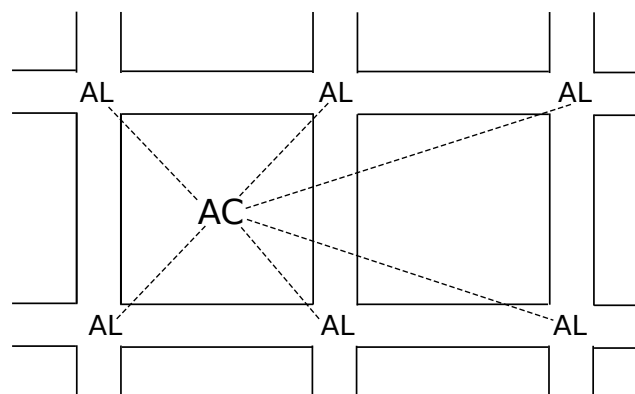


FIGURE 4.3 – Illustration du modèle de France et Ghorbani [2003] avec une architecture hiérarchique composée de six agents AL et un agent AC

D'après les auteurs, le rôle des AC est de fournir un moyen par lequel l'optimum local est légèrement modifié pour améliorer la performance globale. Ce n'est pas un objectif trivial

mais, dans l'article, les auteurs ne détaillent pas tout le fonctionnement du système : les moyens de communication entre les composants locaux et l'AC, et la mise en œuvre. Enfin, le troisième niveau est contrôlé par un agent d'information centralisé (AI) qui fournit les moyens de stockage de toutes les informations pertinentes pour gérer les congestions.

L'exemple proposé est très intéressant. Lorsqu'une intersection est congestionnée, les voisins sont informés afin de réagir en conséquence. Cependant, dans l'article, il n'est pas clair comment les AC calculent une solution "globale" optimale. En plus, dans ce cas, pourquoi ne pas utiliser tout le temps les AC (les AI sont inutiles). Aussi, les auteurs supposent que les AL sont coopératifs et ne précisent pas comment ils font pour la résolution d'éventuels conflits entre eux.

4.4.7.2 Optimisation boîte noire

Dans cette classe d'algorithmes d'optimisation, nous avons identifié trois sous-classes, les algorithmes d'optimisation Hill-Climbing, les algorithmes évolutionnaire et les algorithmes de programmation mathématique.

Hill-Climbing

Plusieurs approches d'optimisation Hill-Climbing existent dans la littérature, mais nous nous contentons de mentionner les trois premières approches présentées dans la section 4.3, à savoir TRANSYT, SCOOT et SCATS. C'est un algorithme itératif de recherche locale des plus simples, qui commence avec une solution arbitraire à un problème, puis tente de trouver une meilleure solution en changeant progressivement la solution, un paramètre à la fois. Si le changement produit une meilleure solution, une mise à jour de la solution est faite, répété jusqu'à ce qu'aucune autre amélioration puisse être trouvée. Nous reviendrons sur cet algorithme en chapitre 7.

Algorithmes évolutionnaires

Nunes et Oliveira [2004] ont étudié la question de l'amélioration de l'apprentissage de plusieurs contrôleurs hétérogènes en utilisant la communication entre agents. Ce que nous voulons dire par contrôleurs hétérogènes, c'est l'utilisation de différents algorithmes d'apprentissage, où les informations de plusieurs agents sont partagées et utilisées dans une simulation simplifiée d'un problème de contrôle de la circulation.

Dans ce travail, ils définissent des équipes de deux d'agents qui sont en charge de deux intersections connectées dans une zone du réseau. Chaque intersection est contrôlée par un agent différent. Les membres d'une équipe peuvent communiquer avec leur partenaire dans la même zone ou avec des membres d'autres équipes qui résolvent des problèmes similaires dans différentes zones (trois zones ont été utilisées dans le travail cité). La récompense à chaque pas de temps est la somme pondérée de deux termes qui représentent le compromis entre l'amélioration de la performance individuelle à une intersection, et la

qualité globale dans la zone contrôlée par l'équipe. Différents types d'agents ont été utilisés et comparés : les agents-EA (Algorithme Évolutionnaire) qui sont fondés sur l'évolution d'une population de réseaux de neurones artificiels [Rumelhart *et al.*, 1985], les agents-QL (Q-Learning connexionniste) utilisant l'algorithme ConnqQL [Lin, 1992], et les agents-H (agents heuristiques) qui, selon un seuil, mettent au vert les feux de signalisation des voies qui ont des taux d'occupation élevés.

Optimisation mathématique

Brockfeld *et al.* [2001] ont proposé une approche inspirée des travaux de Biham *et al.* [1992] pour la modélisation de la circulation au centre ville, et des travaux de Nagel et Schreckenberg [1992] pour la modélisation de la circulation autoroutière. Le modèle proposé est fondé sur des automates cellulaires qui représentent un modèle minimal de simulation dans le sens où il est capable de reproduire plusieurs caractéristiques de base du trafic réel en utilisant seulement quelques règles comportementales qui définissent la dynamique des véhicules. Bien que leur modèle d'automates cellulaires soit très simple, comme le montre la figure 4.4 (discrétisation de l'espace, routes avec une seule voie, etc), les auteurs ont étudié et simulé l'impact de la coordination entre feux de signalisation.

L'objectif de ce travail consistait à trouver les équations mathématiques pour déterminer les temps de cycle optimaux des feux de circulation du réseau qui a une géométrie très simple (réseau carré à deux dimensions). Dans ce cas, en raison à la fois de la géométrie du réseau et des densités similaires dans les deux directions du trafic (Nord et Est), les auteurs ont pu déterminer les paramètres qui favorisent la création d'ondes vertes avec une large bande passante.

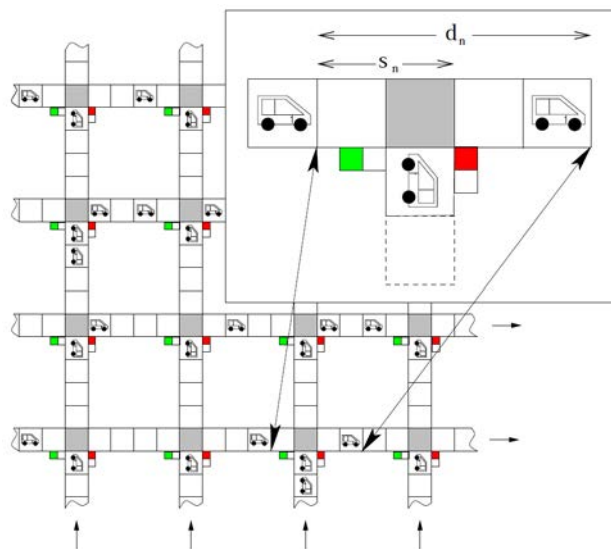


FIGURE 4.4 – Présentation du modèle (automates cellulaires) de Brockfeld *et al.* [2001] : les routes ont deux directions (Nord et Est) et sont composées d'une seule voie.

Enfin, nous pouvons citer les travaux de Mohring *et al.* [2006], lesquels introduisent deux

approches d'optimisation des feux de signalisation, à l'aide de la programmation linéaire mixte en nombres entiers (MIP). Les inconvénients des approches proposées résident dans le fait qu'elles ne traitent que les feux de signalisation avec des durées fixes et un nombre limité d'intersections à la fois, car la programmation linéaire mixte est associée à un grand temps de calcul.

On retrouve les mêmes problèmes dans les travaux de Köhler *et al.* [2005]. Les auteurs décrivent un modèle pour l'optimisation des décalages des feux de signalisations de plusieurs intersections. Leur solution est modélisée aussi à l'aide de la programmation linéaire mixte en nombres entiers (MIP). Cependant, elle prend en charge des temps de cycles non-uniformes sous certaines conditions de saturation et les durées des feux de signalisation sont fixes. Néanmoins, ces approches garantissent un optimum global sur les intersections contrôlées.

4.4.7.3 Algorithmes de contrôle

La plupart des travaux que nous allons présenter dans cette section sont fondés sur le Q-learning. C'est pour cette raison que nous introduisons d'abord son fonctionnement.

Apprentissage par renforcement : Q-learning

Dans ce type d'approches, il s'agit de permettre au système d'apprendre un comportement maximisant une mesure de performance définie d'une manière numérique. C'est cette mesure que le système cherche à optimiser sur le long terme. Généralement, l'apprentissage par renforcement est modélisé avec les processus décisionnels de Markov (MDP). Ce sont des processus stochastiques contrôlés satisfaisant la propriété de Markov, assignant des récompenses aux transitions d'états [Sigaud et Buffet, 2010].

Ils sont définis par un quadruplet (S, A, T, r) où :

- S est l'espace d'états dans lequel évolue le processus ;
- A est l'espace des actions qui contrôlent la dynamique de l'état ;
- T décrit les transitions possibles du système ; et
- r est la fonction de récompense sur les transitions.

L'algorithme Q-learning est fondé sur l'estimation des valeurs état-action, qu'on appelle les Q-valeurs (fonction de valeurs). Plus précisément, une Q-valeur $Q(s, a)$ représente la somme actualisée maximale de récompense future qu'un agent peut espérer recevoir s'il commence à l'état s , choisit l'action a et puis continue à suivre une politique optimale. L'algorithme Q-learning estime $Q(s, a)$ pendant que l'agent agit dans un environnement donné. Son équation de mise à jour est la suivante :

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right], \quad (4.1)$$

où α est le taux d'apprentissage et γ est un facteur d'actualisation compris entre 0 et 1 et permettant, selon sa valeur, de prendre en compte les récompenses plus ou moins loin dans le futur (pour le choix des actions de l'agent). La fonction de valeur de l'agent sera donc :

$$V(s) = \max_{a \in A} Q(s, a). \quad (4.2)$$

Si toutes les paires état-action sont visitées infiniment, alors l'algorithme Q-learning garantit de converger vers les Q-valeurs correctes avec une probabilité de 1 [Watkins et Dayan, 1992].

Par exemple, dans [Ferreira et Khosla, 2000], les auteurs proposent un système multi-agent utilisant des fonctions de valeurs distribuées. En d'autres termes, un Q-learning distribué, inspiré des travaux de Schneider *et al.* [1999]. Dans le modèle proposé, pour chaque intersection (agent), l'état local est représenté par l'ensemble des durées des feux de signalisation, le nombre de véhicules présent sur chaque voie entrante dans l'intersection et le temps depuis la dernière mise à jour du plan des feux. Le critère d'optimisation est la minimisation du temps d'attente des véhicules au feu vert. Ce contrôleur a été implémenté dans un simulateur d'une petite section de la zone urbaine de Pittsburgh, PA (USA). Les résultats présentés montrent que le système d'apprentissage implémenté dans 7 intersections réduit les temps d'attente au feux verts mieux qu'un système avec un plan de feux fixe. Selon les auteurs, l'algorithme Q-learning distribué a trouvé une solution qui est la somme pondérée des futures récompenses espérées sur tous les agents. Néanmoins, la sélection des paramètres et la question de la convergence doivent être discutées car, même si chaque intersection atteint son optimum local, cela n'implique pas un optimum global au niveau du réseau.

Camponogara et Kraus Jr [2003] reviennent sur ce point et considèrent que calculer un ensemble de politiques, une pour chaque agent intersection, n'est pas forcément optimal d'un point de vue global. Le problème est que cet ensemble de politiques, même s'il existe, ne peut égaler les performances d'une politique centralisée optimale qui maximise la somme des récompenses de tous les agents, ce qui est l'objectif principal de l'exploitation d'un réseau de transport. Dans la solution qu'ils proposent, la tâche de contrôle de chaque agent est modélisée comme un problème d'apprentissage par renforcement ordinaire, en considérant les autres agents comme une partie de l'environnement, et tout en sachant que cet environnement est non-stationnaire puisqu'il dépend des autres politiques implémentées par les autres agents.

Planification en ligne

PRODYN (voir section 4.3.4) fait de la planification en ligne à l'aide de la programmation dynamique. En effet, à chaque pas de temps, il calcule une solution optimale des plans des feux pour un horizon fini. Ceci revient à résoudre un nouveau MDP à chaque itération. Il entre donc bien dans la catégorie des algorithmes de contrôle.

Approche par phéromones

Les travaux d'éthologues sur le comportement collectifs d'animaux sociaux comme les fourmis [Robinson, 1992; Gordon, 1996] ont abouti à la mise au point d'algorithmes et d'applications fondés sur l'intelligence en essaim.

De Oliveira et Bazzan [2006, 2007] proposent une nouvelle approche fondée sur l'utilisation des phéromones. Les stimuli d'accomplissement ou de modification des tâches sont fournis par les véhicules qui, en attendant le prochain feu vert, déposent des phéromones. Ainsi, le volume du trafic en provenance d'une direction peut être évalué par l'agent situé à l'intersection, ce qui peut déclencher une commutation des plans des feux de signalisation. Chaque intersection se comporte de façon à réduire les retards et fluidifier le trafic tout en influençant indirectement les intersections voisines. Les plans des feux de signalisation sont considérés ici comme des tâches devant être exécutées par les agents/intersections, sans aucun contrôle centralisé ou mécanisme d'allocation de tâches. La particularité de ce système est de ne pas utiliser d'autres informations.

L'approche a été implémentée et testée dans un simulateur de trafic microscopique. Les agents intersections perçoivent les phéromones et sélectionnent les plans de feux appropriés. Les expérimentations montrent que les intersections tendent à se synchroniser pour améliorer le trafic, sans aucune décision centralisée. Cependant, il est mentionné que de telles approches prennent beaucoup de temps pour converger vers une solution optimale. Le temps nécessaire pour atteindre une coordination stable peut être élevé, ce qui est incompatible avec un environnement hautement dynamique.

Nous remarquons, comme dans plusieurs travaux de la littérature [Koenig *et al.*, 2001; Glad, 2011], que les approches par phéromones sont très proches des algorithmes d'apprentissage par renforcement tels que le Q-learning. En effet, l'utilisation des phéromones déposées, est assimilable à l'utilisation de la fonction de valeur. Néanmoins, l'approche proposée par De Oliveira et Bazzan [2006] est un peu différente car, les phéromones sont déposées par les véhicules mais la régulation est traitée au niveau des agents intersections (et non au niveau des agents mobiles).

Théorie des jeux

Bazzan [2005] a proposé aussi des techniques de la théorie des jeux évolutionnaires, où chaque agent (représentant une intersection) agit dans un environnement dynamique non seulement pour son propre objectif local, mais aussi pour un objectif global. Ce résultat est obtenu avec des agents ayant seulement des connaissances locales qu'ils obtiennent à partir de leurs capteurs locaux. Néanmoins, ils effectuent aussi des simulations ailleurs dans le réseau, et reçoivent une récompense calculée à partir d'une matrice de gain. Pendant le processus d'apprentissage, une évaluation pour chaque stratégie est calculée et influence la prochaine génération de stratégies qui seront utilisées par les agents pour effectuer les expérimentations.

Dans le même contexte, Doniec *et al.* [2005] proposent aussi un modèle de coordination fondé sur la théorie des jeux, dans lequel les joueurs sont les véhicules et non plus les

intersections. Dans ce travail, les auteurs montrent qu'un mécanisme non-anticipatif de coordination n'est pas suffisant pour faire face à des situations de circulation complexes. Pour y remédier, ils proposent d'ajouter une couche anticipative reposant sur la manipulation de réseaux de contraintes. Notons que le but de ce travail est de simuler le trafic urbain.

Co-learning

Wiering [2000] et Steingrover *et al.* [2005] ont proposé des approches multi-agents utilisant l'apprentissage par renforcement pour les feux de signalisation et les véhicules. Dans leurs études, chaque intersection apprend une fonction de valeurs qui estime les temps d'attentes de certaines configurations des plans des feux de signalisation. En plus, ils introduisent la notion de Co-learning. Ils ont étudié comment utiliser le même système d'apprentissage par renforcement, et les fonctions de valeurs apprises par les intersections, pour que chaque véhicule calcule sa trajectoire optimale vers sa destination. Donc, tous les agents (véhicules et intersections) partagent le même objectif d'utilisation d'une politique qui minimise la même fonction de valeurs. Ces solutions sont très intéressantes. Cependant, la complexité du problème de gestion de trafic et la complexité de la solution (calculatoire) font d'elles des solutions non-réalistes et difficiles à appliquer dans des circonstances réelles [Bazzan, 2009].

Détection de contexte

Dans [Da Silva *et al.*, 2006], à cause du caractère non-stationnaire du trafic, les auteurs proposent l'utilisation d'un apprentissage par renforcement avec détection de contexte. Plus précisément, un agent central apprend plusieurs politiques optimales selon le contexte de circulation, donc plusieurs modèles de l'environnement. Le contexte est ensuite identifié en ligne.

Algorithme évolutionnaire

Rochner *et al.* [2006] proposent une architecture multi-couche pour le contrôle des feux de signalisation. Il se sont appuyés sur les travaux de Bull *et al.* [2004] qui introduisent l'utilisation des systèmes de classeurs (Learning Classifier System) et les travaux de Helbing *et al.* [2005] qui proposent une stratégie décentralisée pour le contrôle des flux de trafic.

Leur système est composé de trois couches comme le montre la figure 4.5. La première contient un contrôleur de trafic simple au niveau de chaque intersection qui peut s'adapter à la situation du trafic en modifiant les plans des feux de signalisation (en utilisant les données des capteurs). La deuxième couche surveille en temps réel la situation du trafic au niveau de l'intersection et modifie les paramètres de contrôle de la première en cas de besoin. Les données des capteurs de la première couche sont aussi reçues par la deuxième et

servent comme paramètres d'entrées au système de classeurs (LCS). Le problème est que cette optimisation ne s'applique qu'à une seule intersection, c'est à dire que l'optimisation à ce niveau est seulement locale au mieux. D'où la nécessité d'une troisième couche qui introduit un algorithme évolutionnaire (AE) fonctionnant hors ligne. L'algorithme génère de nouveaux LCS, optimisés en fonction des données des capteurs et les insère dans la première couche.

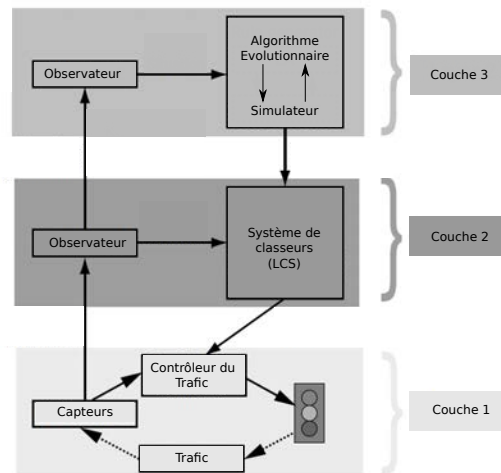


FIGURE 4.5 – Illustration du modèle de Rochner *et al.* [2006]

Pour résumer, la méthode pourrait être utile lorsqu'elle est appliquée à un réseau d'intersections si les problèmes de performance étaient traités (surtout de l'AE). Cependant, les auteurs ne présentent pas de résultats d'évaluation.

Descente de gradient

Enfin, Richter *et al.* [2007] s'intéressent aux méthodes de gradient pour la recherche de politiques optimales. Une autre façon de faire le calcul de la fonction de valeur, consiste à effectuer une recherche directe de la politique (par descente de gradient par exemple), ce qui nécessite d'utiliser un contrôleur paramétré. Dans ce travail, les auteurs utilisent un algorithme connu sous le nom de NAC (Natural Actor-Critic) dans sa version en ligne. Les algorithmes acteur-critique, comparés à des approches classiques de descente de gradient, utilisent une approximation de la fonction de valeur pour estimer le gradient de la fonction à optimiser. Ceci n'est pas sans coût de calcul considérable, et c'est pour cette raison que les auteurs supposent des simplifications telles que : les véhicules ont tous une vitesse uniforme, et ils ignorent les interactions entre véhicules sauf quand ils sont dans les files d'attentes. Ce qui a attiré notre attention dans ce travail, c'est la comparaison faite dans la partie expérimentation. En effet, les auteurs comparent l'approche développée (NAC), une descente de gradient classique (PG) sans estimation de la fonction de valeur, le système SCATS (décrit dans la section 4.3.3), une approche qui applique des durées de feux équitables dans toutes les directions, et enfin une approche qui applique des durées au hasard. Ces approches ont été testées dans différents scénarios de trafic (en fonction

des débits et de leurs directions). Les expérimentations montrent que les algorithmes NAC et PG ont presque les mêmes résultats : ils améliorent la circulation bien plus que SCATS et les autres stratégies dans tous les scénarios réalisés. L'algorithme NAC le fait en trois fois moins de pas d'apprentissage, mais parfois avec des temps de calcul en plus, comparé à PG. Néanmoins, ces méthodes sont souvent complexes à appliquer dans un cas réel où les flux de trafic changent fréquemment, car cette forme d'optimisation est centralisée et vise à trouver l'optimum global au niveau du réseau.

4.4.8 Les limites des approches

Les approches sus-citées (toutes les approches à part les quatre de référence) que nous appellerons les approches avancées, sont toutes très intéressantes et innovatrices. Nous voulons aborder, dans cette section, les problèmes rencontrés par ce type d'approches et donc leurs limites à une implémentation dans les systèmes de gestion de trafic réels.

Complexité des solutions vs. simplicité des modèles

Dans beaucoup de travaux présentés, nous remarquons que la majorité des auteurs testent leurs algorithmes sur des simulateurs où plusieurs aspects du trafic réel ont été négligés afin de réduire la complexité calculatoire, comme c'est le cas par exemple pour les approches issues de l'apprentissage par renforcement. En plus, même en simplifiant les modèles, l'exécution des algorithmes restent parfois aussi coûteuses en termes de temps de calcul.

Nous savons que les objectifs de certains travaux sont clairs (étude des algorithmes, étude du comportement du trafic,...). Néanmoins, nous remarquons que les auteurs adaptent aussi l'environnement à leur besoin, par exemple à l'aide de la discrétisation, pour que leurs solutions fonctionnent. C'est le cas de l'approche par phéromones. La mise en œuvre des phéromones dans un environnement réel comme une ville reste difficile. Ceci justifie que ces solutions ne sont pas actuellement envisageables dans un cadre réel de trafic. Toutefois, il existe des travaux qui essaient de mettre en œuvre des systèmes avec des phéromones pour la robotique d'intérieur [Glad *et al.*, 2010].

Manque de coordination entre intersections

Un autre problème de la plupart des solutions réside dans le fait qu'il n'existe pas assez de travaux qui traitent le problème de coordination entre intersections. Beaucoup de travaux s'intéressent uniquement à l'optimisation locale (au niveau de chaque intersection) afin d'optimiser globalement la circulation des véhicules alors que cette optimisation n'est pas forcément systématique.

Prenons comme exemple le réseau de deux intersections présenté par la figure 4.6. Si pour l'intersection 1 la circulation est dense dans le sens $a \mapsto b$, et pour l'intersection 2

la circulation est dense dans le sens $c \mapsto d$, et si chacune d'entre elles essaye d'optimiser localement, le résultat final est que tous les véhicules passés par l'intersection 2, vont être ralentis au niveau de l'intersection 1. Pour résumer, optimiser localement n'implique pas forcément une optimisation globale, sauf si les flux de véhicules ne sont pas en conflit.

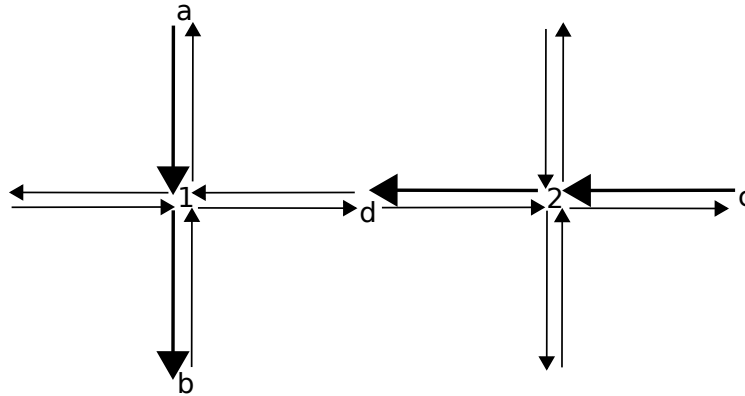


FIGURE 4.6 – Coordination entre intersections

4.5 Approche pour véhicules autonomes (IAV)

Dans cette section, nous considérons les solutions qui s'intéressent au contrôle de véhicules intelligents autonomes (IAV), comme c'est le cas du projet InTraDE. Nous avons choisi de présenter les travaux spécialement dédiés aux IAV mais aussi les modèles qui, à notre sens, ne peuvent s'appliquer que sur ce type de véhicules. Nous avons décidé de ne pas mentionner les travaux de Co-learning (apprentissage des trajectoires des véhicules et des plans des feux des intersections) introduits dans la section 4.4 car les intersections ne tiennent pas compte de cette forme d'intelligence.

4.5.1 Approche reposant sur l'historique des attentes des véhicules

[Balan et Luke \[2006\]](#) proposent un contrôleur fondé sur l'historique des attentes des véhicules et qui vise à fournir une sorte d'équité globale. Ils se sont inspirés de travaux antérieurs, tels que ceux de [Wiering \[2000\]](#) qui a exploré ce qui se passe lorsque les intersections connaissent les durées de voyage des conducteurs.

Dans cette approche, les informations des véhicules sont recueillies par les feux de circulation (principalement la part du trajet parcouru par rapport à leur objectif final). Ils essaient de garantir une équité en fonction de l'historique des retards des véhicules. Ceci est réalisé à l'aide d'un système de crédits accumulés avec les retards et dépensés pour gagner en priorité. Les feux de circulation prennent leurs décisions sur les crédits restants de plusieurs véhicules à l'intersection. Lorsqu'un véhicule arrive à un carrefour, il rend compte à l'intersection de son temps moyen d'attente sur toutes les intersections

passées, ce qui permet d'établir les plans de feux. Ce temps est aussi utilisé comme une mesure pour évaluer l'efficacité de la commande.

L'idée de ce travail est intéressante. Néanmoins, cette approche optimise les flux au niveau de chaque intersection localement, ce qui ne garantit pas ses performances au niveau macroscopique du réseau.

4.5.2 Négociation véhicule à véhicule

L'une des premières approches pour véhicules autonomes a été proposée par [Naumann et Rasche \[1997\]](#). Cette approche repose essentiellement sur la communication et la négociation entre les véhicules afin de déterminer l'ordre de passage et de sortie des intersections [[Naumann et al., 1997](#)]. Dans ce travail, les auteurs modélisent une intersection comme un ensemble de trajectoires parcourues par les véhicules avec des zones critiques où on ne peut avoir qu'un seul véhicule à la fois. Dans la figure 4.7, les points noirs représentent ces zones critiques.

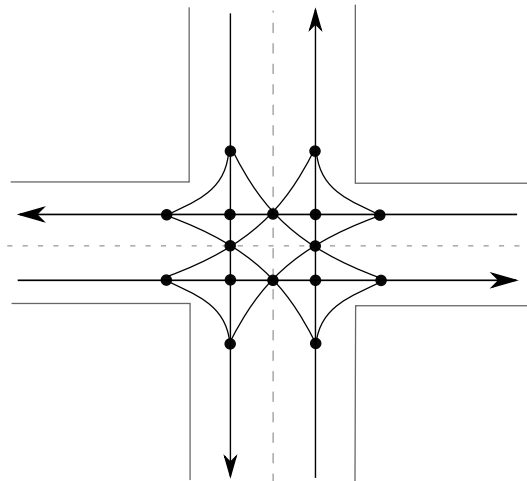


FIGURE 4.7 – Modélisation de la géométrie des intersections

L'idée de base de cette approche est l'utilisation d'un protocole de communication type Anneau à jetons (Token-ring) entre les véhicules autonomes dans une zone déterminée autour de l'intersection. Ainsi, les informations essentielles sur la position, la vitesse et l'accélération des véhicules sont échangés en vue d'une décision sur l'opportunité ou non de s'engager dans l'intersection. Pour une exécution complètement décentralisée, le véhicule le plus proche de l'intersection et qui est en possession du jeton, prend le rôle de l'agent manager. Ainsi, il décidera de l'ordre de priorité de sortie de tous les autres véhicules de l'intersection. Sa tâche se termine juste avant de quitter l'intersection quand, à son tour, il doit alors passer le jeton à un autre véhicule. Par ailleurs, l'évitement de collision est assuré au moyen d'algorithmes fondés sur des sémaphores qui permettent à un seul véhicule à la fois de rester dans un segment critique d'intersection.

Cette approche est connue pour ses limites pratiques. Premièrement, elles dépendent clairement du nombre de véhicules voulant négocier leurs passages par l'intersection, car

l'augmentation du nombre des véhicules implique une complexité croissante lors des négociations. Deuxièmement, elle suppose que toutes les intersections ont la même géométrie, ce qui n'est pas réaliste dans le contexte actuel (il n'y a pas que des routes avec une seule voie et des angles de 90°). Si nous avons plus de voies, nous aurions plus de calculs et plus de temps de traitement. Enfin, cette approche dépend aussi des capacités de communication : Peut-on assurer la réception de tous les messages envoyés par les véhicules ?

4.5.3 Approche par réservations

L'approche par réservations a été introduite par [Dresner et Stone \[2004\]](#). Elle se passe des feux de signalisation conventionnels et à la place elle introduit un agent manager au niveau de chaque carrefour. Chaque véhicule voulant passer doit réserver un temps de passage et une route à prendre, et respecter un certain nombre de règles. Parmi ces règles, il doit garder une vitesse constante fixée à l'avance à l'approche de l'intersection et il ne doit pas changer de route (c'est à dire que les véhicules vont toujours tout droit).

La réservation se déroule comme suit : Le véhicule informe l'agent manager en envoyant un message contenant sa vitesse, sa direction, son accélération maximale, sa décélération maximale, le temps qu'il lui reste avant d'arriver à l'intersection (estimé par l'IAV) et d'autres propriétés du véhicule. Chaque agent manager connaît les informations nécessaires sur la géométrie de son intersection. Il discrétise alors la zone de croisement sous forme de grille, comme le montre la figure 4.8, afin de pouvoir simuler la trajectoire des véhicules à l'intérieur de la zone de conflit.

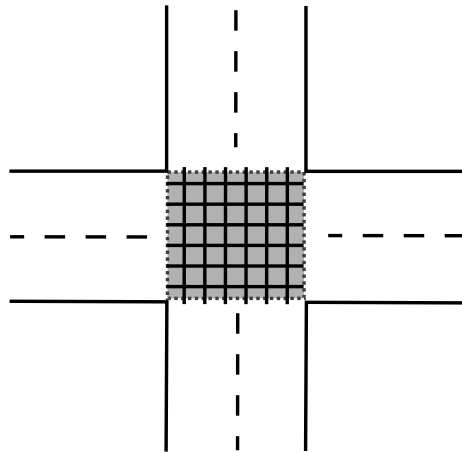


FIGURE 4.8 – Discrétisation de la zone de croisement par l'agent manager

Ensuite, l'agent manager simule le trajet du véhicule avec les autres trajets déjà réservés en fonction du temps (l'espace qui va être réservé). Si le trajet en question est en conflit avec d'autres réservations (c'est à dire l'espace discrétisé est réservé au même moment par deux véhicules), alors la réservation est rejetée. Le véhicule doit alors décélérer et demander une autre réservation. S'il arrive à l'intersection sans avoir une réservation, alors il doit s'arrêter. Si la réservation est acceptée alors le véhicule peut passer.

Dans [Dresner et Stone, 2005], certaines des hypothèses précédentes ont été assouplies : il n'est plus exigé de maintenir une vitesse constante dans l'intersection et la possibilité de changement de route a été introduite. Concernant la vitesse, dans la deuxième version de l'approche, l'agent manager peut valider une réservation si, en modifiant la vitesse, il ne risque pas d'y avoir de collisions. Le protocole amélioré proposé repose sur des règles que les véhicules sont censés suivre : le véhicule doit essayer de suivre les actions prévues par l'agent manager et il ne peut pas essayer d'améliorer son parcours (l'agent manager ignore une nouvelle demande si une réservation a été déjà validée).

L'avantage de cette approche est que, si on a plusieurs véhicules qui veulent passer et que leurs trajectoires ne se croisent pas, alors on peut satisfaire tout le monde avec un passage simultané. Cependant, dans le cas contraire, on est obligé de donner la priorité à un véhicule par rapport aux autres, ce qui revient à utiliser le principe des feux de signalisation.

4.6 Bilan

Le premier chapitre de cette partie a été l'occasion d'étudier les approches déjà existantes pour la gestion d'intersections. Nous avons commencé par présenter les solutions traditionnelles à base de règles de coordination (stop, panneaux, etc...) lesquelles garantissent uniquement la sécurité des usagers de la route.

Ensuite, nous avons introduit les solutions avancées pour la gestion du trafic et les approches conçues pour les véhicules autonomes, qui ont comme objectif principal l'optimisation de la circulation, en mettant en avant les problèmes de réalisation mais aussi les limites posées par ces différents systèmes.

Enfin, nous donnons une synthèse des approches dans deux tableaux récapitulatifs. Nous récapitulons toute la section 4.4 dans le tableau 4.2, qui synthétise les différentes approches et qui les classe selon un certain nombre de dimensions. Pour chaque approche, nous donnons ici :

- les perceptions de l'algorithme,
- les variables de contrôle,
- l'optimisation dans un environnement discret ou continu,
- le critère d'optimisation,
- le mode de fonctionnement de l'algorithme (centralisé ou décentralisé),
- l'adaptation de l'algorithme d'optimisation (moyenne ou instantanée),
- et enfin l'algorithme d'optimisation.

Dans le tableau 4.1, adapté d'après Bazzan [2009], les travaux sont classés d'abord selon que l'approche s'intéresse : (i) à une intersection isolée à la fois, (ii) à un groupe d'intersections, ou (iii) à l'optimisation globale du réseau de transport. On y distingue aussi trois grandes classes d'approches :

- les approches de référence (section 4.3),
- les approches avancées (section 4.4) divisées en trois sous-classes :
 - les approches fondées sur la physique, les mathématiques ou les automates cellulaires,

- les approches multi-agents (SMA),
- les approches fondées sur l'apprentissage,
- et les approches pour véhicules autonomes (IAV) (section 4.5).

TABLE 4.1 – Résumé des solutions citées

	Intersections Isolées	Intersections connectées (optim. locale)	Optim. globale
App. Référence	PRODYN		TRANSYT SCOOT SCATS
App. Avancées			
Physi.	[Gershenson, 2004]		[Brockfeld <i>et al.</i> , 2001] [Köhler <i>et al.</i> , 2005] [Mohring <i>et al.</i> , 2006]
SMA	[Bull <i>et al.</i> , 2004] [Doniec <i>et al.</i> , 2005]	[Kosonen, 2003] [Sánchez et Aguirre, 2007] [France et Ghorbani, 2003] [Bhourri <i>et al.</i> , 2011] [Hounsell et Shrestha, 2012] [Rochner <i>et al.</i> , 2006]	[De Oliveira et Bazzan, 2006] [De Oliveira et Bazzan, 2007]
Apprent.	[Da Silva <i>et al.</i> , 2006]	[Camponogara et Kraus Jr, 2003] [Nunes et Oliveira, 2004] [Steingrover <i>et al.</i> , 2005] [Wiering, 2000] [Ferreira et Khosla, 2000] [Richter <i>et al.</i> , 2007]	[Bazzan, 2005]
App. IAV	[Dresner et Stone, 2004] [Balan et Luke, 2006] [Naumann et Rasche, 1997]		

Ayant présenté d'une façon assez indépendante toutes les approches, nous remarquons que presque toutes ces solutions optimisent une ou des intersections indépendamment de ce qui se passe au niveau du réseau. Nous pouvons constater à partir du tableau 4.1 qu'il manque beaucoup de travaux de recherche s'intéressant à l'optimisation globale des intersections dans les réseaux de transport, déjà dans les approches dites avancées, mais surtout il n'en existe aucune à notre connaissance dédiée aux véhicules intelligents autonomes. C'est pour cette raison que nos travaux, dans les chapitres suivants, vont désormais s'intéresser au développement d'idées pour l'optimisation du réseau de transport d'IAV, premièrement d'un point de vue local puis ensuite d'un point de vue global.

TABLE 4.2: Présentation de toutes les solutions citées selon les différentes dimensions identifiées

Approches	Perceptions	Var de contrôle	Discret/ Continu	Critère	(Dé) Centralisation	Adaptation	Algo d'optim
TRANSYT [Robertson, 1969]	nbre de véhicules	plan des feux	?	temps	centralisé	moyenne	Hill-Climbing
SCOOT [Hunt et al., 1981]	nbre de véhicules	plan des feux	?	temps	décentralisé	moyenne	Hill-Climbing
SCATS [Lowrie, 1982]	nbre de véhicules	plan des feux	?	temps	décentralisé	moyenne	Hill-Climbing
PRODYN [Henry et al., 1984]	nbre de véhicules	plan des feux	?	temps	décentralisé	instantanée	Prog dyn ¹²
[Brockfeld et al., 2001]	densité des flux	phases	discret	retard	centralisé	moyenne	mathématique
[Gershenson, 2004]	véhicules en attente	plan des feux	discret	temps	décentralisé	moyenne	à base de règles
[Köhler et al., 2005]	temps perdu	phases	continu	retard	centralisé	moyenne	MIP
[Mohring et al., 2006]	phéromones	phases	continu	retard	centralisé	moyenne	MIP
[De Oliveira et Bazzan, 2006]	densité des flux	plan des feux	discret	temps	décentralisé	instantanée	à base de règles
[France et Ghorbani, 2003]	nbre véhicules	plan des feux	discret	flux	décentralisé	instantanée	à base de règle
[Rochner et al., 2006]	nbre véhicules	plan des feux	indépendant	retard	centralisé	moyenne	AE ¹³
[Kosonen, 2003]	temps perdu	plan des feux	continu	retard	décentralisé	instantanée	logique floue
[Sanchez et Aguirre, 2007]	files d'attente	plan des feux	continu	retard	décentralisé	instantanée	négociation
[Bhouri et al., 2011]	files d'attente	plan des feux	discret	temps	décentralisé	instantanée	à base de règles
[Hounsell et Shrestha, 2012]	files d'attente	plan des feux	discret	temps	décentralisé	instantanée	à base de règles
[Ferreira et Khesla, 2000]	nbre de véhicules	plan des feux	continu	temps	décentralisé	instantanée	à base de règles
[Camponogara et Kraus Jr., 2003]	files d'attente	plan des feux	discret	temps	décentralisé	instantanée	QL ¹⁴
[Bazzan, 2005]	files d'attente	phases	continu	temps	décentralisé	instantanée	QL
[Domiec et al., 2005]	véhicules	priorités	continu	temps	décentralisé	moyenne	théorie des jeux
[Wiering, 2000]	temps d'attente	plan des feux	discret	matrice de gain	décentralisé	instantanée	théorie des jeux
[Steingrover et al., 2005]	temps d'attente	plan des feux	discret	temps	décentralisé	instantanée	QL
[Da Silva et al., 2006]	nbre de véhicules	plan des feux	discret	temps	décentralisé	instantanée	QL et Prog dyn
[Nunes et Oliveira, 2004]	densité des flux	plan des feux	discret	nbre de véhicules	décentralisé	instantanée	QL DC ¹⁵
[Richter et al., 2007]	nbre de véhicule	plan des feux	continu	taux d'occupation	centralisé	instantanée	QL+AE
[Naumann et Rasche, 1997]	véhicules	plan des feux	discret	nbre véhicules	centralisé	instantanée	NAC ¹⁶
[Balan et Luke, 2006]	historique véhicule	véhicules	indépendant	temps	décentralisé	instantanée	négociation
[Dresner et Stone, 2004]	véhicules	véhicules	indépendant	retard	décentralisé	instantanée	à base de règles
			continu	temps	décentralisé	instantanée	à base de règles

A pp. Avancées

10. Non identifié
11. Temps de traversée des véhicules
12. Programmation dynamique
13. Algorithmes évolutionnaires
14. Q-Learning
15. Apprentissage par renforcement avec détection de contexte
16. Natural Actor Critic

Chapitre 5

Modélisation et simulation

Sommaire

5.1	Choix de modélisation	74
5.1.1	Modélisation du réseau	74
5.1.1.1	Réseau et trafic	75
5.1.1.2	Les véhicules	76
5.1.1.3	Les intersections	77
5.1.2	Autonomie, intelligence et communication	78
5.2	Critères d'optimisation	79
5.2.1	L'énergie	79
5.2.2	Le temps	79
5.3	Simulateur	80
5.3.1	Les simulateurs existants	80
5.3.2	Pourquoi un nouveau simulateur ?	81
5.3.3	Le simulateur développé	82
5.4	Bilan	84

Dans ce chapitre, nous précisons d'abord le problème qui nous intéresse en le modélisant et en expliquant les choix de modélisation. En effet, nous allons traiter un problème d'optimisation, d'où la nécessité d'un modèle incluant une dynamique et des critères d'optimisation. Ensuite nous décrivons le simulateur qui va servir à l'évaluation et la validation expérimentale. Nous présentons, ici, toutes les informations (nos hypothèses) qui vont aider le lecteur à ne pas se poser des questions lors de la lecture des chapitres suivants.

Nous allons expliciter, dans la section 5.1, nos besoins pour la modélisation d'un réseau de transport de véhicules autonomes intelligents. Nous détaillons les modèles pris en compte au niveau du réseau de transport (injections, routes, etc...), des véhicules et des intersections. Ensuite, dans la section 5.2, nous exposons les critères d'optimisation choisis pour l'amélioration des traversées de véhicules. La section 5.3 présente le simulateur développé qui nous permettra d'évaluer nos approches dans les chapitres suivants. Enfin, en section 5.4, nous concluons ce chapitre en dressant un bilan.

5.1 Choix de modélisation

Nous voulons, dans cette deuxième partie de la thèse, étudier le problème de la gestion d'intersections dans un réseau de transport. Nous rappelons que dans ce travail nous ne nous intéressons pas aux véhicules conduits par les humains, mais aux véhicules autonomes.

L'objectif, ici, est de fluidifier d'une manière adaptative le trafic afin d'éviter au maximum les retards et les congestions. Nous voulons dans notre approche éviter de ralentir les véhicules engagés sur les routes, ce qui nous permettra d'économiser du temps et de l'énergie pour les véhicules. Vu que les problèmes de congestion sont situés en particulier au niveau des intersections, nous voulons éviter au maximum de les arrêter à ces endroits.

Dans les sections suivantes, nous détaillons le modèle du réseau routier étudié, ainsi que les besoins en autonomie et en communication de nos véhicules et de nos intersections.

5.1.1 Modélisation du réseau

Comme on l'a précisé dans le chapitre 2, il existe deux façons pour modéliser le trafic : avec des modèles microscopiques qui sont individu-centrés ou avec des modèles macroscopiques qui considèrent le trafic comme un flux dans un graphe. Nous voulons étudier un système réaliste. Ceci nous amène à considérer les modèles microscopiques [Treuil *et al.*, 2008]. Concernant les déplacements des véhicules et le temps, il existe deux façons de les modéliser, soit en les discrétisant, soit en continu. Nous visons un contrôle fin des véhicules, pour nous permettre d'étudier avec précision les temps de traversées et la consommation d'énergie, donc nous adoptons un déplacement continu dans l'espace. Cependant, pour le temps, nous choisissons d'utiliser un pas de temps discret, mais assez fin pour rester réaliste.

Nous décomposons l'étude de notre modèle en trois parties. La première traite la modélisation du réseau, la deuxième concerne la modélisation des véhicules et enfin la troisième s'intéresse aux intersections.

5.1.1.1 Réseau et trafic

Nous nous intéressons dans le cadre du projet InTraDE aux ports maritimes avec des espaces confinés. Nous voulons étudier des approches généralistes et qui traitent toutes les topologies de réseau, mais en particulier nous voulons pouvoir étudier des réseaux avec des routes et des croisements complexes, c'est-à-dire avec des angles différents à chaque intersection, et des longueurs différentes.

Modèle des routes suivies

Nous considérons des routes avec une ou deux voies, c'est à dire un ou deux flux opposés (représentées en figure 5.1).

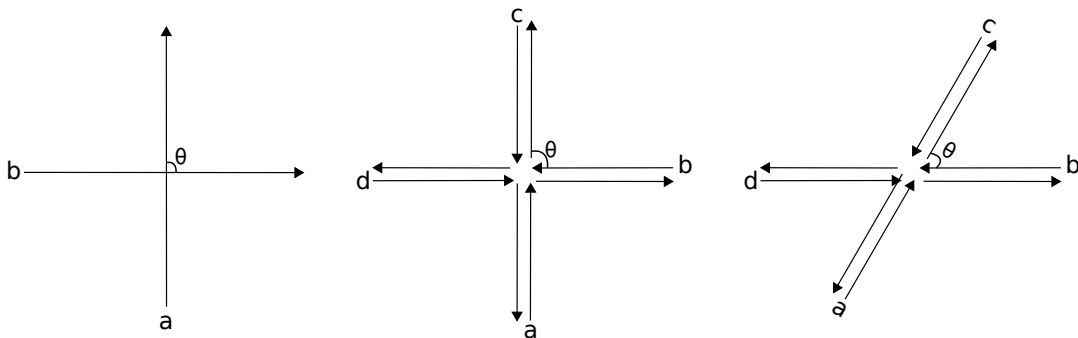


FIGURE 5.1 – Croisements à 2, 4 flux et angle variable

Nous traitons des réseaux routiers modélisés par des *graphe* composés de croisements de routes à une ou deux voies. Les routes peuvent se croiser avec un angle quelconque. Nous supposons aussi que les routes sont assez larges pour faire passer les véhicules.

Nous supposons dans cette thèse que les intersections permettent de traverser une route mais pas d'en changer. En d'autres termes, nous faisons l'hypothèse que nos véhicules vont toujours tout droit aux intersections. Ceci nous permet de simplifier les calculs et de réduire la complexité des algorithmes dans un premier temps, mais dans la discussion finale nous reviendrons sur ce point pour expliquer comment y remédier.

Génération des véhicules

Concernant les injections des véhicules dans le réseau, nous voulons des débits contrôlables randomisés. Pour ce faire, des sources (pour générer) et des puits (pour récupérer les véhicules à la sortie) sont installés aux extrémités de chaque route. Le nombre de

sources et de puits dépend donc du nombre de routes. Chaque source injecte des véhicules suivant une distribution de Poisson. Nous utilisons pour les injections un processus de Bernoulli de paramètre,

$$\lambda = \frac{1}{D}, \quad (5.1)$$

où D est la durée moyenne, en secondes, entre deux injections consécutives. Nous choisissons d'injecter les véhicules si et seulement s'il y a assez d'espace pour les poser (longueur d'un véhicule). Ceux-ci apparaissent dans le réseau à vitesse non nulle (vitesse de croisière), et suivent les routes sur lesquelles ils ont été injectés.

5.1.1.2 Les véhicules

Caractéristiques

Nous considérons, dans notre analyse, que tous les véhicules autonomes sont identiques, d'une largeur l et d'une longueur L , comme le montre la figure 5.2. Dans les algorithmes que nous allons présenter dans les chapitre suivants, les dimension des véhicules jouent un rôle important. En effet le passage par les intersections du réseau va dépendre de la taille des véhicules. Dans ce cas, cette hypothèse simplificatrice n'est certes pas réaliste, mais une solution pour le bon fonctionnement de nos approches dans un cadre réel est que tous les véhicules ont les caractéristiques du pire véhicule (le plus grand).

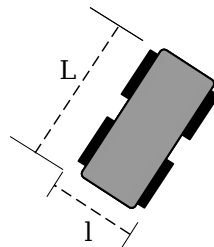


FIGURE 5.2 – Dimension d'un véhicule

- Chaque véhicule a une position donnée (x, y) et une orientation ω .
- La vitesse maximale (V_{Max}) des véhicules est de $10m/s$ ce qui est à peu près de $40km/h$.
- Nous considérons que la vitesse de croisière (v_{crois}) est égale à la vitesse maximale $10m/s$.
- Nous choisissons un modèle à variation de vitesse progressive qui cherche à reproduire un déplacement plus proche du déplacement réel des véhicules. C'est-à-dire, qu'à chaque pas de temps, nous calculons la nouvelle vitesse du véhicule en fonction de son accélération. L'accélération maximale est de $1.5m/s^2$ et la décélération maximale est de $-1.5m/s^2$.

Notre modèle considère que les véhicules se déplacent dans un environnement parfait (pas de glissement ni de frottement). Les informations relatives à chaque véhicule (posi-

tion, orientation et vitesse) sont utilisées afin de calculer leurs déplacements. Les véhicules ne sont donc pas soumis aux bruits de perception et de localisation.

Contrôle longitudinal

Le contrôle longitudinal par défaut calcule une commande de vitesse ou d'accélération longitudinale visant à garder une distance de sécurité entre les véhicules, pour pouvoir freiner en cas d'urgence.

Plusieurs approches de contrôle existent mais nous avons choisi d'utiliser un contrôleur longitudinal (développé dans l'équipe MaIA de l'INRIA) par [Scheuer et al. \[2009\]](#), qui est adapté à nos hypothèses de travail et assure la non-collision entre les véhicules.

Ce contrôleur n'a pas besoin d'infrastructures particulières telles que le GPS ou la communication mais juste des perceptions locales sur le véhicule prédécesseur (s'il existe). Il est conçu pour garantir la non-collision de façon à gérer les incertitudes des perceptions. L'idée de ce contrôleur découle de l'étude d'un pire cas de suivi de deux véhicules, où un premier véhicule avance avec une accélération maximale pendant que son prédécesseur freine au maximum. Sachant que la propriété de non-collision est obtenue quand l'inter-distance entre les deux véhicules est strictement positive, dans ce travail, les auteurs considèrent que l'inter-distance doit être supérieure à une certaine valeur $d_{critique}$ calculée à partir des propriétés des véhicules.

Nous insistons sur le fait que ce contrôleur assure la non-collision uniquement entre les véhicules d'un même flux et entre les intersections, c'est à dire en dehors de la zone de croisement.

5.1.1.3 Les intersections

Comme nous l'avons précisé en section 5.1.1.1, nous supposons dans ce travail que les routes peuvent se croiser avec un angle quelconque. Ces angles sont variables d'une intersection à une autre, et sont compris théoriquement dans 0 et $+\pi$. En réalité, nous ne tenons pas compte des angles très proches de 0 et de π car, dans ces cas, soit les véhicules sont sur la même route soit ils sont en sens opposés, ce qui n'est pas réaliste pour une intersection.

Chaque intersection est caractérisée par le croisement de deux routes qui ont la même largeur. Concernant les longueurs de ces routes au niveau de l'intersection, elles dépendent principalement du rayon d'action de l'intersection. La figure 5.3 illustre un exemple d'intersection de deux routes A et B .

Pour chaque intersection, nous définissons deux seuils mesuré en mètres : un rayon d'action qu'on appellera R (le disque en gris dans la figure 5.3), et un rayon r_0 en dessous duquel les véhicules ne seront plus considérés par l'intersection car ils sont dans la zone de croisement (le cercle en rouge dans la figure 5.3).

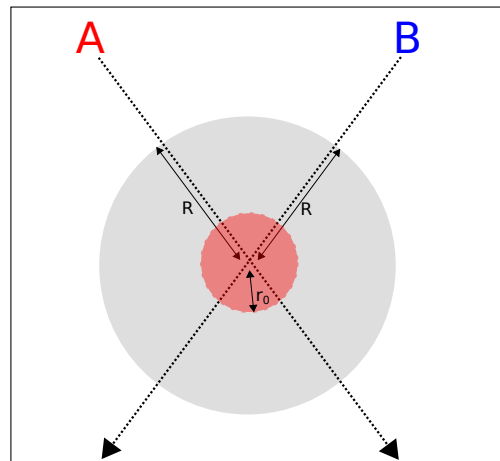


FIGURE 5.3 – Rayon d'action d'une intersection (R) et zone de croisement (r_0)

5.1.2 Autonomie, intelligence et communication

Le trafic routier est par définition distribué et en règle générale exempt de tout contrôle central. Chaque intersection sera en charge de la gestion locale de son trafic, ceci implique que nos intersections doivent avoir connaissance du trafic des véhicules pour s'y adapter en continu. Nous allons présenter, dans cette section, les besoins en autonomie et en communication de nos véhicules et de nos intersections.

Intersections

Les intersections doivent gérer leurs entrées, c'est à dire la détection des véhicules en provenance des intersections voisines. D'autre part, nous considérons des intersections indépendantes, c'est à dire formant un système décentralisé et ceci pour deux raisons simples. La première est que nous voulons que notre système soit robuste : s'il y a un problème au niveau d'une intersection, il n'impliquera pas les autres. La deuxième raison est que nous voulons optimiser les flux de trafic au niveau de chaque intersection par l'intermédiaire des variations des phases, c'est à dire calculer à chaque instant la valeur optimale des phases des intersections. Comme on le verra, pour les alléger, ces calculs doivent être distribués.

Un autre aspect important, dans la modélisation de notre système, est la communication. Après avoir détecté les véhicules entrants, chaque intersection sera en charge de communiquer à ces derniers des consignes pour appliquer l'optimisation déterminée. En plus de cette communication avec les véhicules, nous voulons aussi que chaque intersection communique avec ses intersections voisines. L'idée ici est d'utiliser uniquement les données des intersections voisines pour s'adapter, sachant que cette adaptation se fera à travers des algorithmes de contrôle intelligents. Nous aurons l'occasion de présenter en détail le fonctionnement de ces algorithmes dans les chapitres suivants.

Véhicules

Comme précisé précédemment, nous considérons dans ce qui suit que nos véhicules reçoivent des consignes des intersections aux niveaux des croisements, d'où la nécessité aussi de communiquer avec les intersections à leurs tours.

A part le fait de pouvoir circuler dans le réseau de routier, nos véhicules intelligents autonomes auront aussi besoin d'une capacité de calcul pour mettre en œuvre les consignes fournies. Notre objectif sera de maintenir une vitesse de croisière à l'extérieur des croisements ; ensuite, dès qu'ils arrivent au niveau des intersections, ils seront pris en charge par cette dernière.

5.2 Critères d'optimisation

Dans le projet InTraDE, la réduction de la consommation des véhicules en termes d'énergie est importante. Plus particulièrement, les véhicules autonomes intelligents sont électriques et sont équipés de batteries (moins d'autonomie et plus long à recharger). Pour mesurer les performance de cette nouvelle approche, nous considérons deux critères d'optimisation : l'énergie et le temps.

5.2.1 L'énergie

Nous prenons en compte, comme critère d'optimisation, la variation de l'énergie cinétique seulement à l'accélération. C'est à dire que nous considérons que chaque véhicule qui décélère ne consomme pas d'énergie et nous essayons de minimiser la consommation globale de tous les véhicules en mouvement. Par exemple, pour une accélération entre V_{min} et V_{max} , l'énergie consommée sur un segment $i \rightarrow j$ s'écrit sous cette forme :

$$\Delta E_{c_{acc}} = \frac{1}{2}mV_{max}^2 - \frac{1}{2}mV_{min}^2.$$

5.2.2 Le temps

Nous tenons compte aussi de la performance, en termes de temps de parcours, réalisée par les véhicules, car l'optimisation de l'énergie peut engendrer des congestions et donc des retards. En pratique, nous essayons de minimiser les temps de traversée globaux de tous les véhicules en mouvement.

Remarque : Dans la première partie de la thèse, nous avons considéré le retard comme critère d'optimisation, et non le temps de traversée ; ceci est dû principalement à la nature du problème qui diffère de celui étudié dans cette partie, mais aussi afin de simplifier les calculs. En effet, nous rappelons que le retard est déterminé à partir de la durée de la traversée d'un véhicules et d'une durée calculé théoriquement. Or dans cette partie du travail nous considérons plusieurs routes avec des longueurs différentes.

Enfin, nous insistons sur le fait que les deux critères choisis dans cette partie sont liés. Nous reviendrons sur ce point en chapitre 7.

5.3 Simulateur

Généralement, l'objectif principal d'un simulateur de trafic est de reproduire le comportement de chaque véhicule d'une manière fine, et d'arriver à des simulations les plus réalistes possibles.

5.3.1 Les simulateurs existants

Notre première constatation est qu'il existe un nombre important de simulateurs développés. Nous distinguons deux classes de simulateurs : les simulateurs de flux de trafic conçus pour simuler le trafic en général (plusieurs contextes), et les simulateurs spécialement dédiés aux véhicules autonomes. Notons qu'il n'est pas difficile de passer d'une classe à une autre.

Dans cette section, nous nous sommes inspirés des recherches présentées dans la thèse de [Ksontini \[2013\]](#) et des travaux rencontrés dans la littérature pour présenter brièvement quelques simulateurs microscopiques pertinents.

Simulation de flux de trafic : ARCHISIM est probablement l'un des simulateurs les plus réalistes dans la littérature. Il a été développé par [Espié \[1995\]](#) pour le compte de l'IFSTTAR¹⁷ dans le cadre de la simulation de flux de trafic. Il repose sur une approche comportementale afin de simuler un trafic complexe en se basant sur des travaux en psychologie de la conduite. L'originalité de cette approche réside dans le fait que le simulateur en question permet de simuler des comportements de conduites différents.

Nous avons déjà évoqué SITRAS, qui est un simulateur développé par [Hidas \[2002\]](#) pour la simulation des comportements des conducteurs lors de l'interruption de la circulation sur une voie du réseau routier (section 2.3.1). Il repose sur des modèles de changements de voie (voulus, forcés et coopératifs) pour éviter les accidents. Les modèles en question concernent les flux de véhicules qui sont sur une route composée de plusieurs voies.

MOVSIM [[Treiber et Kesting, 2010](#)] est aussi un simulateur de trafic qui décrit le mouvement de chaque véhicule individuellement. Il s'intéresse aux convois de véhicules. Ses points forts consistent en la mise en place de différents modèles de comportement pour les véhicules suiveurs, en utilisant un modèle propre à lui pour la réduction de la consommation de carburant ou un autre modèle pour le changement de voie.

SUMO [[Behrisch et al., 2011](#)] est un simulateur de trafic utilisé dans plusieurs travaux de recherche pour la validation des approches. Parmi ces travaux nous trouvons principalement : le choix de l'itinéraire, les algorithmes pour les feux de circulation, les com-

17. l'Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux

munications avec les véhicules, la simulation des stratégies de conduite, et la gestion de la circulation automatiques.

Dans le cadre du projet InTraDE, nous disposons de l’outil de simulation SCANeR-studio¹⁸, qui est développé par l’entreprise OKTAL, un des partenaires du projet. Nous l’avons déjà utilisé dans la première partie de cette thèse et avons constaté les difficultés d’intégration des algorithmes (logiciel propriétaire). De plus, cet environnement de simulation nécessite de grandes capacités de calcul, surtout lors des simulations du trafic à grande échelle (plus d’un millier de véhicules).

TrafficGen [Bonhomme *et al.*, 2014] introduit en section 2.3.1 est un outil expérimental de simulation microscopique, en cours de développement, conçu afin de comparer plusieurs approches liées au trafic. Il est présenté comme un outil d’aide à la décision permettant une simple mise à jour continue des données, ainsi qu’une révision facile des modèles. Pour l’instant, il ne permet pas la gestion des intersections.

Il existe aussi d’autres simulateurs commerciaux très puissants tels que VISSIM, AIM-SUN ou PARAMICS qui offrent des environnements de simulation et de test produisant presque toutes les mesures d’efficacité couramment utilisées dans le domaine de l’ingénierie du trafic.

Simulation de véhicules autonomes : VIVUS [Gechter *et al.*, 2012] est aussi un simulateur 3D réaliste de véhicules autonomes dédié à la simulation de convois de véhicules (platooning) mais pas au trafic en général.

Dresner et Stone [2004, 2005] proposent aussi un simulateur de réseau dédié exclusivement à la simulation des véhicules intelligents autonomes appelé AIM¹⁹ (Autonomous Intersection Management). Nous avons détaillé le fonctionnement de ce simulateur dans le cadre de l’approche par réservations dédiée aux véhicules intelligents autonomes en section 4.5.3.

5.3.2 Pourquoi un nouveau simulateur ?

Afin d’évaluer les performances de nos algorithmes, nous avons décidé de développer notre propre outil de simulation. Nous justifions dans cette section ce choix.

A notre connaissance il n’existe pas de simulateur microscopique adapté aux besoins décrits en section 5.1. Nous avons commencé par une étude avec des simulateurs existants, mais le premier obstacle rencontré vient de la particularité de nos croisements des routes. A notre connaissance, peu de simulateurs peuvent modéliser des intersections avec un angle quelconque. En plus, très peu d’entre eux ont une licence libre.

D’autre part, il est délicat de reprendre les simulateurs issus d’autres recherches, pour les modifier à notre guise, surtout quand on parle de système complexe tel que le trafic. En plus, nous voulons simuler un grand nombre de véhicules, et les simulateurs les plus fins

18. <http://www.scanersimulation.com/software/software-research.html>

19. <http://www.cs.utexas.edu/~aim/>

(physique des véhicules), tels que ARCHISIM, MOVSIM ou VIVUS, sont tous coûteux en temps de calcul.

Le simulateur AIM est le simulateur le plus proche de nos besoins. Cependant, nous n'avons ni le même modèle de routes (routes droites), ni le même modèle d'intersections (angle fixe à 90°), ni le même modèle de véhicules (communications, autonomie, et perceptions).

5.3.3 Le simulateur développé

Nous avons développé (en java) un simulateur à espace continu et à temps discret d'un réseau de routes (la figure 5.4 est une capture d'écran).

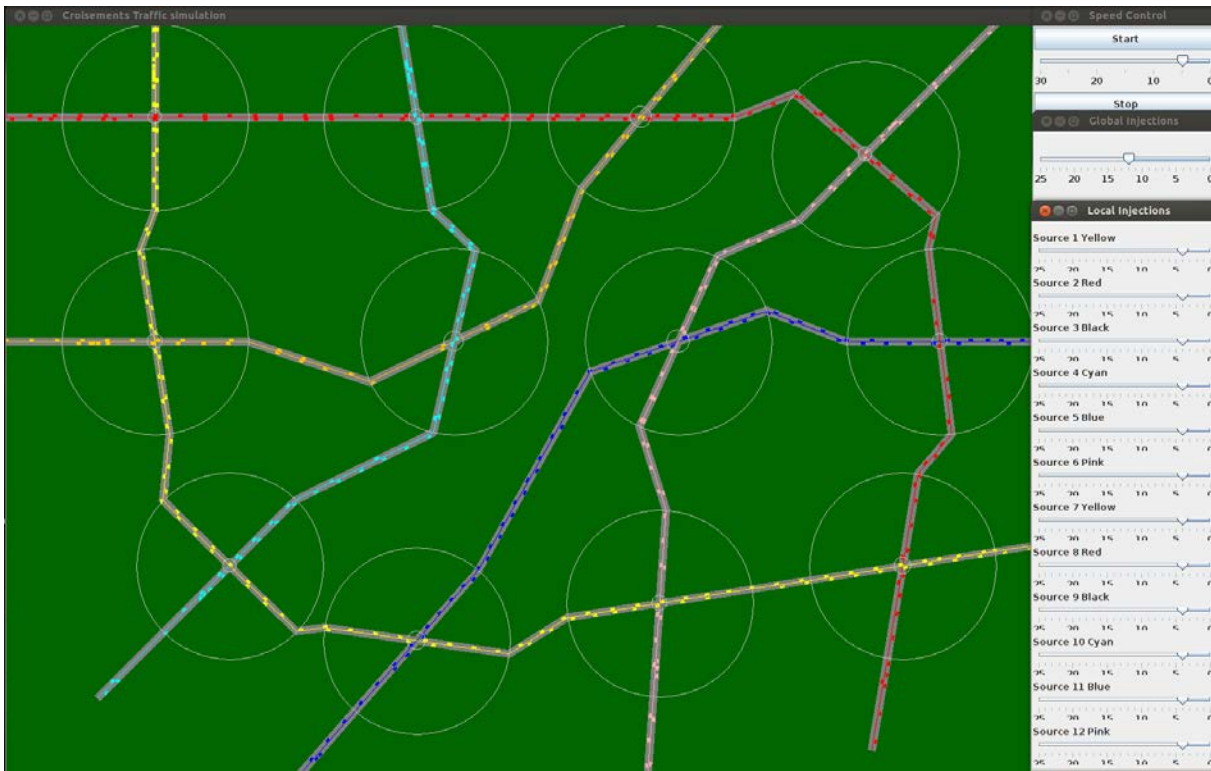


FIGURE 5.4 – Le simulateur développé avec un réseau de 12 intersections

Notre simulateur fonctionne de manière asynchrone, c'est-à-dire qu'à chaque pas de temps, chaque véhicule puis chaque intersection exécute une tâche. Ce pas de temps est relativement faible, de l'ordre de 0,05s. Pour l'exécution des tâches et afin d'éviter certains conflits d'espace (au sein d'une voie) entre les véhicules, nous considérons toujours ces derniers dans l'ordre d'entrée dans le réseau.

Nous avons développé une interface pour visualiser le réseau de trafic (figure 5.4), qu'on peut désactiver, par exemple pour accélérer les simulations et les calculs. L'interface aussi offre la possibilité d'accélérer la simulation ou de modifier les débits de chaque sources d'injection des véhicules.

Le simulateur est composé de quatre modules, comme le montre la figure 5.5, qui se lancent dans l'ordre suivant :

1. le réseau (construction),
2. les intersections,
3. les véhicules, et
4. la simulation.

Trois entrées sont exigées afin de pouvoir simuler un réseau de transport : (i) la configuration du réseau, c'est-à-dire la position des intersections et l'ensemble des routes qui passent par ces dernières, (ii) les débits d'entrée au cours du temps pour chaque source du réseau, et (iii) la stratégie employée par les intersections pour gérer les traversées des véhicules (par exemple feux de signalisation).

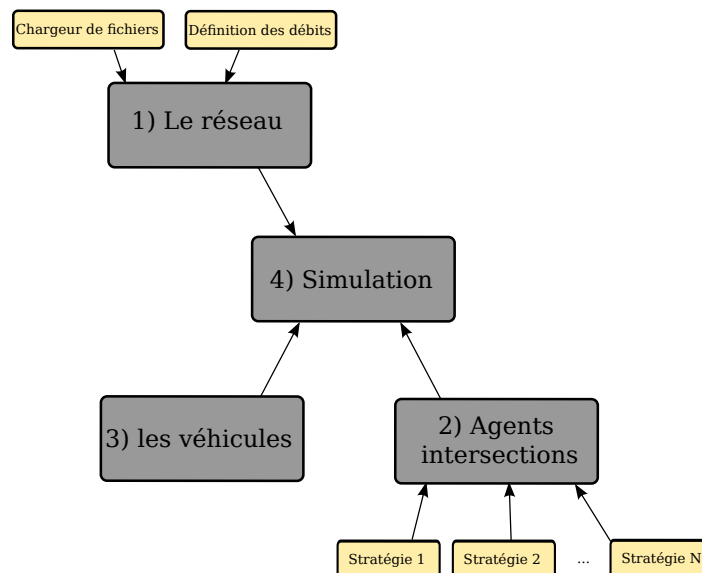


FIGURE 5.5 – Architecture du simulateur (modules et dépendances)

Concernant la construction du réseau, nous avons développé une méthode pour la configuration et la lecture du réseau. En effet, l'utilisateur a besoin de décrire le réseau à étudier sous forme d'un fichier *XML* qui est composé de deux parties. La première concerne la position des intersections, décrite comme dans cet exemple :

```

<Agents>
  <Object>
    lasttrafficsimulation.Agent
  <Methods>
    setX:=290
    setY:=390
  <\Object>
<\Agents>

```

où $setX$ et $setY$ sont deux paramètres qui vont définir la position d'une intersection. Ensuite vient une deuxième partie qui concerne la définition des routes, comme dans l'exemple suivant :

```
<Routes>
  <Object>
    lasttrafficsimulation.Route
    <Methods>
      setRoute:=1
      setRoute:=2
      setRoute:=4
      setTitle:=Paris
  <\Object>
<\Routes>
```

qui spécifie que la route *Paris* passe par la première intersection, puis la deuxième, et enfin la quatrième, dans l'ordre de leur apparition. Un exemple d'un fichier *XML* complet, définissant un réseau composé de cinq intersections et de trois routes, est illustré en annexe [B](#).

Nous avons intégré dans le simulateur la possibilité de lecture de scénario en fonction des débits et du temps, par exemple dans le cas où nous voulons analyser une certaine configuration des débits d'entrée pour une heure, ensuite une autre pendant deux heures.

L'utilisation de plusieurs comportements (stratégies) pour les véhicules et les intersections est prise en compte. Afin d'analyser ces stratégies, le simulateur renvoie des statistiques tels que : les temps de trajet par véhicule, la consommation d'énergie, les paramètres de chaque intersection comme l'évolution des phases, etc.

5.4 Bilan

Les différents besoins de modélisation présentés dans ce chapitre donnent un aperçu sur notre façon d'attaquer le problème de gestion des flux de véhicules autonomes dans les réseaux de transport. Nous avons commencé par exprimer nos besoins, tout en présentant nos choix de modélisation. Ces besoins ont été exprimés de différents points de vues, en premier celui du réseau de transport, ensuite celui des intersections et des véhicules.

Nous avons dédié la deuxième partie de ce chapitre aux simulateurs existants, ensuite nous avons justifié notre choix de conception d'un nouvel outil. Les simulateurs qui peuvent modéliser des intersections avec un angle quelconque sont complexes et délicats à reprendre. Enfin, nous avons présenté ce nouvel outil que nous avons utilisé à des fins d'expérimentation dans le reste de cette thèse.

Dans les chapitres suivants, nous présentons d'abord une approche originale pour l'optimisation des flux de trafic au niveau de chaque intersection du réseau, avant de s'intéresser à la synchronisation des intersections pour optimiser les flux au niveau du réseau.

Chapitre 6

Synchronisation de flux de véhicules dans une intersection

Sommaire

6.1	Synchronisation d'un croisement	86
6.1.1	Approche par synchronisation locale	86
6.1.2	Période minimale de croisement	87
6.1.2.1	Cas $\theta = 90^\circ$	87
6.1.2.2	Cas $\theta < 90^\circ$	88
6.1.2.3	Cas $\theta > 90^\circ$	89
6.1.3	Synchronisation temporelle	90
6.1.4	Routes à deux voies opposées	92
6.1.5	Adaptation des vitesses	94
6.1.6	Rayons de la zone de contrôle	96
6.2	Expérimentations	97
6.2.1	Simulations	98
6.2.2	Comparaison des stratégies	98
6.2.3	Effet de variation du paramètre R (rayon d'action)	100
6.2.4	Effet de variation du paramètre D (débit)	100
6.3	Synchronisation n-véhicules/p-véhicules	100
6.4	Conclusion	102

Dans ce chapitre, nous examinons la possibilité de croiser des flux de véhicules autonomes sans les arrêter, afin d'éviter les congestions (retards) et les pertes d'énergies. Pour cela, nous définissons des mécanismes locaux permettant de suffisamment espacer les véhicules et de synchroniser leur passage. Un premier travail avait été mené dans le cadre du stage de Master de [Albouze \[2010\]](#), dans l'équipe MaIA, montrant la faisabilité du croisement alterné sans arrêt de deux routes avec un angle de 90° . Ce chapitre explore et étend cette idée.

Pour mettre en place un croisement sans arrêts, il faut examiner les paramètres spatiaux d'une intersection (taille, angle entre les routes) qui influencent la synchronisation temporelle. A partir de cette étude, nous définissons des agents de contrôle qui déterminent la vitesse des véhicules qui arrivent à une intersection pour assurer l'alternance entre les flux. Nous proposons un contrôle aux intersections basé sur la synchronisation temporelle des véhicules et nous décentralisons le problème en gérant indépendamment chaque intersection.

En section [6.1](#) nous présentons le principe de la synchronisation temporelle de deux routes à une intersection. Nous expliquons (i) comment calculer la période minimale entre deux véhicules qui se suivent et (ii) l'algorithme utilisé par l'agent de contrôle. Ensuite, la section [6.2](#) est consacrée à l'étude expérimentale en simulation de notre modèle comparé à une solution de l'état de l'art. En section [6.3](#) nous discutons la possibilité d'alterner n -véhicules d'un côté et p -véhicules de l'autre, et enfin en section [6.4](#) nous dressons un bilan et présentons les perspectives de ce travail.

6.1 Synchronisation d'un croisement

6.1.1 Approche par synchronisation locale

Nous introduisons dans cette section notre approche. Avant d'en préciser les détails, nous rappelons que nous traitons des réseaux routiers composés de croisements de routes à une ou deux voies. Les routes peuvent se croiser avec un angle quelconque. Nous considérons d'abord le cas quand $\theta = 90^\circ$, puis généralisons.

Notre approche repose sur la synchronisation locale des véhicules, c'est à dire au niveau de chaque intersection. Elle consiste à faire passer de façon alternée et sans arrêt des véhicules de chaque route en respectant une inter-distance suffisante afin d'éviter les collisions.

La gestion des croisements est réalisée par un agent, dit agent de contrôle, en chaque intersection. Son rôle principal est de communiquer avec les véhicules afin de gérer leur passage. Cet agent dispose d'un rayon d'action limité qui définit la distance en dessous de laquelle il commence à communiquer avec les véhicules qui s'approchent de l'intersection.

L'agent de contrôle régule les vitesses des véhicules à l'intérieur de son rayon d'action. Nous détaillons dans ce qui suit comment déterminer la période d'alternance entre les

routes et comment l'agent, en s'appuyant sur cette période, contrôle les véhicules qui s'approchent de son intersection.

6.1.2 Période minimale de croisement

Nous allons déterminer, dans cette section, les conditions permettant le croisement en alternance de deux flux de véhicules (sur deux routes à une voie). Dans notre analyse, nous considérons que tous les véhicules sont autonomes et identiques. Ils ont une largeur l , une longueur L , et tous les véhicules roulent par défaut à une vitesse V .

Dans un premier temps, on s'intéresse à la synchronisation de deux flux A et B pour permettre leur croisement sans arrêt. Nous cherchons à déterminer le temps minimal entre deux véhicules du flux A , noté T_{min} , permettant à un véhicule du flux B de passer. Nous distinguons trois cas selon l'angle de croisement θ entre deux routes : $\theta = 90^\circ$, $\theta < 90^\circ$ et $\theta > 90^\circ$.

6.1.2.1 Cas $\theta = 90^\circ$

La zone de croisement est définie par le carré de côté l correspondant à l'espace partagé par les deux routes, comme illustré dans la figure 6.1.

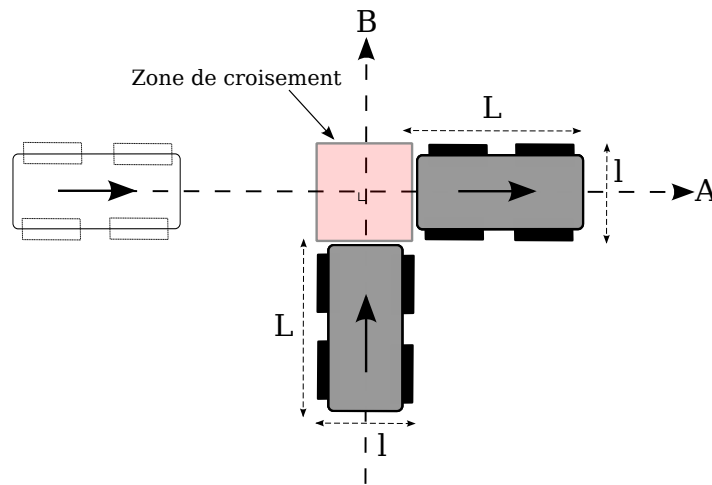


FIGURE 6.1 – Zone de croisement de deux flux avec un angle $\theta = 90^\circ$

Pour qu'un véhicule traverse complètement cette zone il lui faut un temps $(l + L)/V$. On en déduit la période minimale de passage de deux véhicules successifs (un de chaque flux) :

$$T = 2(L + l)/V. \quad (6.1)$$

L'inter-distance entre deux véhicules de la même file doit donc être au minimum $D_{min} = 2(L + l) - L = L + 2l$ (on soustrait la longueur d'un véhicule).

6.1.2.2 Cas $\theta < 90^\circ$

Dans ce cas de figure, un véhicule peut entrer dans la zone de croisement avant que l'autre ne soit complètement sorti (voir figure 6.2-c). La zone de croisement doit donc être partagée par deux véhicules, un de chaque file (flux).

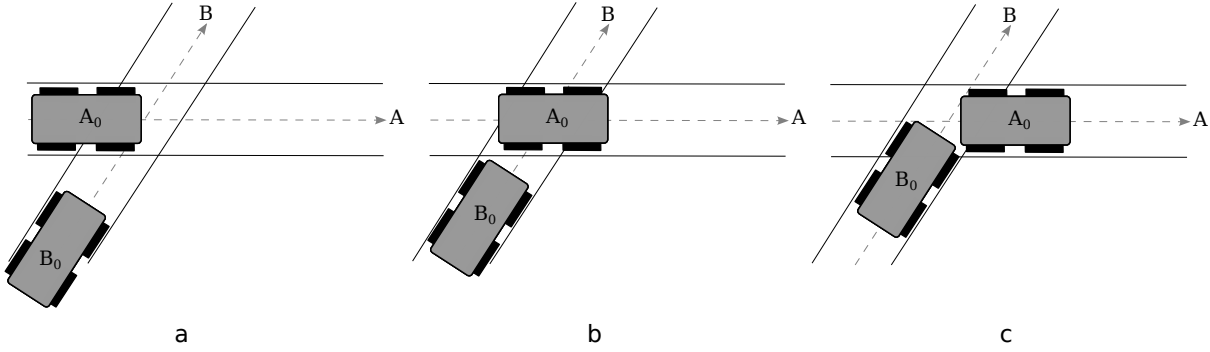


FIGURE 6.2 – Zone de partage

L'ordre de passage est $A_0, B_0, A_1, B_1, A_2, B_2, \dots$. Les figures 6.2 a, b, et c sont une animation. L'instant c est le moment où les véhicules sont tangents. Pour chaque véhicule on va spécifier deux instants :

- quand un véhicule commence à sortir de la zone de croisement (le véhicule A_0 dans la figure 6.2-c),
- et quand un véhicule continue à entrer dans la zone de croisement (le véhicule B_0 dans la figure 6.2-c).

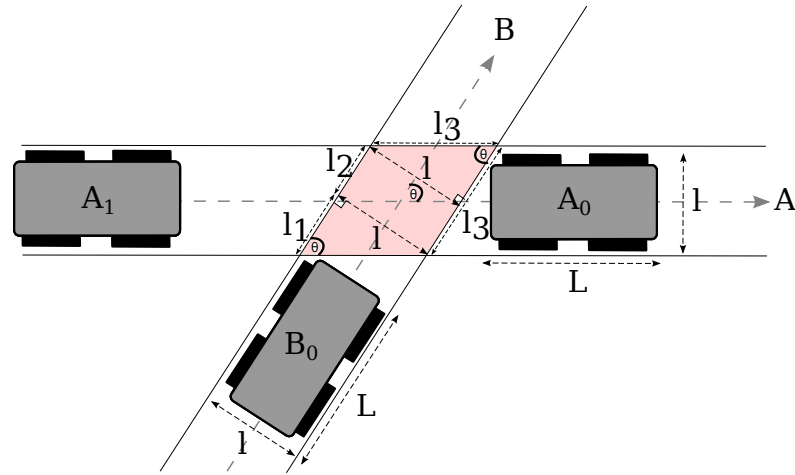
A partir de ces deux cas, en nous appuyant sur la figure 6.3 (qui introduit les distances l_1, l_2, l_3), nous déterminons la période et l'inter-distance nécessaire à un croisement. Considérons l'évolution du scénario :

- à $t = 0$, A_0 sortant, B_0 entrant (figure 6.2-c) ;
- à $t = (l_2 + L)/V$, B_0 sortant, A_1 entrant ;
- à $t = [(l_2 + L) + (l_2 + L)]/V$, A_1 sortant, B_1 entrant ;
- ...

D'après la figure 6.3, on peut déterminer $l_1 = \frac{l}{\tan \theta}$ et $l_3 = \frac{l}{\sin \theta}$, d'où : $l_2 = l_3 - l_1 = \frac{l}{\sin \theta} - \frac{l}{\tan \theta}$. On peut donc écrire la période sous la forme :

$$\begin{aligned} T &= 2 \left(\frac{l}{\sin \theta} - \frac{l}{\tan \theta} + L \right) / V \\ &= 2 \left(l \frac{(1 - \cos \theta)}{\sin \theta} + L \right) / V. \end{aligned} \quad (6.2)$$

Afin de vérifier le résultat de cette formule, considérons les deux cas limites quand $\theta \rightarrow 0$ et $\theta \rightarrow \frac{\pi}{2}$. Si $\theta \rightarrow 0$, $\frac{(1 - \cos \theta)}{\sin \theta}$ tend vers 0 donc la période T tend vers $2L/V$, ce qui est logique car les deux files roulent dans le même sens. Si $\theta \rightarrow \frac{\pi}{2}$, $\frac{(1 - \cos \theta)}{\sin \theta}$ tend vers 1 donc la période T tend vers $2(l + L)/V$ comme dans la formule 6.1.

FIGURE 6.3 – Croisement de deux flux avec un angle $\theta < 90^\circ$

6.1.2.3 Cas $\theta > 90^\circ$

L'ordre de passage est toujours $A_0, B_0, A_1, B_1, A_2, B_2, \dots$ (figure 6.4). Quand $\theta > 90^\circ$, nous n'avons plus la possibilité de partager la zone de croisement entre les deux flux. Il faut qu'un véhicule A ou B soit complètement sorti avant qu'un autre commence à entrer. Pour chaque véhicule on va spécifier deux situations, comme dans le cas précédent :

- quand un véhicule sort de la zone de croisement,
- et quand un véhicule entre dans la zone de croisement.

En nous appuyant sur la figure 6.4 (qui introduit d'autres distances l_1, l_2, l_3), nous déterminons la période et l'inter-distance nécessaires au croisement. Considérons l'évolution du scénario :

- à $t = 0$, A_0 sort, B_0 entre ;
- à $t = (l_3 + l_2 + L)/V$, B_0 sort, A_1 entre ;
- à $t = [(l_3 + l_2 + L) + (l_3 + l_2 + L)]/V$, A_1 sort, B_1 entre ;
- ...

D'après la figure 6.4, on peut déterminer : $l_3 = \frac{l}{\sin \pi - \theta} = \frac{l}{\sin \theta}$ et $l_2 = l_3 |\cos(\pi - \theta)| = l_3 |\cos(\theta)| = \frac{l}{\tan \theta}$. On peut donc écrire la période sous la forme :

$$\begin{aligned} T &= 2 \left(\frac{l}{\sin \theta} + \frac{l}{\tan \theta} + L \right) / V \\ &= 2 \left(l \frac{(1 + \cos \theta)}{\sin \theta} + L \right) / V. \end{aligned} \quad (6.3)$$

Afin de vérifier le résultat de cette formule, considérons les deux cas limites quand $\theta \rightarrow \frac{\pi}{2}$ et $\theta \rightarrow \pi$. Si $\theta \rightarrow \frac{\pi}{2}$, $\frac{(1 + \cos \theta)}{\sin \theta}$ tend vers 1 donc la période T tend vers $2(l + L)/V$ comme dans la formule 6.1. Si $\theta \rightarrow \pi$, $\frac{(1 - \cos \theta)}{\sin \theta}$ tend vers ∞ ce qui est logique car les deux files ne peuvent pas rouler dans des sens contraires.

En pratique pour tout θ , T sera augmentée de ϵ une marge de sécurité d'inter-distance entre véhicules.

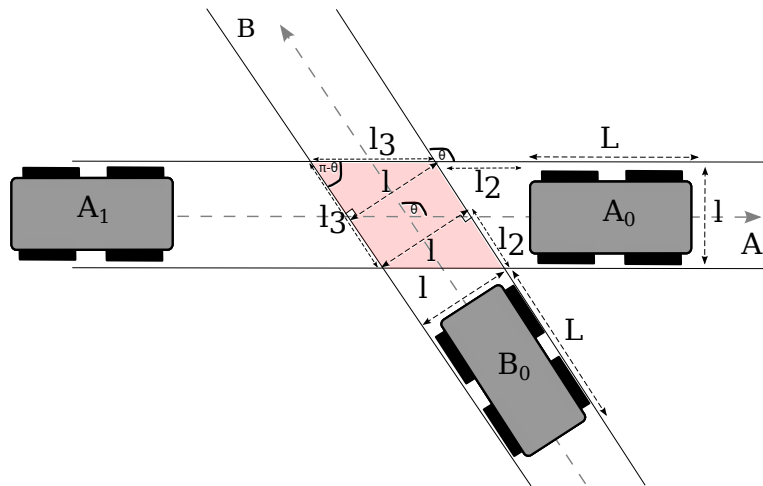


FIGURE 6.4 – Croisement de deux flux avec un angle $\theta > 90^\circ$

6.1.3 Synchronisation temporelle

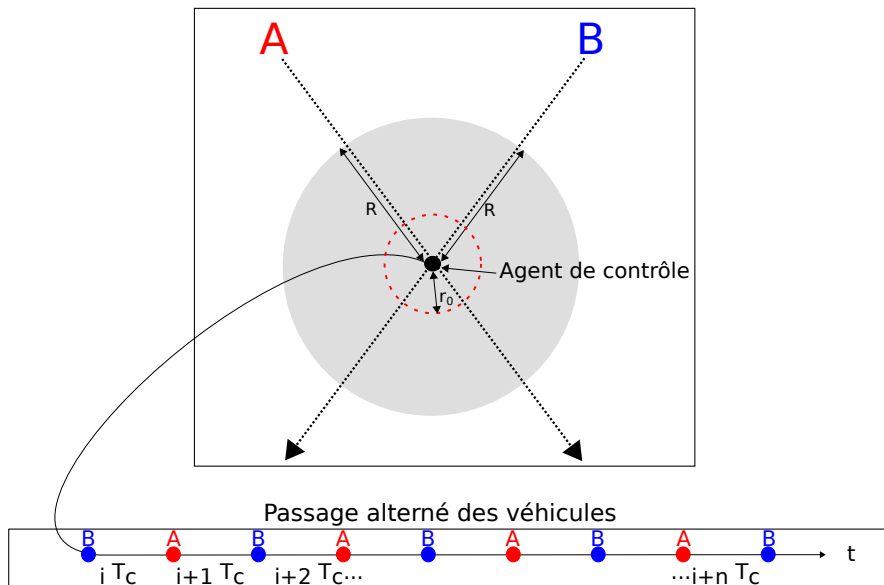


FIGURE 6.5 – Principe de synchronisation d'une intersection

Pour ne pas arrêter les flux, notre approche consiste à faire passer les véhicules alternativement entre les deux routes, ce qui nous amène à organiser le passage des véhicules toutes les demi-périodes $T_c = T/2$. Les véhicules d'un flux passeront aux multiples pairs de T_c et les véhicules de l'autre flux aux multiples impairs, en minimisant les changements de vitesse pour réduire la consommation d'énergie. La figure 6.5 illustre ce principe de passage alterné de période T_c .

L'agent de contrôle est chargé de calculer les consignes de vitesse des véhicules quand ils sont dans son rayon d'action R , c'est à dire à une distance comprise entre r_0 et R , r_0

étant la distance à partir de laquelle le véhicule sera trop près du croisement pour recevoir une modification de vitesse ($0 < r_0 < R$) et devra rouler à la vitesse de croisière.

Chaque flux provenant de chaque route peut être contrôlé indépendamment des autres. L'algorithme 6.1 donne le contrôle d'un flux par l'agent connaissant la parité $p \in [0, 1]$ de ce flux. L'algorithme est exécuté à fréquence régulière (tous les pas de temps dans la simulation).

Algorithme 6.1 : Contrôle d'un flux par l'agent de contrôle

Entrées : parité du flux $p \in [0, 1]$, pas de simulation actuel t

```

1  $last = 0$ ;
2 pour chaque véhicule  $i$  perçu à une distance  $r_i \in ]r_0; R[$ 
3 (pris dans l'ordre croissant de leur distance) faire
4   Percevoir  $r_i, V_i$ ;
5   /* ( $n_i$  est enregistré dans la mémoire de l'agent) */
6   si  $n_i = -1$  /*  $i$  vient d'entrer dans la zone contrôlée */
7   alors
8      $t_a = t + r_i/V_i$ ;
9      $n_i = \lceil t_a/T - p/2 \rceil$ ;
10    si  $n_i \leq last$  alors  $n_i = last + 1$ ;
11     $last = n_i$ ;
12     $t_p = n_i * T + p * T_c - r_0/v_{crois}$ ;
13    Envoyer le profil d'accélération au véhicule  $i$ ;

```

L'algorithme traite toujours les véhicules du plus proche de l'intersection au plus éloigné pour assurer une affectation respectant le même ordre de passage. L'agent de contrôle a besoin de percevoir leur distance à l'intersection (r_i) et leur vitesse (V_i).

Considérons un seul véhicule i ; le calcul de l'instant de passage par le centre de l'intersection est donné par la ligne 7 (en supposant sa vitesse V_i). Nous faisons le choix de n'autoriser que des ralentissements, alors le véhicule devra passer par l'intersection au moment $(2n_i - p)T$, où n_i est un entier qui correspond au plus proche multiple de parité p de T (calculé comme dans la ligne 8).

L'agent de contrôle doit veiller à ce que cette valeur n'ait pas été déjà attribuée à un autre véhicule. À cette fin, la variable $last$ enregistre la dernière valeur attribuée à un véhicule et un test est effectué (ligne 9) pour déterminer s'il faut utiliser la valeur calculée ou $last + 1$.

L'agent calcule n_i une seule fois à l'entrée du véhicule dans son rayon d'action. Dans le cas contraire, à cause des variations de vitesse, la valeur n_i pourrait changer d'une itération à une autre. Ceci est obtenu en affectant la valeur $n_i = -1$ aux véhicules en dehors des rayons d'action des agents. L'agent de contrôle stocke n_i d'une itération à une autre et ne calcule n_i que si et seulement si $n_i = -1$ (ligne 5).

Une fois n_i connu, l'agent calcule la date de passage t_p du véhicule i à la distance r_0 du centre d'intersection (ligne 11), où $n_i * T + p * T_c$ est l'instant d'arrivée à l'intersection

en fonction du flux et r_0/v_{crois} le temps nécessaire pour parcourir la distance r_0 à vitesse de croisière. Puis en conséquence, il détermine le profil d'accélération à appliquer jusqu'à $r_i = r_0$. Nous détaillerons en section 6.1.5 comment l'agent calcule ce profil d'accélération.

6.1.4 Routes à deux voies opposées

Dans cette section, nous allons montrer comment utiliser la solution présentée dans les sections précédentes dans le cas de deux routes composées chacune de deux voies opposées.

Considérons deux routes qui se croisent avec deux voies opposées chacune ((A, C) et (B, D)). Soit λ la distance qui sépare deux voies opposées de la même route (figure 6.6). Les deux voies opposées d'une route donnée devraient traverser l'intersection en même temps, de sorte que l'ordre de passage soit $(A_0, C_0), (B_0, D_0), (A_1, C_1), (B_1, D_1)...$

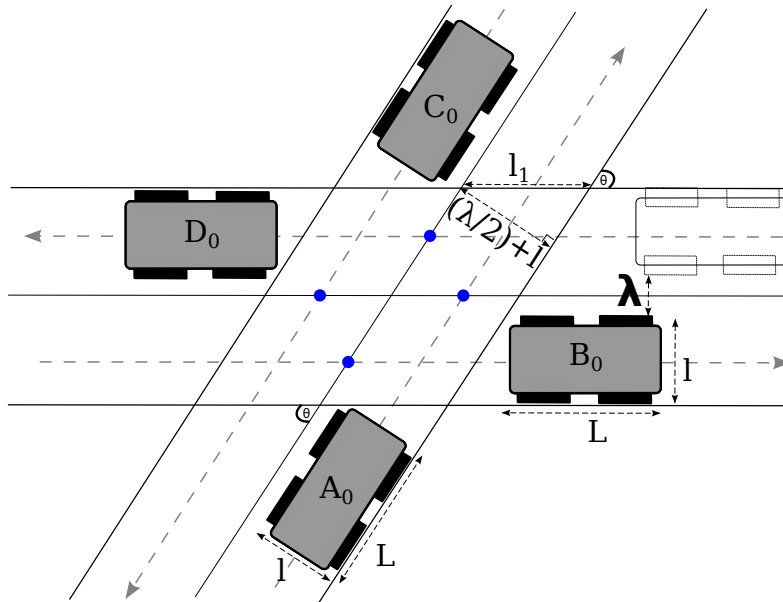


FIGURE 6.6 – Zone de croisement de deux routes avec 4 flux

La zone de croisement est maintenant définie par le losange correspondant à l'espace partagé par les deux routes (voir figure 6.6). L'algorithme de l'agent de contrôle reste inchangé, chaque voie étant traitée indépendamment des autres voies, et chaque voie ayant sa propre variable *last*. Il faut juste faire attention au fait que, pour une voie donnée, les distances ne sont plus calculées par rapport au centre de la zone de croisement, mais par rapport au milieu du segment de la voie à l'intérieur de la zone de croisement, comme illustré par les "points" sur la figure 6.6.

Prenons l'exemple de la figure 6.7 qui illustre, avec quatre figures d'une intersection, une période complète de croisement entre les deux routes (A, C figure 6.7-1 et B, D figure 6.7-3). Nous divisons cette période en quatre instants clés (les figures 6.7-1, 2, 3 et 4).

Pour traverser et sortir complètement de la zone de croisement, un véhicule prend un

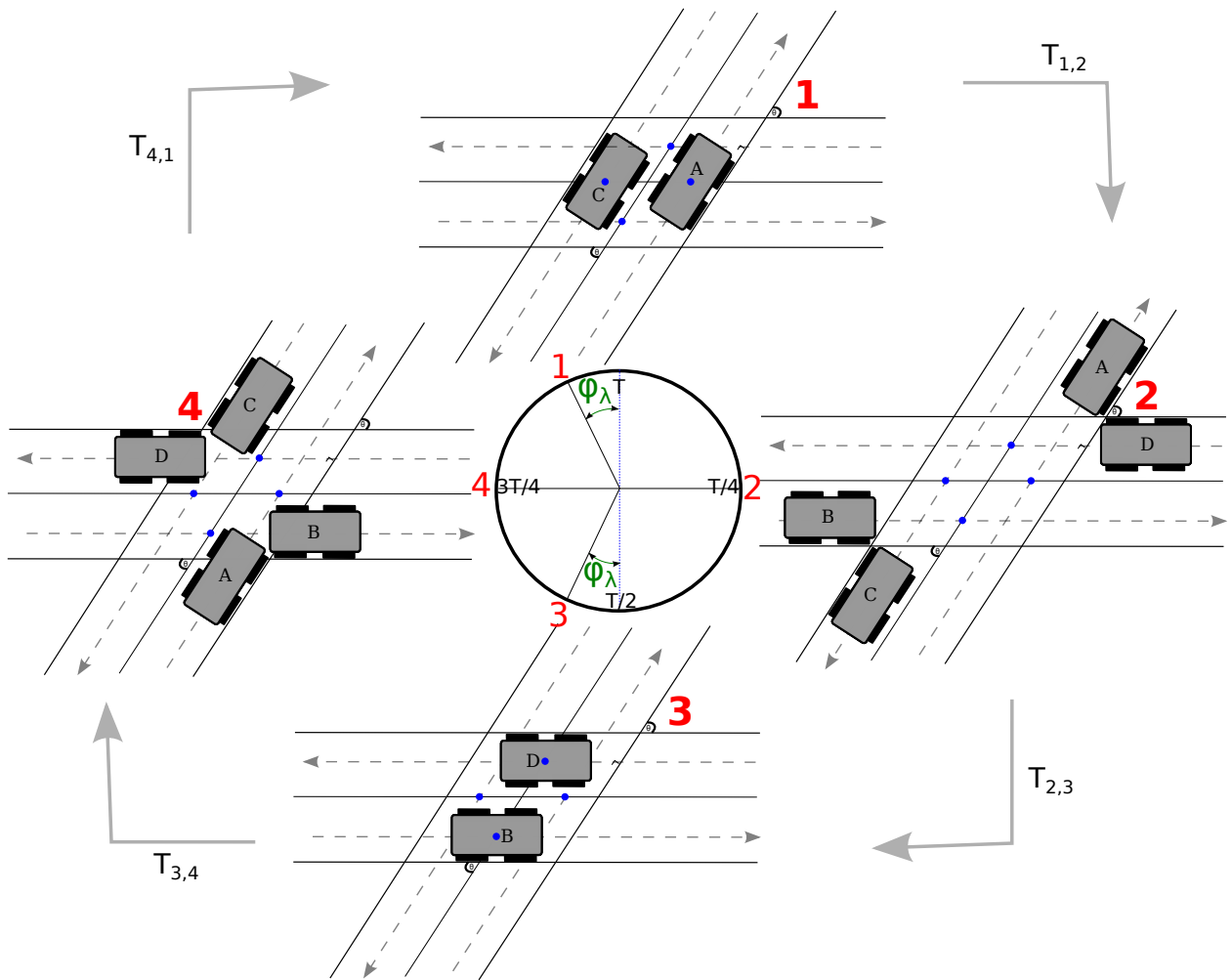


FIGURE 6.7 – Animation de croisement de quatre véhicules

temps $(2l_1 + L)/V$. Sur la figure 6.7, cette demi-période se calcule aisément en évaluant le temps entre les instants 2 et 4 (ou, inversement, 4 et 2).

D'après la figure 6.6, nous pouvons déterminer $l_1 = \frac{l + \frac{\lambda}{2}}{\sin \theta}$. Donc la période de passage minimale des deux routes est :

$$T = 2 \left(\frac{2(l + \frac{\lambda}{2})}{\sin \theta} + L \right) / V. \quad (6.4)$$

Après avoir calculé la période de passage, nous nous intéressons maintenant aux points de référence de chaque voie qui posent une contrainte supplémentaire par rapport à des routes à une seule voie. A cause de la distance entre les points de référence et l'angle de l'intersection, nous nous retrouvons avec des temps $T_{1,2} = T_{2,3}$, $T_{3,4} = T_{4,1}$, et $T_{1,2} \neq T_{3,4}$, ce qui implique un léger déphasage entre les deux routes de l'intersection qu'on appellera ϕ_λ (voir le cercle temporel au milieu de la figure 6.7). En effet, ϕ_λ est la différence entre le quart de la période ($\frac{T}{4}$) et l'instant $T_{4,1}$. Si nous considérons le véhicule C comme véhicule

de référence, d'après la figure 6.8, nous avons $T_{4,1} = (l_1 + \frac{l}{2} - h)/V$ sachant que $h = \frac{l}{2 \tan \theta}$, et $\frac{T}{4} = (l_1 + \frac{l}{2})/V$ d'où :

$$\phi_\lambda = (\frac{l}{2 \tan(\theta)})/V.$$

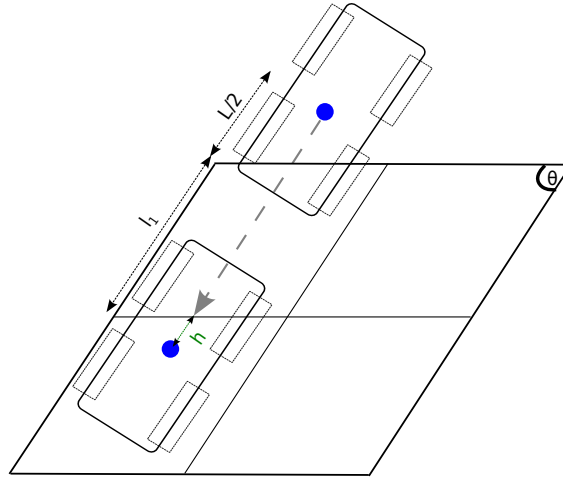


FIGURE 6.8 – Temps nécessaire pour passer de la figure 6.7-4 à la figure 6.7-1

Donc pour remédier à ce problème, il faut tenir compte de ces déphasages dans l'algorithme lors du calcul des instants de passage, pour que la phase $\phi_{B,D}$ congrue à $\phi_{A,C} + \frac{T}{2} - 2\phi_\lambda[T]$.

6.1.5 Adaptation des vitesses

L'algorithme 6.1 permet de gérer les flux des véhicules uniquement à l'arrivée, dès qu'un véhicule franchit R . Dans cette section, nous allons étudier comment l'agent de contrôle de chaque intersection détermine le profil d'accélération de ces derniers.

Nous rappelons que l'agent de contrôle d'une intersection détermine l'instant de passage du véhicule à travers l'intersection au moment $t_p = n_i * T + p * T_c - r_0 / v_{crois}$ où $n_i = \lceil t_a / T - p / 2 \rceil$ (lignes 8 et 11 de l'algorithme 6.1). Nous avons choisi, dans ce travail, de ralentir les véhicules. Dans le cas où nous voulons accélérer, il suffit de choisir $n_i = \lfloor t_a / T - p / 2 \rfloor$. Si nous faisons abstraction de l'accélération et la décélération entre les intersections, la nouvelle vitesse à appliquer sera simplement $v_x = \frac{r_i}{t_p - t}$. Cependant, nous considérons que l'accélération est continue.

Nous faisons le choix d'un profil d'accélération tel que la figure 6.9 qui illustre un exemple pour un véhicule i détecté entre R et r_0 . Quand un véhicule franchit le rayon R , l'agent de contrôle le perçoit à r_i avec une vitesse v_i . En fonction de ses capacités d'accélération et de décélération (a_1 et a_2 dans la figure 6.9), il doit adapter sa vitesse pour atteindre v_x puis passer par r_0 à vitesse de croisière. En effet, chaque véhicule détecté dans la zone de contrôle, reçoit quatre contraintes temporelles à respecter :

- δt_{a_1} est la durée de décélération afin d'atteindre la vitesse v_x ,
- δt_{a_2} est la durée d'accélération afin d'atteindre la vitesse v_{crois} ,
- un "plateau" de distance d à vitesse constante, et
- δt_1 et δt_2 sont deux durées additionnelles fixes de sécurité en fonction de la distance $(l + \frac{\lambda}{2})$ qui permettent de stabiliser les vitesses à l'avance et éviter les changements brusques.

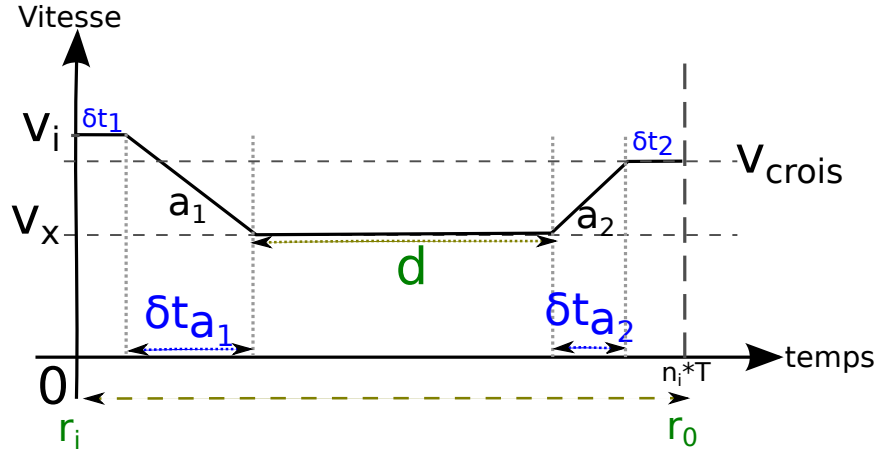


FIGURE 6.9 – Profil d'accélération

Nous allons maintenant calculer v_x pour obtenir δt_{a_1} et δt_{a_2} (les formules 6.6 et 6.7). D'après la figure 6.9, le temps de parcours de r_i à r_0 est donné par :

$$t_p = 2\delta t_{l+\frac{\lambda}{2}} + \delta t_{a_1} + \delta t_{a_2} + \frac{d}{v_x}, \text{ où} \quad (6.5)$$

$$\delta t_{a_1} = \frac{v_x - v_i}{a_1}, \quad (6.6)$$

$$\delta t_{a_2} = \frac{v_{crois} - v_x}{a_2}, \quad (6.7)$$

$$\delta t_1 = \frac{l + \frac{\lambda}{2}}{v_{crois}}, \text{ et} \quad (6.8)$$

$$\delta t_2 = \frac{l + \frac{\lambda}{2}}{v_i}, \quad (6.9)$$

et la distance parcourue de r_i à r_0 est :

$$r_p = 2l + \lambda + \int_0^{\delta t_{a_1}} (v_i + a_1 t) dt + \int_0^{\delta t_{a_2}} (v_{crois} - a_2 t) dt + d. \quad (6.10)$$

En utilisant les deux formules 6.5 et 6.10 et le paramètre d , on obtient cette équation du second degré :

$$v_x^2 \left(\frac{a_1 - a_2}{2a_1 a_2} \right) + v_x \left(t_p - \delta t_1 - \delta t_2 + \frac{a_2 v_i - a_1 v_{crois}}{a_1 a_2} \right) - \left(r_p - 2l - \lambda + \frac{a_2 v_i^2 - a_1 v_{crois}^2}{2a_1 a_2} \right) = 0. \quad (6.11)$$

Les détails de calcul sont présentés en annexe A.1.

Le discriminant de cette équation est :

$$\begin{aligned} \Delta &= b^2 - 4ac \\ &= \left(t_p - \delta t_1 - \delta t_2 + \frac{a_2 v_i - a_1 v_{crois}}{a_1 a_2} \right)^2 + 4 \left(\frac{a_1 - a_2}{2a_1 a_2} \right) \left(r_p - 2l - \lambda + \frac{a_2 v_i^2 - a_1 v_{crois}^2}{2a_1 a_2} \right). \end{aligned}$$

Si le discriminant Δ est strictement positif, l'équation admet deux solutions v_{x1} et v_{x2} données par les formules suivantes : $v_{x1} = \frac{-b - \sqrt{\Delta}}{2a}$ et $v_{x2} = \frac{-b + \sqrt{\Delta}}{2a}$.

S'il est nul, l'équation admet une racine double : $av_x^2 + bv_x + c = a \left(v_x + \frac{b}{2a} \right)^2$ et $v_{x1} = v_{x2} = -\frac{b}{2a}$, sinon l'équation n'admet pas de solution réelle.

L'agent détermine en conséquence, les commandes d'accélération et de décélération à appliquer et envoie au véhicule le profil d'accélération. Par la suite, le véhicule appliquera la vitesse de croisière v_{crois} à l'intérieur de l'intersection (après r_0).

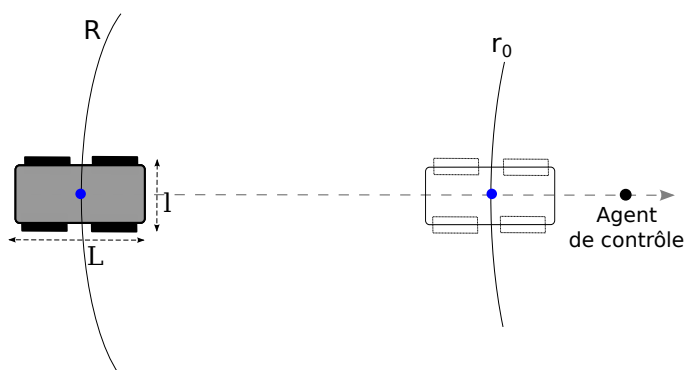
Remarque : Dans le cas où le discriminant de l'équation quadratique 6.11 n'admet pas de solution, c'est à dire qu'il n'existe aucune solution (vitesse) à appliquer en fonction de l'accélération maximale et la décélération maximale qui permettra au véhicule d'arriver au bon moment ou que V_x n'est simplement pas réaliste ($V_x < 0$ ou $V_x > V_{Max}$), le véhicule doit s'arrêter. Ceci ne peut arriver que si le véhicule est très proche du rayon r_0 , et donc en conséquence, la distance entre les rayons r_i (où le véhicule est détecté) et r_0 ne peut pas satisfaire cette adaptation de vitesse. C'est pour cette raison que nous allons étudier, dans la section suivante, les tailles minimales de nos deux rayons.

6.1.6 Rayons de la zone de contrôle

Les rayons r_0 et R (voir figure 6.10) de chaque intersection ne doivent pas être choisis arbitrairement. En effet, ils dépendent fortement de l'angle θ résultant du croisement des routes de chaque intersection.

Concernant r_0 , le rayon minimum doit satisfaire la condition que lorsque le centre de véhicule i est à la distance r_0 du centre de l'intersection, le nez de i n'a pas encore pénétré dans la zone de passage. Cette condition implique :

$$r_0 \geq \frac{L}{2} + \frac{l \max(\cos \frac{\theta}{2}, \cos \frac{\pi - \theta}{2})}{\sin \theta} \quad (6.12)$$

FIGURE 6.10 – Un véhicule en direction de l’intersection entre R et r_0

(les détails de calcul sont dans l’annexe A.2).

Le rayon R doit être tel que l’équation quadratique 6.11 peut toujours être résolue (satisfaisant les contraintes mathématiques mentionnées dans la section précédente).

Pour chaque véhicule, le cas limite, est quand il devrait attendre presque une période entière avant de passer, c’est à dire que la durée du profil d’accélération est $t_p - t = \frac{R-r_0}{V} + T + \epsilon$ au lieu de $\frac{R-r_0}{V} + \epsilon$ (nous avons remplacé r_i par R le rayon de détection).

Ce problème devient plus complexe lorsque plusieurs véhicules se suivent et forment des files d’attente. En effet, il n’y a pas de solution idéale pour R , car il peut toujours y avoir des véhicules qui arrivent en grand nombre, et saturent le segment entre R et r_0 . Dans la pratique (voir section 6.2), nous veillons à ce que le rayon R soit assez grand pour permettre l’application du profil d’accélération, mais nous vérifions que ce rayon reste raisonnable (de l’ordre de 100 mètres).

6.2 Expérimentations

Dans cette section nous évaluons notre approche en plaçant un agent de contrôle sur chaque intersection de deux routes avec deux flux opposés comme le montre la figure 6.11-a. Nous testons notre approche sur un réseau de trois intersections organisées sous forme de triangle (figure 6.11-b), mais les résultats présentés sont les évaluations sur une seule intersection.

Nous comparons notre approche à une stratégie basée sur les feux de circulation à durée fixe. Un feu de circulation est placé à l’entrée des croisements pour chaque flux. Les feux permettent d’alterner le passage des véhicules entre deux routes, pendant des périodes fixes et égales. Quand la stratégie des feux de circulation est appliquée et quand les véhicules perçoivent un feu rouge à une distance r_0 de l’intersection, ils s’arrêtent. Quand le feu vire au vert, ils démarrent progressivement.

6.2.1 Simulations

Nous avons utilisé notre simulateur à espace continu et à temps discret (figure 6.11 *a* et *b* sont des captures d'écran).

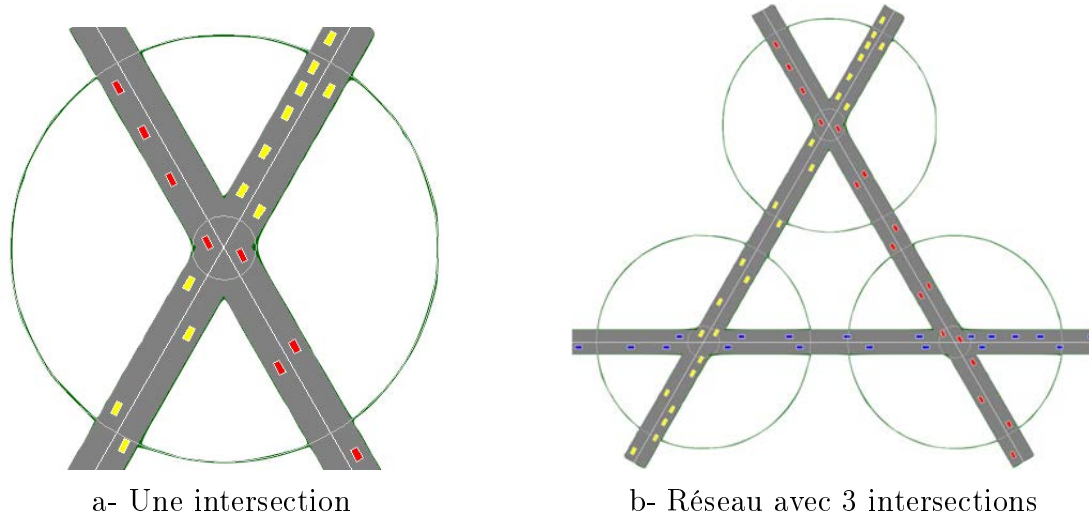


FIGURE 6.11 – Illustration du simulateur

Les paramètres de simulation utilisés sont :

- chaque route est de longueur $1000m$ (donc chaque segment est de longueur $500m$) ;
- l'angle de croisement est $\frac{\pi}{3}$;
- le rayon d'action R de l'agent de contrôle est fixé à $200m$, tandis que $r_0 = 30m$ (déterminé à partir de l'équation 6.12) ;
- les véhicules ont les dimensions $L = 5m$, $l = 2,5m$, ainsi qu'une marge de sécurité d'inter-distance $\epsilon = 1,5m$;
- le pas de simulation est de $0,1s$.

Pour cette configuration, la période minimale est de $T = 3,2s$. Nous rappelons que la vitesse maximale des véhicules est de $10m/s$, l'accélération maximale est de $1,5m/s^2$, la décélération maximale est de $-1,5m/s^2$, et que chaque source injecte des véhicules suivant un processus Bernoulli de paramètre $\lambda = \frac{1}{D}$ où D est la durée moyenne, en secondes, entre deux injections consécutives.

Pour l'approche par feux de circulation, nous pouvons fixer les temps de feux rouge et vert pour chaque flux. Nous avons varié leur durée sachant que la valeur de référence est le cas $30s$ pour chaque couleur. Pour toutes les intersections, les feux d'une même route changent de phases en même temps.

6.2.2 Comparaison des stratégies

Notre objectif est de comparer notre approche avec les feux de circulation en observant les retards résultants. Nous considérons le retard comme la différence entre un temps de

passage théorique (calculé à partir de la distance à parcourir pour traverser une intersection) et le temps de passage réel. Pour chaque approche, nous avons calculé la moyenne et l'écart type des retards pour 100 véhicules qui traversent le réseau. Nous avons utilisé une fréquence d'injection élevée en chaque source $D = 4s$ donnant un trafic dense.

La figure 6.12 présente les résultats de simulation de notre approche et de l'approche par feux de circulation avec différentes durées de phases (10s, 20s, 30s). Sur l'axe des abscisses nous avons le nombre de véhicules ayant quitté le réseau dans l'ordre de sortie et sur l'axe des ordonnées le retard moyen (plus l'écart-type) des véhicules en secondes. Les courbes tracées sont des moyennes sur 100 simulations. Nous arrêtons les simulations dès que 100 véhicules sont sortis mais nous n'arrêtons pas les injections de véhicules.

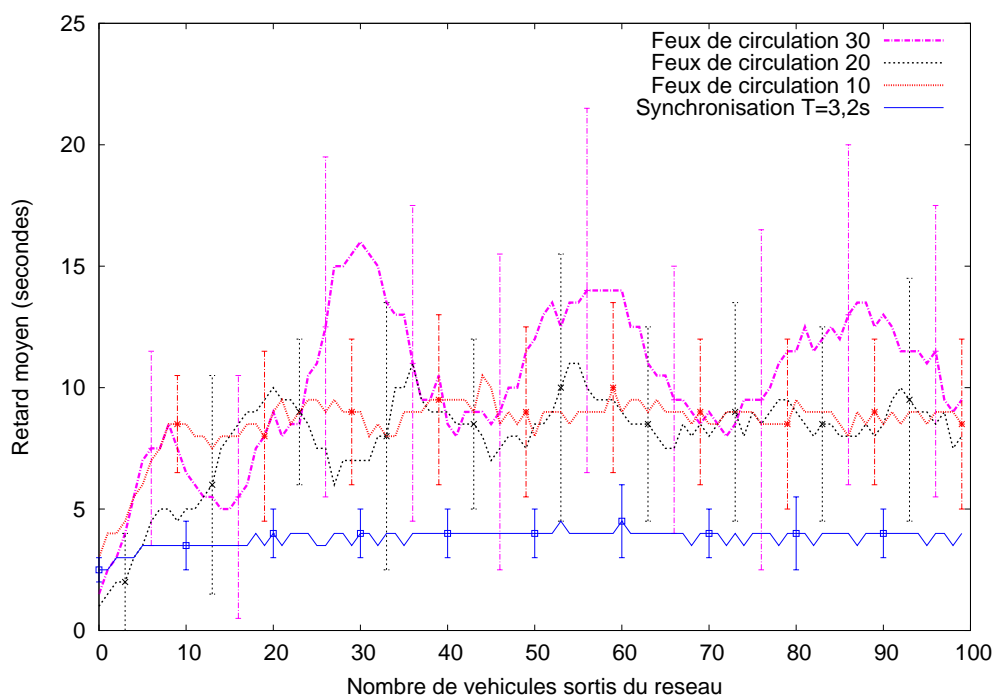


FIGURE 6.12 – Synchronisation vs. Feux de circulation à une intersection

L'observation des courbes montre que notre approche est clairement plus performante sur cette expérimentation que les feux de circulation (spécialement en comparant la courbe utilisant une durée de feux 30s). Nous remarquons que l'approche par feux de circulation se dégrade quand on augmente la phase. Sur 100 simulations, le pire retard réalisé par un véhicule avec l'approche par synchronisation ne dépasse pas les 6s, alors que, pour l'approche par feux de circulation, il dépasse les 20s (valeur maximale des écarts-types utilisant une durée de feux 30s). La moyenne générale des retards est de l'ordre de 4s en synchronisation locale alors que pour la meilleure courbe des feux de circulation, elle est de l'ordre de 9s (utilisant une durée de feux de 10s ou 20s). Nous avons aussi testé l'approche par feux de circulation avec des durées de phases inférieures à 10s et nous avons observé que la moyenne des retards augmente significativement. Ceci est dû principalement aux véhicules qui s'arrêtent et requièrent ainsi plus de temps pour traverser l'intersection, ce

qui conduit même à la formation de files d'attente et à des collisions quand la durée des phases est égale ou inférieure à 5s.

6.2.3 Effet de variation du paramètre R (rayon d'action)

Dans cette section, nous avons mesuré l'effet de la variation du rayon d'action R de l'agent de contrôle afin d'évaluer s'il influe sur la qualité de notre solution (sans modifier r_0). Le tableau 6.1 donne les moyennes et les écarts-types de la consommation d'énergie cinétique seulement à l'accélération (section 5.2) lorsque $R = 50m$, $R = 100m$ et $R = 200m$. Elle est mesurée pendant la traversée de 1000 véhicules. Nous observons que les retards ne changent pas de manière significative, mais il y a une différence en termes de consommation d'énergie lorsque nous réduisons le rayon. Elle augmente quand R est réduit. En effet, la faible distance oblige l'agent de contrôle à ralentir brutalement les véhicules (pour corriger leur avance) et donc à générer en sortie une accélération consommatrice d'énergie.

TABLE 6.1 – Moyennes et Écarts-types de l'énergie consommée en fonction du rayon d'action $R = 50m$, $R = 100m$ et $R = 200m$

$D = 4s$	$50m$	$100m$	$200m$
Moyennes d'énergie	70,7	51,72	34,62
Écart-types	$\pm 20,1$	$\pm 18,98$	$\pm 17,25$

6.2.4 Effet de variation du paramètre D (débit)

Nous voulons étudier, ici, l'efficacité de notre approche lorsque le débit de véhicules est important. La figure 6.13 montre les variations des temps d'attentes selon les débits d'injection. Les histogrammes tracés sont des moyennes sur 1000 véhicules successifs.

Nous avons utilisé les valeurs D de 1s, 2s, 3s, 4s et 6s qui correspondent respectivement à des débits de 60, 30, 20, 15 et 10 véhicules par minute. Selon la figure, le retard moyen augmente légèrement (moins d'une seconde) lorsque le débit est inférieur ou égal à 20veh/m ($D=3s$), parce que la période minimale d'une intersection est dans notre cas 3,2s. Immédiatement après, en utilisant les débits 30veh/m et 60veh/m, le système est saturé. Les véhicules s'accumulent et ont des retards croissants. Cette limite a été aussi constatée dans le cadre des approches par réservation [Dresner et Stone, 2004].

6.3 Synchronisation n -véhicules/ p -véhicules

Nous avons démontré que nous pouvons croiser en alternance deux flux de véhicules passant un à un sans arrêt avec une période T . Dans cette partie, on s'intéresse à la

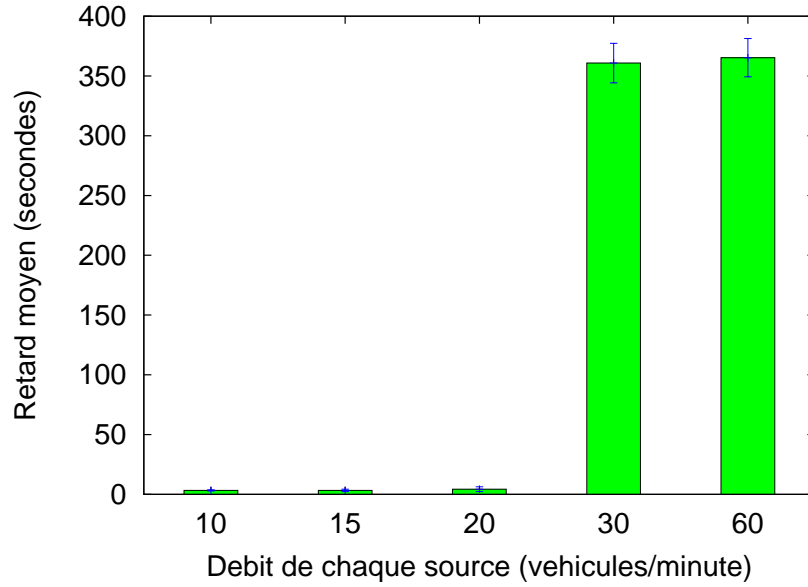
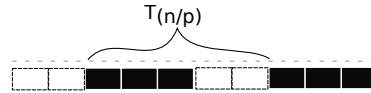


FIGURE 6.13 – Mesure du retard moyen des véhicules à la variation du débit

possibilité de réduire le nombre de périodes lorsque les débits entre les routes sont très différents. L'idée est de faire passer des paquets de véhicules, n véhicules d'un flux contre p véhicules de l'autre. Cette solution peut correspondre à des scénarios où nous connaissons a priori l'hétérogénéité des débits.

Nous débutons par l'étude d'une intersection avec un angle $\theta = 90^\circ$. La figure 6.14 illustre ce cas quand $n = 3$ et $p = 2$.

FIGURE 6.14 – Croisement de deux flux $\theta = 90^\circ$, $n = 3$, $p = 2$

Nous savons, par la formule 6.1, que la période de passage en alternance pour un croisement de deux véhicules est $T_{1/1} = 2(L + l)/V$, ce qui implique une période de passage par véhicule $T_{veh(1/1)} = \frac{T_{1/1}}{1+1} = (L + l)/V$. Maintenant si on considère n et p quelconque alors $T_{n/p} = [(n + p)L + 2l]/V$ et $T_{veh(n/p)} = \frac{T_{n/p}}{n + p} = (L + \frac{2}{(n + p)}l)/V$.

Pour $n = 3$, $p = 2$, on peut déterminer $T_{3/2} = (5L + 2l)/V$, ce qui donne $T_{veh(3/2)} = (L + \frac{2l}{5})/V$.

Donc, avec un angle de croisement $\theta = 90^\circ$, passer des groupes de véhicules réduit la période par véhicule car $T_{veh(n/p)}$ décroît avec $n + p$.

Considérons maintenant les deux autres cas : quand $\theta > 90^\circ$ et $\theta < 90^\circ$. D'après les formules 6.2 et 6.3 on a $T_{(1/1)} = 2(L + l(\theta))/V$, avec $l(\theta) = l \frac{(1 + \cos \theta)}{\sin \theta}$ quand $\theta > 90^\circ$ et

$l(\theta) = l \frac{(1 - \cos \theta)}{\sin \theta}$ quand $\theta < 90^\circ$. D'où $T_{veh(1/1)} = (L + l(\theta))/V$, ce qui implique que

$$T_{veh(n/p)} = (L + \frac{2}{(n+p)}l(\theta))/V < T_{veh(1/1)}, \text{ pour } (n+p > 2). \quad (6.13)$$

Nous avons démontré, dans cette partie, qu'alterner n -véhicules/ p -véhicules permet de réduire la période par véhicule et augmente la période par bloc. Cette solution peut être efficace si a priori nous connaissons les débits. Néanmoins, cela nécessite la création de paquets de véhicules de taille limité en fonction du rayon R (sinon le profil d'accélération n'admet plus de solution), et sachant que plus les paquets sont gros, plus les véhicules devront ralentir et réaccélérer.

6.4 Conclusion

Dans ce chapitre, nous avons abordé le problème du croisement automatique de véhicules sur un réseau, ou plus généralement d'agents mobiles autonomes. Pour cela, nous avons proposé une stratégie de contrôle locale aux intersections. Elle consiste à espacer les véhicules de sorte qu'ils peuvent se croiser sans s'arrêter. En synchronisant l'arrivée des véhicules avec une période similaire sur les flux, mais déphasée d'une demi-période selon la route, les véhicules de chaque flux passent alternativement. Ainsi les véhicules, si besoin, ne subissent potentiellement qu'un léger ralentissement, ce qui est mieux que de les arrêter.

Nous avons défini sur ce principe un agent de contrôle en chaque intersection qui utilise uniquement ses perceptions locales du trafic (rayon d'action R). Il détermine la consigne de passage pour le croisement de chaque véhicule en fonction de sa distance et de la parité assignée à sa route. L'étude expérimentale a permis de montrer que la taille du rayon d'action R influence directement l'optimisation, la capacité de régulation du trafic aux croisements et le gain en temps. Par ailleurs, ce gain est important par rapport à un système classique de feux de circulation.

Dans le prochain chapitre, nous examinons comment faire communiquer les agents de croisement pour les synchroniser, dans le but d'améliorer encore le trafic sur le réseau. Cela passe par la recherche de relations de synchronisation entre plusieurs croisements liés par une même route.

Chapitre 7

Synchronisation des intersections d'un réseau

Sommaire

7.1	Notations et hypothèses	104
7.1.1	Contrainte de périodicité	106
7.1.2	Contrôle des phases	106
7.2	Critères d'optimisation séparables	107
7.3	Coordination et optimisation des phases des intersections	110
7.3.1	Hill-Climbing (HC)	110
7.3.2	Paysages des fonctions à optimiser	111
7.3.2.1	Périodicité	111
7.3.2.2	Temps	111
7.3.2.3	Énergie	112
7.3.2.4	Similarité entre les deux critères	113
7.3.3	Algorithme de recherche stochastique distribuée (DSA)	114
7.3.3.1	Principe de la recherche stochastique distribuée	114
7.3.3.2	Besoin du saut $T/2$	114
7.3.3.3	Algorithme proposé	115
7.3.4	De l'optimisation hors ligne à l'optimisation en ligne	115
7.3.4.1	L'optimisation hors ligne	116
7.3.4.2	L'optimisation en ligne	116
7.4	Stratégie premier arrivé premier servi (FCFS)	117
7.5	Expérimentations	118
7.5.1	Paramètres de simulation	118
7.5.2	Courbes d'optimisation obtenues	119
7.5.3	Optimisation hors ligne	121
7.5.4	Optimisation en ligne	122
7.6	Conclusion	123

Nous avons montré au chapitre précédent que l'application de l'algorithme de synchronisation par l'agent de contrôle en chaque intersection permet de fluidifier le trafic. Nous étudions, dans ce chapitre, la possibilité de coordonner les agents de contrôle de chaque intersection. Chaque agent de contrôle aura pour objectif soit d'optimiser l'économie d'énergie, soit d'optimiser les retards en tenant compte des agents de contrôle voisins. Intuitivement, nous voulons tendre vers des « ondes vertes » qui facilitent la traversée rapide d'une succession d'intersections à travers la coordination des agents de contrôle.

Dans la première section, nous commençons par introduire les notations et les contraintes de notre problème qui serviront pour la suite. En section 7.2 nous présentons nos critères d'optimisation. Ensuite, en section 7.3, nous exposons les différents algorithmes développés. Une stratégie opportuniste appelée FCFS et inspirée de [Dresner et Stone \[2004, 2005\]](#) sera présentée en section 7.4. Enfin, en section 7.5, nous comparons cette stratégie à notre approche et nous concluons.

7.1 Notations et hypothèses

Dans le chapitre 6, nous avons développé une approche permettant le croisement automatique sans arrêt de véhicules, qui traite chaque intersection indépendamment des autres. Elle est basée sur la synchronisation périodique de l'arrivée des véhicules de chaque route entrante.

L'idée que nous développons dans ce chapitre consiste à bien synchroniser les intersections entre elles (figure 7.1), de telle sorte que chaque véhicule quittant l'intersection i arrive au bon moment à l'intersection suivante j pour qu'il passe sans arrêt et avec un contrôle minimale (modification de vitesse) de la part de l'agent contrôle de j . La résolution de ce problème implique le contrôle du déphasage entre les périodes des intersections.



FIGURE 7.1 – Réseau composé de 5 intersections ; synchroniser les intersections implique le contrôle des déphasages de chacune d'elles

Nous considérons, dans ce qui suit, chaque intersection comme le résultat du croisement de deux routes à deux voies, comme dans la figure 7.2. Nous notons $\varphi_{i,j}$ la phase du flux périodique de véhicules sortants de l'intersection i vers l'intersection j .

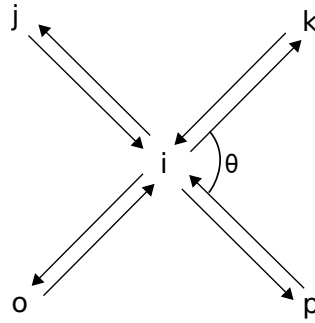


FIGURE 7.2 – L’intersection i est le résultat du croisement de deux routes à deux voies

Dans l’exemple de la figure 7.2, l’intersection i doit assurer le passage des véhicules vers les intersections o , k , j et p . Dans la section 6.1.4, nous avons proposé une solution pour la synchronisation de deux routes qui implique que les deux voies opposées de chaque route doivent faire traverser l’intersection aux véhicules en même temps. Pour l’exemple de l’intersection i , il en résulte que les véhicules allant vers j et p passent ensemble, ensuite ceux en direction de o et k ensemble. Nous en déduisons la périodicité de l’intersection i décrite par la figure 7.3.

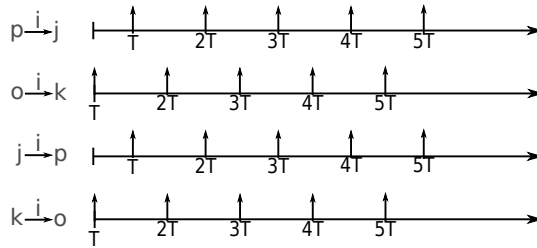


FIGURE 7.3 – Représentation des temps de passage vus comme des signaux périodiques selon les 4 directions de circulation

Donc pour chaque intersection i , nous avons deux signaux périodiques avec une différence de phase (déphasage) des périodes entre les deux routes de chaque intersection congrue à $\frac{T}{2} - 2\phi_\lambda[T]$. Ceci implique, comme le montre l’exemple de la figure 7.3, que $\varphi_{i,k} = \varphi_{i,o}$ et $\varphi_{i,j} = \varphi_{i,p}$, ainsi que $\varphi_{i,k} \equiv \varphi_{i,j} + \frac{T}{2} - 2\phi_\lambda[T]$ et $\varphi_{i,o} \equiv \varphi_{i,p} + \frac{T}{2} - 2\phi_\lambda[T]$. Nous en déduisons que les véhicules provenant de i et en direction de j passeront à des temps congrus à $\varphi_{i,j} + \frac{T_{min}^i}{2} - 2\phi_\lambda[T_{min}^i]$, alors que les véhicules provenant de i et en direction de k passeront à des temps congrus à $\varphi_{i,j} + T_{min}^i[T_{min}^i]$.

Pour pouvoir établir une coopération entre les différents agents de contrôle, nous posons trois hypothèses que nous justifierons dans les sections suivantes :

- chaque agent de contrôle d’une intersection i connaît ses voisins et la distance qui les sépare ;
- la période de chaque intersection T_{min}^i est perçue par les voisines proches ;
- les phases des voies d’une intersection sont connues par ses voisines.

Dans ce qui suit, nous appellerons la phase $\vec{\phi}_i$ une phase de référence de l’intersection i et nous déterminerons les autres phases par les formules sus-citées.

7.1.1 Contrainte de périodicité

On a montré, dans le chapitre 6, que pour synchroniser deux flux de véhicules arrivant de deux routes, chaque intersection doit respecter une période de passage T_{min} . Cette période dépend de l'angle du croisement θ . Si nous considérons un réseau quelconque, notre réseau sera composé de n croisements donc de n périodes minimales qu'on notera T_{min}^n .

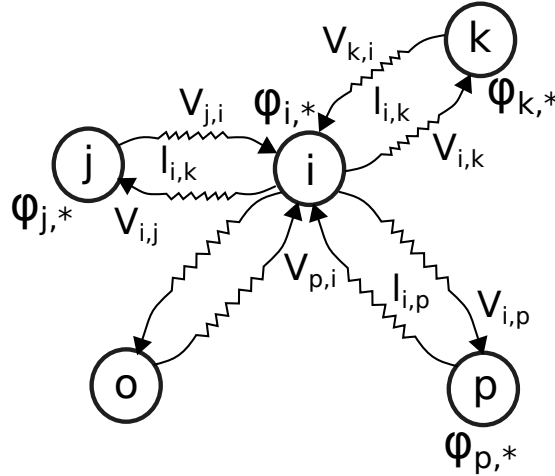
Sur une voie, l'intersection à la plus grande période minimale impose son débit maximum à toutes les autres intersections suivantes. Si ces débits maximums sont respectés, on ne créera pas d'accumulation de véhicules à l'intérieur du réseau mais pas de garantie aux entrées. Pour un réseau avec des périodes hétérogènes, les durées de parcours des segments seront variables, et donc incontrôlables. Cependant, si nous utilisons une période unique, les phases des intersections permettent de contrôler ces durées de parcours. Donc, l'idée est que toutes les intersections passent les véhicules à la plus grande période des n intersections $\forall i, T_{min}^i = \max(T_{min}^n)$. Ceci impliquera peut être un léger retard, mais une fois la période définie pour toutes les intersections du réseau, elle va nous permettre d'éviter l'entassement des véhicules entre deux intersections successives (conservation des véhicules). En plus, dans les vrais réseaux de transport, l'angle du croisement ne varie pas beaucoup d'une intersection à une autre lorsque la valeur de θ_i reste dans l'intervalle habituel $[\frac{\pi}{3}, \frac{2\pi}{3}]$.

7.1.2 Contrôle des phases

Nous voulons dans ce qui suit optimiser le trafic routier à travers les phases des intersections. L'objectif est de tendre vers des ondes vertes, typiquement comme dans les grandes artères des villes où les conducteurs des véhicules traversent plusieurs feux verts successifs sans s'arrêter.

Compte tenu de notre approche par alternance et du signal périodique de toutes les intersections, la seule variable de contrôle est la phase $\phi_i \in [0, T[$ du signal périodique de chaque intersection. Comme illustré dans la figure 7.4, une intersection i interagit directement avec ses voisines. Dans l'exemple, les intersections voisines de i sont $\mathcal{E}_i = \{j, k, o, p\}$. Leurs phases contraignent la durée de la traversée de chaque segment. Par exemple, nous pouvons obtenir une vitesse moyenne souhaitée $V_{i,j}$ sur le segment $i \rightarrow j$, en réglant bien les phases ϕ_i et ϕ_j . Maintenant, si nous passons à l'échelle de toute l'intersection i , chacun des huit segments représente une contrainte à satisfaire (en fonction du critère d'optimisation), tandis que seulement cinq paramètres peuvent être réglés. Ce qui signifie que, même sur ce petit exemple, des compromis seront nécessaires.

Nous allons maintenant discuter des critères d'optimisation que l'on pourra contrôler avant de décrire les algorithmes d'optimisation que nous allons utiliser.

FIGURE 7.4 – La phase optimale ϕ_i de l'intersection i dépend des phases de ses voisines

7.2 Critères d'optimisation séparables

Nous nous intéressons, dans cette section, aux mêmes critères introduits dans la section 5.2 (le temps et l'énergie).

Notons $\vec{\phi}$ le vecteur de toutes les phases des intersections du réseau $\vec{\phi} = (\vec{\phi}_1, \vec{\phi}_2, \dots, \vec{\phi}_n)$. Les deux objectifs sus-cités seront examinés indépendamment :

1. minimiser les temps de traversée, c'est-à-dire la somme des temps de traversée $t_{i,j}$ sur chaque segment orienté $i \rightarrow j$, pondérée en fonction du débit des flux $W_{i,j} = W_l$ (où l est la voie correspondante au segment $i \rightarrow j$ et \mathcal{E} l'ensemble des voies du réseau) afin de tenir compte de l'état réel de chaque segment de route :

$$t_{global}(\vec{\phi}) = \sum_{(i \rightarrow j) \in \mathcal{E}} W_{i,j} t_{i,j}(\vec{\phi}), \text{ et}$$

2. minimiser l'énergie totale consommée, c'est-à-dire la somme de l'énergie cinétique $E_{i,j}$ sur chaque segment orienté $i \rightarrow j$, pondérée en fonction du débit des flux $W_{i,j}$:

$$E_{global}(\vec{\phi}) = \sum_{(i \rightarrow j) \in \mathcal{E}} W_{i,j} E_{i,j}(\vec{\phi}).$$

Notons qu'il n'y a pas de risque que cette minimisation conduise au blocage de certains véhicules, puisque nous supposons que nous utilisons une régulation du trafic périodique qui permet à tous les véhicules de passer.

Ici, nous faisons l'hypothèse que tous les véhicules traversent les intersections à la même vitesse par défaut. Ceci est suffisant pour garantir que le temps de traversée local $t_{i,j}$ et la consommation d'énergie locale $E_{i,j}$ ne dépendent que de ϕ_i et ϕ_j .

Comme nous pouvons l'observer, les deux critères ci-dessus sont additivement sépara-

bles, puisqu'ils peuvent être écrits de la manière suivante :

$$f_{global}(\vec{\phi}) = \sum_{(i \rightarrow j) \in \mathcal{E}} f_{i,j}(\phi_i, \phi_j),$$

où $f_{i,j}$ représente soit l'énergie cinétique, soit le temps de traversée.

Au contraire, minimiser le pire temps de traversée d'une voie d'un segment conduira à un critère d'optimisation non séparable :

$$t_{worst}(\vec{\phi}) = \max_l \sum_{(i,j) \in \mathcal{E}_l} W_{i,j} t_{i,j}(\vec{\phi}),$$

où \mathcal{E}_l est un ensemble de segments de la voie l .

Un premier avantage de ces critères séparables est qu'ils peuvent être calculés de manière efficace en parallèle, ce qui peut être profitable pour les grands réseaux. En outre, cette séparabilité nous permet de factoriser l'optimisation, c'est-à-dire d'associer à chaque intersection un critère individuel et une variable de contrôle. Un deuxième avantage est que chaque phase ϕ_i influence uniquement ses segments de voies entrantes et sortantes, ainsi chaque intersection i optimisera :

$$f_i(\vec{\phi}) = \sum_{j \in \mathcal{E}_i} [f_{i,j} W_{i,j} + f_{j,i} W_{j,i}],$$

où $f_{i,j}$ représente soit l'énergie cinétique, soit le temps de traversée, et \mathcal{E}_i l'ensemble des voisins de i .

En conséquence, nous allons naturellement obtenir des algorithmes d'optimisation décentralisés, réduisant ainsi considérablement la complexité de la recherche. Cependant, cette séparabilité n'empêche pas de tomber dans un optimum local.

Évaluation des critères

Avant de proposer une approche, nous avons besoin de savoir évaluer ces critères d'optimisation. La fonction *Évaluation* va nous permettre d'évaluer la phase courante de chaque intersection à l'instant t à l'aide du profil d'accélération défini dans la section 6.1.5.

La mesure de performance à une intersection donnée est évaluée simplement en prenant la somme des mesures de performance des huit segments voisins, un segment par voie entrante et un segment par voie sortante. Pour chaque segment, les véhicules n'interagissent pas entre eux parce qu'ils sont séparés par une période T . La mesure de performance associée à chaque segment est calculée comme illustré par la figure 7.5, correspondant à la somme de :

1. l'énergie ou le temps nécessaire à un véhicule quittant une intersection i en direction de la prochaine zone de contrôle (dans la figure, entre i et R roulant à une vitesse constante v_{crois}),
2. l'énergie ou le temps requis dans la zone d'adaptation (dans la figure, entre R et r_0) calculé comme dans la section 6.1.5, et,
3. l'énergie ou le temps prévu pour passer à travers la prochaine intersection j (dans la figure, entre r_0 et j à vitesse v_{crois}).

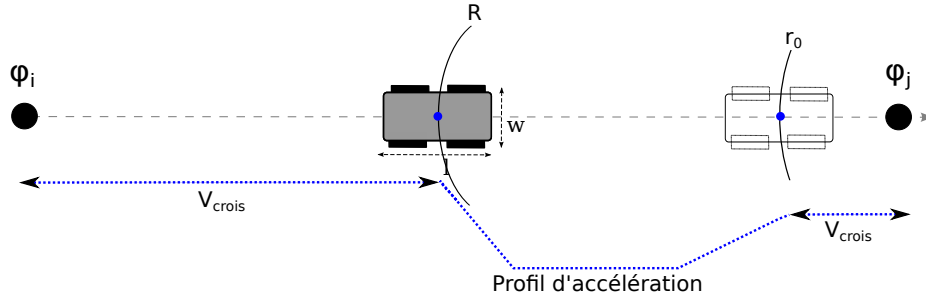


FIGURE 7.5 – Mesurer la performance d'un véhicule sur le segment $i \rightarrow j$ requière la décomposition de ce segment en trois parties : (1) entre i et la zone de contrôle de j de rayon R , (2) la phase d'adaptation dans la zone de contrôle entre R et r_0 , et (3) la zone de croisement de rayon r_0 .

La fonction d'évaluation calcule analytiquement le critère local en fonction de la phase de l'intersection elle même et des phases de ses voisines en supposant que les véhicules sont parfaits, c'est à dire qu'ils suivent exactement une solution idéale, en particulier parce qu'ils peuvent changer leurs paramètres de contrôle (accélération) à tout moment.

L'algorithme 7.1 illustre la fonction *Évaluation* considérant uniquement la zone d'adaptation (entre R et r_0). Les deux autres zones qui constituent la mesure peuvent être déterminées :

1. pour le critère temps, à partir de v_{crois} et la distance séparant l'intersection i du rayon R et l'intersection j de r_0 , et
2. pour le critère énergie, ces deux zones n'ont pas d'influence car il n'y a pas de variations de vitesse.

Algorithme 7.1 : Évaluation de la phase actuelle pour l'intersection i

- 1 Fonction **Évaluation** ($\varphi_i^t, \varphi_{voisins}^{t-1}$)
 - 2 **pour** chaque intersection j voisine de i **faire**
 - 3 $f_{i,j} = f_{i,j} + \mathbf{Profil\ d'accélération}(\varphi_i^t, \varphi_{voisins}^{t-1});$
 - 4 $f_{j,i} = f_{j,i} + \mathbf{Profil\ d'accélération}(\varphi_{voisins}^{t-1}, \varphi_i^t);$
 - 5 **Retourne** $f_i^E(\vec{\phi})$ ou $f_i^t(\vec{\phi});$
-

A partir de cette évaluation, l'objectif de chaque intersection est de trouver le bon déphasage de telle sorte que les véhicules maintiennent le plus longtemps possible v_{crois} entre intersections successives, que ce soit pour le critère temps ou le critère énergie car les véhicules ne peuvent pas dépasser cette vitesse.

7.3 Coordination et optimisation des phases des intersections

L'objectif des intersections n'est plus limité uniquement au calcul du profil d'accélération. Nous voulons que chaque intersection "fluidifie" au mieux l'état de son trafic, en tenant compte des intersections voisines. A l'aide de cette coordination entre intersections, chacune d'entre elles cherche à trouver son optimum individuel dans le but d'observer un comportement émergent qui correspond à l'optimum du réseau.

7.3.1 Hill-Climbing (HC)

Pour réaliser l'optimisation du réseau, différents algorithmes de recherche locale peuvent être envisagés, comme le Hill-Climbing, la descente de gradient (lorsque le gradient de la fonction d'évaluation peut être calculé) ou la recherche tabou [Russell et Norvig, 1995].

Nous considérons dans un premier temps une approche simple à l'aide de l'algorithme Hill-Climbing qui est une technique d'optimisation mathématique itérative commençant avec une solution arbitraire à un problème, puis tentant de trouver une meilleure solution en changeant progressivement un seul élément de la solution à la fois.

A chaque itération t , de chaque intersection i , l'algorithme cherche localement la meilleure phase tout en tenant compte des phases des intersections voisines $j \in \mathcal{E}_i$, et assigne cette valeur à l'intersection i : $\phi_{t+1}(i)$.

Les détails sont présentés dans l'algorithme 7.2, où $\Delta\phi_t$ est l'intervalle de recherche qui devrait diminuer lentement pour assurer la convergence, et $\delta\phi_t/\Delta\phi_t$ un entier constant.

Algorithme 7.2 : Hill-Climbing-iteration($i, t, \vec{\phi}_t$)

```

1  $V_{min} \leftarrow Evaluation(\phi_t(i), \vec{\phi})$  ;
2  $\phi_{min}(i) \leftarrow \phi_t(i)$  ;
3 pour  $\phi_{tmp} \leftarrow \phi_t(i) - \Delta\phi_t$  à  $\phi_t(i) + \Delta\phi_t$  pas =  $\delta\phi_t$  faire
4    $V_{tmp} \leftarrow Evaluation(\phi_{tmp}, \vec{\phi}_t)$  ;
5   si  $V_{tmp} < V_{min}$  alors
6      $V_{min} \leftarrow V_{tmp}$  ;
7      $\phi_{min} \leftarrow \phi_{tmp}$  ;
8  $\phi_{t+1}(i) \leftarrow \phi_{min}$  ;
```

Enfin, pour éviter que l'optimisation ne finisse assez tôt sur un plateau, cette dernière n'est arrêtée que lorsque nous observons un certain nombre d'itérations fixé à la main à l'avance (en pratique nous avons fixé le nombre d'itérations à 100).

7.3.2 Paysages des fonctions à optimiser

En vue d'améliorer l'algorithme utilisé, nous étudions les critères à optimiser, à savoir le temps de traversée et l'énergie consommée par les véhicules. Pour ce faire, nous fixons les phases des voisines et regardons l'évaluation de chaque critère.

7.3.2.1 Périodicité

Afin de synchroniser les flux de véhicules, chaque intersection i doit respecter une période de passage T_{min}^i . Optimiser i revient à optimiser les flux de toutes les voies entrantes et sortantes du croisement, c'est à dire que $f_i = (f_{Nord} + f_{Sud} + f_{Est} + f_{Ouest})$. Étant donné que la phase ϕ_i évolue dans le domaine $[0, T[$, nous optimisons donc une fonction périodique.

7.3.2.2 Temps

En considérant le temps comme critère d'optimisation, nous avons varié la phase entre $[0, 3T]$ et nous avons tracé l'évolution de notre critère dans la figure 7.6.

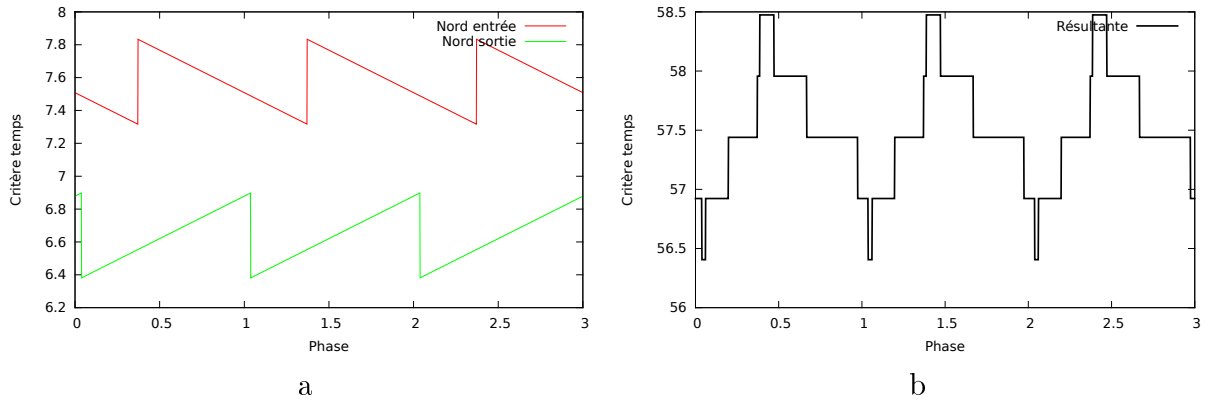


FIGURE 7.6 – Le critère temps en fonction de la phase : *a*- critère temps pour les deux segments entrant et sortant du flux nord, *b*- critère temps pour une intersection (la somme des 4 flux de 2 segments).

La courbe 7.6-a représente l'évolution du critère pour les deux segments entrant et sortant du flux nord. Nous remarquons déjà que le critère temps est linéaire et que les deux courbes ont deux pentes égales mais opposées à cause du même débit circulant sur les deux segments (figure 7.7).

Cette égalité crée une dualité entre les deux segments. En effet, en optimisant sa phase ϕ_i , l'intersection devra tenir compte des deux segments équitablement. Ceci impliquera des créneaux, au moment de l'évaluation de la phase.

La courbe 7.6-b illustre la fonction à optimiser $f_i = f_{Nord} + f_{Sud} + f_{Est} + f_{Ouest}$ en fonction de la phase. Nous remarquons qu'il existe des optima locaux pour l'intersection i , et aussi beaucoup de plateaux résultant de ces créneaux.

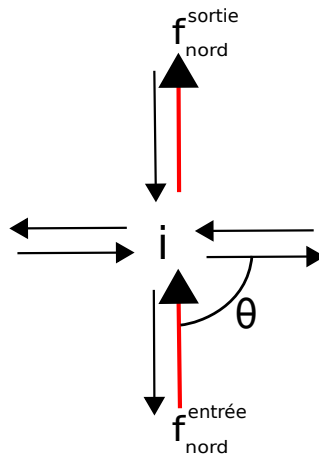


FIGURE 7.7 – Optimiser la voie orientée vers le nord et composée de deux segments (l’entrant et le sortant) revient à optimiser ϕ_i pour le flux f_{Nord}^{entree} et f_{Nord}^{sortie} .

7.3.2.3 Énergie

Passons maintenant au critère énergie. Nous avons aussi varié la phase dans l’intervalle $[0, 3T]$ et nous avons tracé l’évolution de notre critère dans la figure 7.8.

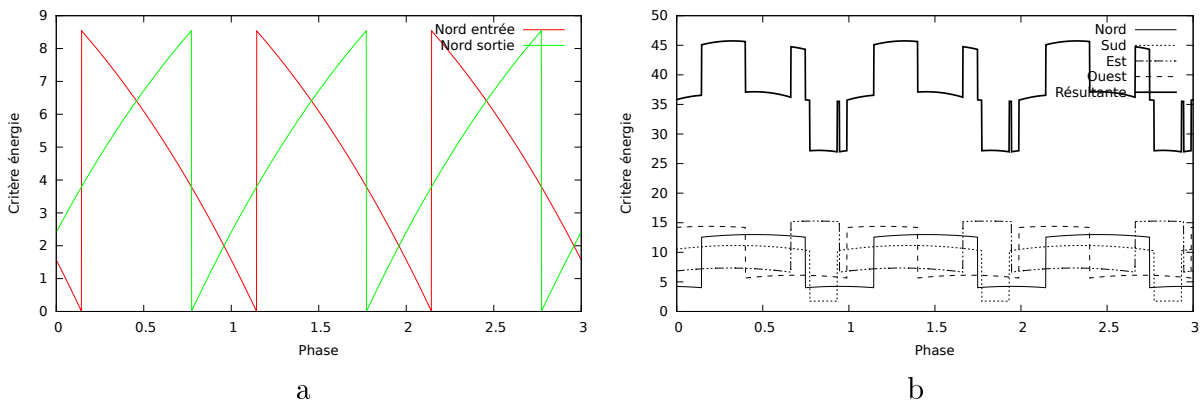


FIGURE 7.8 – Le critère énergie en fonction de la phase : *a*- critère énergie pour les deux segments entrant et sortant du flux nord, *b*- critère énergie pour une intersection

La courbe 7.8-a représente aussi l’évolution du critère pour les deux segments entrant et sortant de la voie allant vers nord. Nous remarquons que, lorsqu’une courbe croît, l’autre décroît. Ceci est dû à la dualité qui existe entre les deux segments.

Ce critère quadratique fait apparaître, au moment de l’évaluation de l’intersection, des créneaux arrondis. La courbe 7.8-b illustre la fonction à optimiser $f_i = f_{Nord} + f_{Sud} + f_{Est} + f_{Ouest}$ en fonction de la phase. Nous remarquons ici aussi qu’il existe des optima locaux et beaucoup de “plateaux arrondis”.

7.3.2.4 Similarité entre les deux critères

Dans cette section, nous voulons étudier les similarités entre les deux critères. Pour une intersection i du réseau, nous avons varié sa phase dans l'intervalle $[0, 3T]$ tout en fixant les phases de toutes les autres intersections. Nous avons testé trois configurations différentes où nous avons utilisé les mêmes phases initiales des voisins pour étudier les deux critères.

Les figures 7.9 *a*, *b* et *c* illustrent l'évolution du critère énergie en fonction de la phase et les figures 7.9 *d*, *e* et *f* illustrent l'évolution du critère temps en fonction de la phase pour les trois mêmes configurations des phases des intersections voisines.

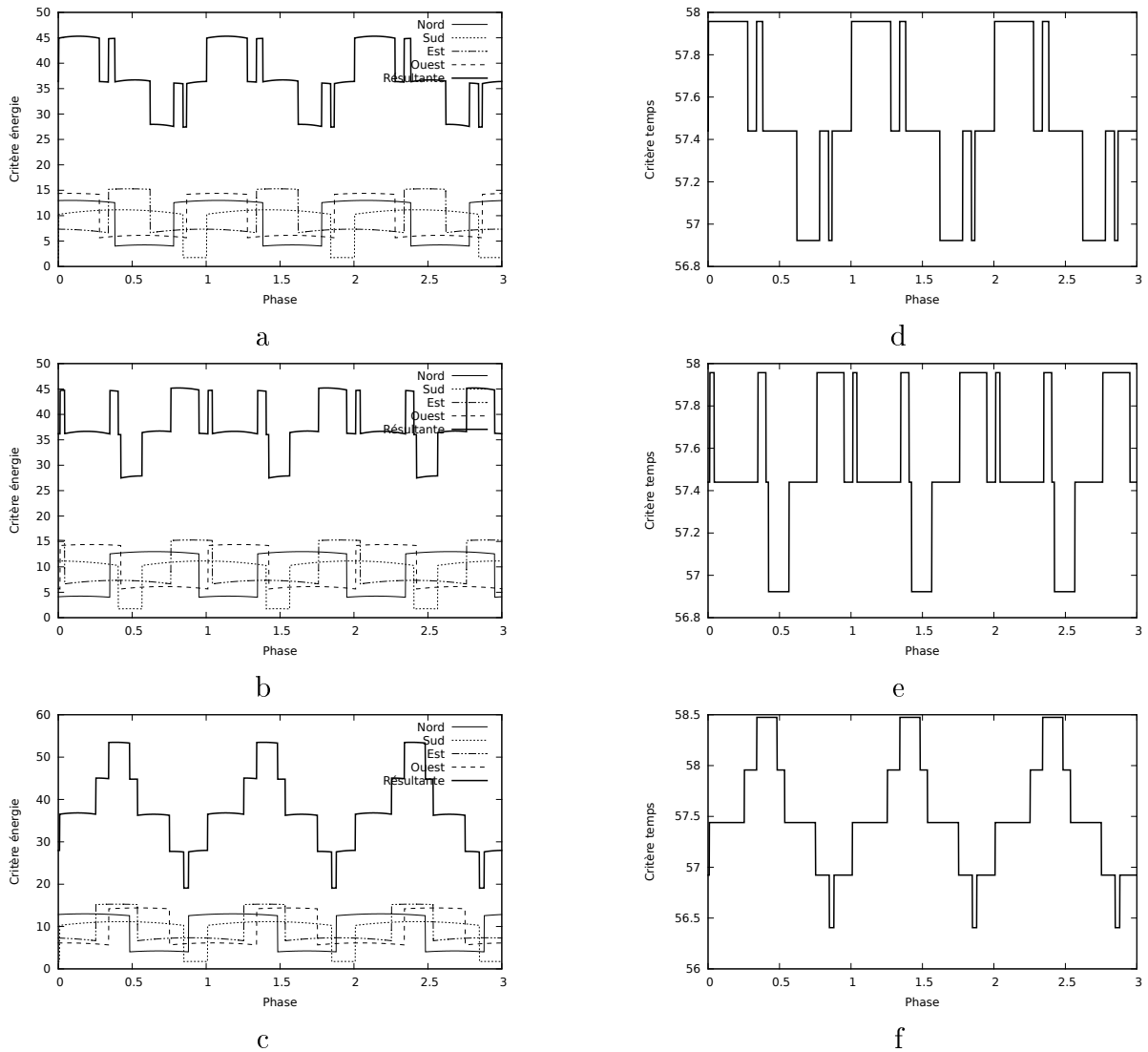


FIGURE 7.9 – Comparaison entre les courbes d'optimisation obtenues en fonction du critère énergie (*a*, *b*, *c*) et le critère temps (*d*, *e*, *f*) pour les mêmes trois configurations

Nous remarquons, dans notre cas, que les critères temps et énergie ont la même forme générale, quasiment identique à une transformation affine près. Ceci est dû au fait que

les véhicules ne peuvent que décélérer, et pour revenir à la vitesse de croisière ils mettent du temps et consomment de l'énergie. Dans le cas contraire, si la vitesse de croisière n'était pas égale à la vitesse maximale, nous aurions une différence de forme entre les deux courbes, car les véhicules en réalité auraient le choix non seulement de ralentir pour atteindre une période mais aussi d'accélérer.

Pour résumer, les fonctions obtenues, que ce soit pour le critère temps ou le critère énergie, ne sont pas faciles à optimiser à chaque intersection. Par conséquent, l'optimisation au niveau du réseau va être très difficile pour un simple algorithme de recherche locale comme HC.

7.3.3 Algorithme de recherche stochastique distribuée (DSA)

Dans cette section, nous allons étudier l'algorithme de recherche stochastique distribuée (DSA) afin de remédier aux problèmes résultant des optima locaux. En particulier, nous voulons trouver une solution qui converge efficacement vers un optimum et qui s'adapte continuellement.

7.3.3.1 Principe de la recherche stochastique distribuée

Nous savons qu'il existe une forte dépendance entre les différentes intersections du réseau. Cette dépendance résulte du fait que les voies sortantes d'une intersection sont en même temps les voies entrantes dans d'autres. En effet, la variation de la phase d'une intersection influence les phases des intersections voisines. Un algorithme de recherche locale va à chaque pas de temps chercher les meilleures phases des intersections sans se soucier ni des optima locaux ni de ces dépendances. En plus, quand toutes les intersections modifient leurs phases en même temps, elles risquent d'osciller continuellement entre deux phases.

Pour casser cette dépendance et les désynchroniser, une deuxième solution consiste à utiliser le principe des algorithmes stochastiques de recherche (DSA) [Morris, 1993; Macready *et al.*, 1996; Weixiong *et al.*, 2005]. Ces algorithmes proposent d'introduire un processus stochastique dans la phase d'optimisation. L'idée est que, à chaque pas de temps, chaque intersection i choisit de garder sa phase (ϕ_i) actuelle ou de l'optimiser selon une probabilité ρ . Ce processus stochastique limite les variations continues des phases et ainsi les dépendances.

7.3.3.2 Besoin du saut $T/2$

Étant donné le signal de période T de chaque intersection, nous savons que la solution optimale de chaque intersection i se trouve dans l'intervalle $[\phi_t(i) - \frac{T}{2}, \phi_t(i) + \frac{T}{2}]$. Pour effectuer une recherche globale, la seule solution pour éviter les plateaux consiste à faire un saut de $\frac{T}{2}$ dans l'espace de recherche. Ceci implique qu'au moins l'une des intersections devrait faire ce saut.

7.3.3.3 Algorithme proposé

Dans l'algorithme DSA 7.3 que nous proposons, notre espace de recherche n'est plus borné par $[\phi_t(i) - \Delta\phi_t, \phi_t(i) + \Delta\phi_t]$, mais plutôt entre $\phi_t(i) - \frac{T}{2*k}$ et $\phi_t(i) + \frac{T}{2*k}$, où k est un entier $\in [1, 256]$ comme le montre la figure 7.10.

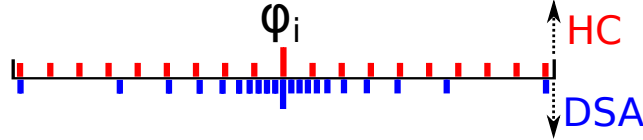


FIGURE 7.10 – Différence entre l'espace de recherche des algorithmes HC et DSA

Algorithme 7.3 : DSA($i, t, \vec{\phi}_t$)

```

1  $\beta$  = valeur prise au hasard dans  $[0, 1[$ ;
2  $\rho$  = constante dans  $[0, 1[$ ;
3  $k \leftarrow 1$ ;
4  $V_{min} \leftarrow Evaluation(\phi_t(i), \vec{\phi}_t)$  ;
5  $\phi_{min}(i) \leftarrow \phi_t(i)$ ;
6 si ( $\beta \leq \rho$ ) alors
7   tant que  $k \leq 256$  faire
8      $\phi_{tmp} \leftarrow \phi_t(i) - \frac{T}{2*k}$ ;
9      $V_{tmp} \leftarrow Evaluation(\phi_{tmp}, \vec{\phi}_t)$  ;
10    si  $V_{tmp} < V_{min}$  alors
11       $V_{min} \leftarrow V_{tmp}$ ;
12       $\phi_{min} \leftarrow \phi_{tmp}$ ;
13     $k \leftarrow k + 1$ ;
14  tant que  $k \geq 1$  faire
15     $\phi_{tmp} \leftarrow \phi_t(i) + \frac{T}{2*k}$ ;
16     $V_{tmp} \leftarrow Evaluation(\phi_{tmp}, \vec{\phi}_t)$  ;
17    si  $V_{tmp} < V_{min}$  alors
18       $V_{min} \leftarrow V_{tmp}$ ;
19       $\phi_{min} \leftarrow \phi_{tmp}$ ;
20     $k \leftarrow k - 1$ ;
21   $\phi_{t+1}(i) \leftarrow \phi_{min}$ ;

```

7.3.4 De l'optimisation hors ligne à l'optimisation en ligne

Nous distinguons deux cas d'utilisations : l'optimisation hors-ligne et l'optimisation en-ligne.

7.3.4.1 L'optimisation hors ligne

L'*optimisation hors ligne* consiste à exécuter l'algorithme d'optimisation avant utilisation du système. Ceci nous permettra de déterminer les meilleures phases à appliquer pour chaque intersection à l'avance. Cette approche est limitée malgré son efficacité. Elle ne permet pas une adaptation autonome au condition du trafic global du réseau et suppose une connaissance des débits fixes du trafic a priori.

7.3.4.2 L'optimisation en ligne

L'*optimisation en ligne* s'applique en cours d'exécution. Elle permet d'adapter en temps réel le déphasage de chaque intersection. Contrairement à l'optimisation hors-ligne, nous pouvons dans ce cas gérer la variabilité du débit. Cependant, cette optimisation peut modifier les instants de passage qui ont été calculés avec précision pour faire passer les véhicules.

Version décentralisée en ligne

Une propriété intéressante des algorithmes proposés est la possibilité de les exécuter, avec un seul agent i à chaque intersection exécutant à chaque pas de temps t une itération de recherche locale et communiquant sa nouvelle phase $\phi_{t+1}(i)$ à ses voisins. Un avantage de ce schéma d'optimisation est sa capacité à s'adapter à l'évolution des conditions du trafic en utilisant uniquement des calculs locaux.

1- Estimation des conditions de trafic Dans ce cas d'optimisation en ligne, les conditions de la circulation ne sont pas connues, et donc chaque intersection doit estimer l'état du trafic. Ici, nous employons une simple approche où chaque agent de contrôle i estime les flux de chacune de ses quatre voies entrantes. Par exemple, pour la voie X , l'agent i compte dans une variable n_m le nombre de véhicules passant à travers l'intersection durant un intervalle temporel Im ; ensuite, il estime le débit de la voie X en utilisant une moyenne glissante

$$\bar{n}_m = (1 - \alpha)\bar{n}_{m-1} + \alpha n_m,$$

où $\alpha \in]0, 1[$ est un paramètre constant qui peut être réglé pour s'adapter à des conditions plus ou moins stables.

2- Exécution et optimisation Cette optimisation en ligne engendre un problème à l'exécution. En effet, la stratégie d'alternance n'est pas destinée à fonctionner avec l'évolution continue des phases.

1. Les véhicules qui ont planifié leur passage à travers la prochaine intersection à l'instant t ne doivent pas changer leur plan à l'instant t' , sauf si ils peuvent réadapter leur profil de vitesse pré-calculée.

2. En plus, modifier l'instant d'arrivée d'un véhicule v à l'intersection i peut conduire à des collisions avec des véhicules qui traversent i avant ou après v (provenant de l'autre route). Cela est d'autant plus probable que la période commune a été calculée de manière à maximiser le débit, c'est à dire, en laissant le moins d'espace libre possible entre les véhicules consécutifs (espace appelé *distance de sécurité*).

Pour surmonter ces limitations, le premier problème est résolu en ne permettant pas de changer l'heure d'arrivée d'un véhicule à sa prochaine intersection une fois celle-ci calculée. Ensuite, on évite d'éventuelles collisions en faisant un compromis entre la variation maximale des phases et la distance de sécurité.

1. Nous distinguons ici entre les phases optimisées et les phases employées pour le contrôle. Nous réduisons la variation maximale des phases, sachant que cette variation devrait rester assez grande pour que les phases s'adaptent rapidement. C'est-à-dire que les intersections ne modifient pas directement leurs phases, mais tendent vers ces phases optimisées, à chaque pas de temps, en respectant cette variation maximale.
2. Nous augmentons aussi la distance de sécurité ϵ qui doit (i) tenir compte de cette variation maximale des phases et (ii) être faible pour minimiser la période, et ainsi maximiser le débit.

En plus, dans les algorithmes 7.2 et 7.3 les phases ne sont pas contraintes à rester dans l'intervalle $[0, T[$. En pratique, les phases appliquées seront plutôt dans $[0, +\infty[$ afin de faciliter la tâche d'allocation d'intervalles de passages des agents de contrôle.

7.4 Stratégie premier arrivé premier servi (FCFS)

Dans le but de comparer notre approche avec une approche typique de l'état de l'art, nous proposons dans cette section une solution qui consiste à faire passer les véhicules dans leur ordre d'arrivée, d'où l'appellation FCFS (First Come First Served). Cette stratégie opportuniste est fondée sur la stratégie dite de réservation de [Dresner et Stone \[2005\]](#) présentée dans le chapitre 4.

L'implémentation de l'approche par réservation dans notre outil telle que décrite dans [\[Dresner et Stone, 2005\]](#) est difficile, car nous n'avons ni la même modélisation ni toutes les données nécessaires à son implémentation. Ce que nous présentons ici est une approche fondée sur la même idée des réservations. La différence entre l'approche par réservation des temps de passage et l'approche FCFS est que cette dernière propose toujours des créneaux de passage fixe dans le temps, toujours selon un signal périodique.

En effet, nous ne réservons plus de créneaux de passage d'une demi-période pour chaque route du réseau, mais chaque intersection alloue la demi-période la plus proche à la route qui a vu venir un véhicule en premier. Donc, la route qui a un véhicule à faire passer utilise la demi-période disponible la plus proche.

Prenons comme exemple une intersection constituée du croisement de deux routes A et B à deux voies. Nous appelons $A1$, $A2$, $B1$ et $B2$ les voies des routes. La figure 7.11 illustre un exemple du principe de l'approche FCFS.

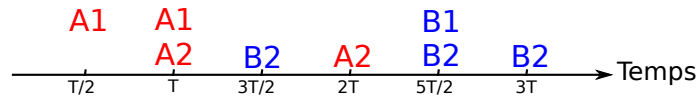


FIGURE 7.11 – Principe de fonctionnement de l'approche First Come First Served

A la première demi-période, la première voie de la route A fait passer un véhicule. Ensuite, à la deuxième demi-période (c'est à dire à T), les deux voies de la routes A font passer un véhicule chacune. Puis, à la troisième demi-période, c'est au tour de la route B et ainsi de suite.

7.5 Expérimentations

Dans cette section, nous présentons les résultats expérimentaux qui évaluent les bénéfices de l'optimisation de phases pour notre approche d'alternance, et nous la comparons à la stratégie FCFS (First Come First Served).

Toutes les expérimentations ont été réalisées sur un réseau composé de 6 routes (12 voies) se croisant en 12 intersections. Deux groupes de 6 voies, et quatre conditions de trafic sont créées. Nous assignons un fort débit ($15 \text{ véhicules} \cdot \text{minute}^{-1}$) ou faible débit ($5 \text{ véhicules} \cdot \text{minute}^{-1}$) à chaque groupe de voies. Ces conditions de trafic seront notées 15–15, 15–5, 5–15, et 5–5. La figure 7.12 est une capture d'écran.

Nous mesurons les critères d'optimisation uniquement à l'intérieur du réseau, c'est-à-dire directement après que les véhicules soient passés à travers la première intersection et jusqu'après la dernière intersection. Ceci nous permet de réduire les variances de nos estimations, sans pour autant détériorer les mesures de performance, car notre approche n'a pas d'influence sur ce qui se passe à l'extérieur.

7.5.1 Paramètres de simulation

Les paramètres de simulation de notre réseau de 12 routes sont les suivants :

- le rayon d'action R des agents de contrôle est fixé à $125m$, tandis que r_0 est déterminé pour chaque intersection à partir l'angle de l'intersection θ et de l'équation 6.12);
- pour rappel, chaque source injecte des véhicules suivant un processus Bernoulli de paramètre $\lambda = \frac{1}{D}$ où D est la durée moyenne, en secondes, entre deux injections consécutives (par exemple, si $D = 4$, ceci implique un débit moyen de 15 véhicules par minute);
- le pas de simulation est de $0.05s$.

Dans ce cas, la période commune du réseau de la figure 7.12 est $T_{max} = 3.5s$. Nous avons fixé les paramètres des algorithmes d'optimisation suivants :

- l'intervalle de recherche $\Delta\phi = 0.05s$ et le pas $\delta\phi = \frac{\Delta\phi}{10}$ pour l'algorithme Hill-Climbing, et,
- la probabilité de mise à jour des agents $\rho = 0.2$ pour l'algorithme DSA.

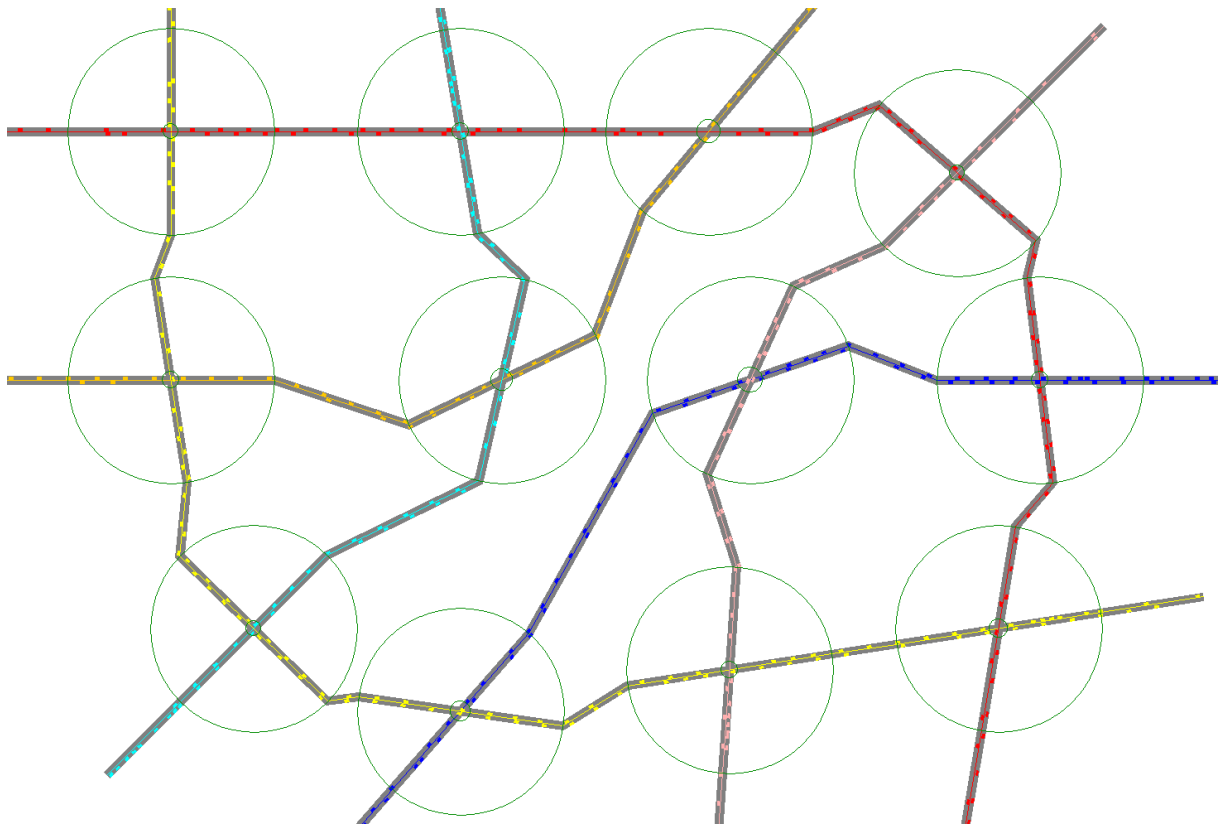


FIGURE 7.12 – Réseau de simulation utilisé avec 12 intersections

Une vidéo montrant le simulateur peut être consulté à l'adresse suivante : http://www.loria.fr/%7emtlig/videos/12_intersections.avi.

7.5.2 Courbes d'optimisation obtenues

Afin de constater la différence entre l'optimisation HC et DSA, nous avons tracé les évolutions des critères locaux des douze intersections en considérant, pour chacune d'elles, sa phase initiale fixée aléatoirement comme point d'origine. Toutes les intersections optimisent le critère énergie. Nous avons testé trois configurations différentes en mode hors ligne où nous avons utilisé les différentes phases initiales prises aléatoirement.

Les figures 7.13 *a*, *b* et *c* illustrent l'évolution des critères locaux (énergie) des douze intersections (notées ici de i_1 à i_{12}) en fonction des itérations en utilisant l'algorithme Hill-Climbing et les figures 7.13 *d*, *e* et *f* en utilisant l'algorithme DSA.

Nous remarquons que les deux algorithmes se stabilisent rapidement, à peine après 25 itérations. A chaque itération, l'algorithme HC cherche une meilleure solution dans son intervalle de recherche, s'il ne trouve pas, il se contente de la solution actuelle. Cependant, l'algorithme DSA n'hésite pas à détériorer la phase d'une intersection pour améliorer la solution des autres (dans les figures 7.13 *d*, *e* et *f*, quand une intersection accroît son critère au lieu de le réduire).

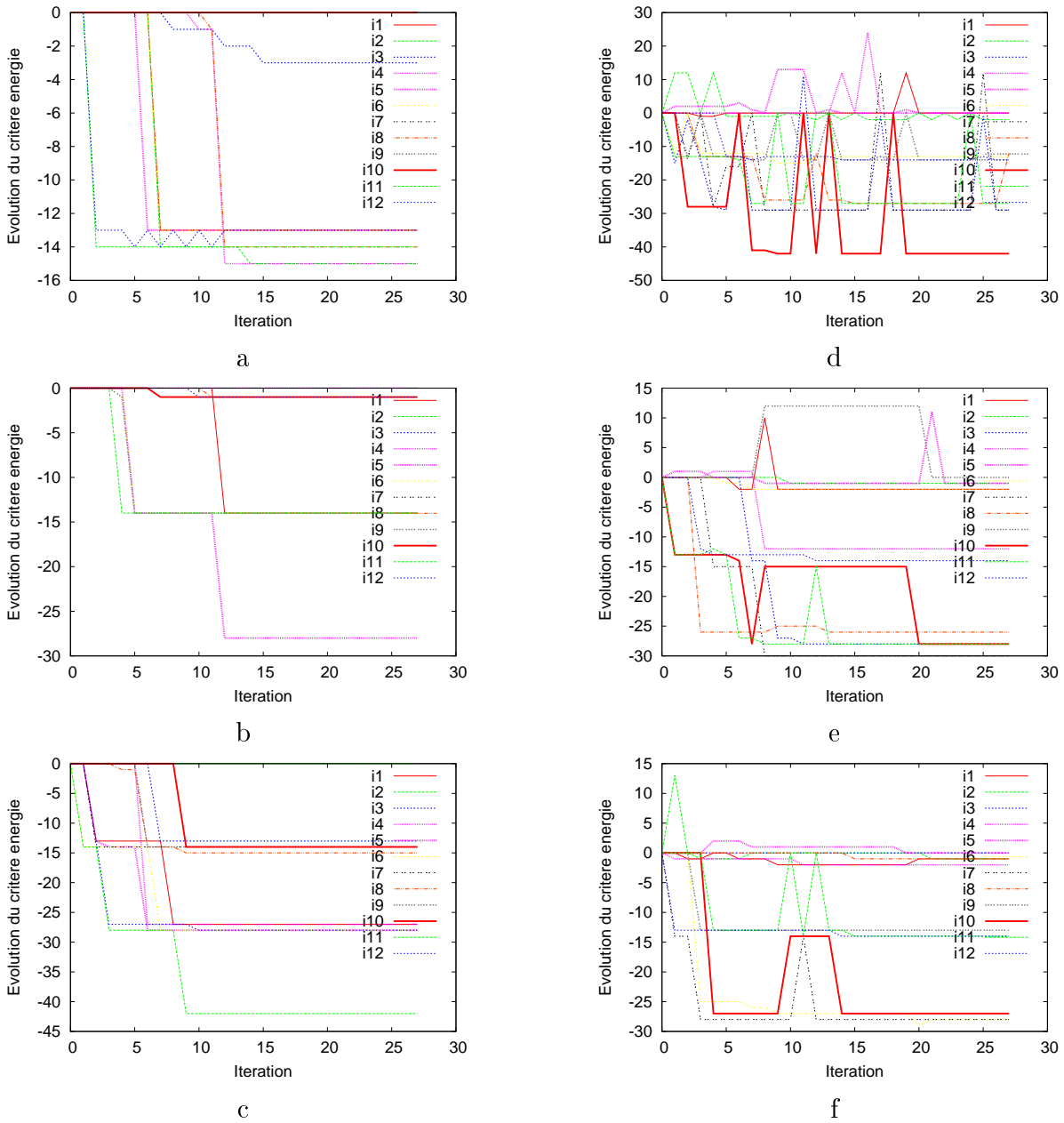


FIGURE 7.13 – Comparaison entre l'évolution des critères locaux (énergie) en fonction des itérations dans HC (a, b, c) et dans DSA (d, e, f)

7.5.3 Optimisation hors ligne

Nous considérons d'abord le cas de l'optimisation hors ligne, en comparant Alt et FCFS. Nous avons évalué les stratégies FCFS et Alt dans les cas :

1. asynchrone du chapitre 6 avec des périodes locales $T_{min}(i)$ et des phases aléatoires,
2. synchrone avec des périodes communes T et des phases aléatoires,
3. avec des périodes communes T et des phases optimisées avec l'algorithme HC, et,
4. uniquement pour Alt, nous avons ajouté des phases optimisées avec DSA.

Chacune des 7 stratégies résultantes est utilisée avec 4 conditions de trafic durant 10 expérimentations d'une heure (temps de simulation).

Les résultats sont présentés dans le tableau 7.1, où nous mesurons l'énergie cinétique moyenne consommée par véhicule.

TABLE 7.1 – Energie cinétique moyenne consommée par véhicule, utilisant FCFS dans ses versions asynchrone, synchrone ou optimisé HC, et Alt dans ses versions asynchrone, synchrone, optim HC et optim DSA ; Les injections sont mesurées en véhicules/minute

Injections		(10k veh)	(7k veh)	(7k veh)	(3,5k veh)	Moyenne
		15-15	15-5	5-15	5-5	
FCFS	async	137,6 ±2,4	104,6 ±2,1	102,3 ±2,2	60,3 ±0,6	101,2
	sync	178,8 ±4,6	130,7 ±5,7	128,6 ±5,7	67,4 ±3,2	126,4
	optim HC	176,9 ±2,5	131,6 ±4,7	129,8 ±3,5	66,1 ±4,1	126,1
Alt	async	111,1 ±1,3	107,1 ±0,8	102 ±0,8	85,9 ±0,8	101,5
	sync	92,6 ±6,4	87,7 ±10,2	88,6 ±8,6	88,5 ±5,1	89,4
	optim HC	83,8 ±7,4	80,2 ±7,4	74,7 ±5,6	83,7 ±12,4	80,6
	optim DSA	71 ±7,7	66,6 ±4,6	62,1 ±4,1	71,6 ±6,8	67,8

Une première observation est que la synchronisation du réseau (utiliser une période T commune) améliore déjà la circulation pour Alt, sauf dans des conditions de faible trafic, alors qu'elle dégrade toujours la circulation pour FCFS (qui bénéficie de son opportunisme dans le cas asynchrone).

Concernant les intersections synchrones, les phases optimisées permettent à Alt de réduire la consommation d'énergie. Tout cela n'a aucune incidence sur FCFS, ainsi les phases optimales de Alt ne sont pas optimales pour lui.

En plus, comme FCFS est opportuniste, il économise de l'énergie dans des conditions de circulation à faible densité (5-5). Néanmoins, plus la circulation est dense, plus des conflits se produisent aux intersections, et donc moins de possibilités peuvent être exploitées par FCFS, ce qui signifie que les véhicules sont souvent obligés de ralentir. Dans les mêmes conditions de circulation très dense, Alt optimisé par HC est plus avantageux : il maintient une faible consommation d'énergie.

Par ailleurs, nous remarquons que l'utilisation de l'algorithme DSA améliore les résultats par rapport à Alt HC, c'est à dire que DSA arrive mieux à éviter plus d'optima locaux que HC.

Les mesures avec le critère des temps de traversée présentées dans le tableau 7.2 conduisent aux mêmes conclusions parce que, dans les deux cas, les critères essaient de garder les véhicules à leur vitesse par défaut autant que possible.

TABLE 7.2 – Moyenne des temps de traversées par véhicule, utilisant FCFS dans ses versions asynchrone, synchrone ou optimisé HC, et Alt dans ses versions asynchrone, synchrone, optim HC et optim DSA ; Les injections sont mesurées en véhicules/minute

Injections		(10k veh) 15–15	(7k veh) 15–5	(7k veh) 5–15	(3,5k veh) 5–5	Moyenne
FCFS	async	143,5 ±0,8	139,9 ±0,4	139,9 ±0,4	136,2 ±0,7	139,9
	sync	149,2 ±1	142,5 ±0,6	142,6 ±0,9	136,4 ±0,5	142,7
	optim HC	148,6 ±0,6	142,9 ±1,2	142,8 ±0,6	136,5 ±0,6	142,7
Alt	async	141,2 ±0,4	140,6 ±0,3	140,3 ±0,3	138,3 ±0,6	140,1
	sync	138,3 ±0,6	138,2 ±0,8	137,9 ±0,7	138,3 ±1,1	138,2
	optim HC	138,3 ±0,4	138,0 ±0,9	137,3 ±0,4	137,9 ±0,6	137,9
	optim DSA	137,3 ±0,4	136,9 ±0,5	136,0 ±0,6	136,9 ±0,8	136,8

7.5.4 Optimisation en ligne

Dans cette section, nous présentons les résultats de l'optimisation en ligne. Nous considérons ici des simulations sur une durée de quatre heures durant laquelle les quatre conditions de trafic utilisées précédemment sont appliquées une heure chacune, et nous comparons :

- FCFS avec des périodes locales et des phases aléatoires,
- Alt avec des phases optimisées hors ligne avec HC sur la base du débit moyen pendant les 4 heures : 10–10,
- Alt avec des phases optimisées hors ligne avec DSA sur la base du débit moyen pendant les 4 heures : 10–10,
- Alt avec des phases optimisées en ligne avec HC et utilisant 6 valeurs du paramètre d'apprentissage α , et,
- Alt utilisant l'algorithme DSA en ligne et utilisant 6 valeurs du paramètre d'apprentissage α .

Chacune des stratégies résultantes est utilisée durant 10 expérimentations de 4 heures (temps de simulation). Comme indiqué précédemment, chaque agent intersection i estime seul les flux sur ses voies, communique sa phase seulement avec ses voisins, et optimise sa propre phase $\phi(i)$.

Le tableau 7.3 présente les estimations empiriques de la consommation d'énergie et des temps de traversée par véhicule en utilisant les algorithmes HC (algorithme 7.2) et DSA (algorithme 7.3).

La version en ligne de la stratégie Alt améliore les résultats de la version hors ligne et est nettement plus efficace que FCFS (avec une plus grande marge dans ce cas) . Cela démontre clairement que l'estimation en ligne des débits des flux et l'optimisation

TABLE 7.3 – Moyenne de l'énergie consommée et des temps de traversées par véhicule utilisant Alt avec des phases optimisées hors ligne, Alt avec des phases optimisées en ligne, et FCFS avec des phases aléatoires

Alg. (α). 10^{-3}	Alt on-line						Alt off-line	FCFS
	(0, 195312)	(0, 390625)	(0, 78125)	(1, 5625)	(6, 25)	(25)		
HC énergie	82 \pm 9,1	81,3 \pm 8	78 \pm 3,8	78,3 \pm 7,3	78,8 \pm 6	80,8 \pm 8	80,1 \pm 6,9	101,2
DSA énergie	70 \pm 9	70,4 \pm 7,5	68,7 \pm 5,4	69,3 \pm 4,9	69,5 \pm 7,2	73,2 \pm 5	70,4 \pm 7,6	101,2
HC temps	138,1 \pm 0,7	138 \pm 0,5	137,8 \pm 0,2	137,9 \pm 0,5	138 \pm 0,3	138,1 \pm 0,4	138,2 \pm 0,4	139,9
DSA temps	137,5 \pm 0,6	137,5 \pm 0,5	137,4 \pm 0,3	137,4 \pm 0,2	137,4 \pm 0,6	138,8 \pm 2,8	137,6 \pm 0,3	139,9

des phases sont efficaces sur ce réseau, **conduisant à une approche adaptative aux conditions du trafic**. Les meilleurs résultats sont obtenus quand $\alpha = 0,78125 \cdot 10^{-3}$, bien que la différence avec les autres valeurs reste d'une faible importance.

Nous remarquons aussi que la version en ligne de Alt utilisant l'algorithme DSA améliore nettement sa version Hill-Climbing. Ceci prouve que l'algorithme DSA (comme HC) fonctionne aussi bien en ligne que hors ligne, et évite toujours mieux les optima locaux. Les meilleurs résultats sont aussi obtenus quand $\alpha = 0,78125 \cdot 10^{-3}$, même s'il n'y a pas une grande différence avec les valeurs $\alpha = 1,5625 \cdot 10^{-3}$ ou $\alpha = 6,25 \cdot 10^{-3}$.

7.6 Conclusion

Dans ce chapitre, nous avons d'abord proposé une approche pour optimiser l'écoulement de véhicules dans un réseau de transport sans les arrêter aux intersections. Cette approche améliore de façon significative les flux tout en utilisant simplement des intersections indépendantes sans arrêt de véhicules. Elle est aussi meilleure qu'une stratégie plus opportuniste, "premier arrivé premier servi" (FCFS), qui empêche la formation d'ondes vertes, dans des conditions de trafic dense. De plus, l'algorithme peut être facilement adapté pour fonctionner en ligne, chaque intersection estimant en permanence les conditions de circulation locales et l'optimisation de sa phase, nécessitant uniquement de communiquer cette phase à ses voisins. Cette version en ligne améliore de manière significative la version hors ligne, montrant sa capacité d'adaptation à l'évolution des flux.

Ensuite, à cause de la nature de notre problème, nous avons montré que nous risquons en optimisant de converger vers des optima locaux et des "plateaux", et nous avons alors proposé une amélioration de la première version par l'utilisation d'un autre algorithme de recherche locale, plus précisément, en utilisant l'algorithme DSA, une variante de Hill-Climbing spécialement conçue pour une optimisation distribuée.

Nous pensons que de nouvelles expériences pourraient permettre d'approfondir les connaissances et les limites de cette approche, en évaluant différents réseaux et différents scénarios. Nous présentons les perspectives de ce travail dans le chapitre qui suit.

Chapitre 8

Conclusion de la thèse

Les véhicules autonomes sont aujourd’hui considérés comme une approche prometteuse pour améliorer le trafic routier, tant sur le plan de la sécurité que celui des économies d’énergie. Dans cette thèse, nous avons étudié la coordination et la résolution de conflits entre véhicules autonomes sur des réseaux routiers denses en intersections.

Les deux contextes étudiés dans cette thèse, partage d’une voie et croisement de voies sans arrêts, visaient tous deux à fluidifier globalement le trafic. Dans les deux cas, nous avons utilisé des règles de coordination réactive complétées d’un processus d’optimisation. Cette approche nous a permis de proposer des solutions décentralisées nécessitant peu de calculs et permettant d’agir en temps réel. Le rapprochement des systèmes multi-agents réactifs et des techniques d’optimisation distribuée est une démarche originale dans le cadre de la régulation du trafic des véhicules autonomes.

Nous synthétisons dans ce qui suit les contributions de chaque partie et précisons pour chacune les principales perspectives.

8.1 Partage d’une voie par deux files de véhicules

8.1.1 Synthèse des contributions

Dans la première partie de cette thèse nous avons abordé le problème du partage de l’espace entre files de véhicules ou plus généralement entre agents mobiles (par exemple des robots). Pour cela, nous avons proposé des modèles reposant sur des comportements de coordination locale. Nous avons tout d’abord proposé un modèle utilisant un mécanisme systématique d’alternance puis un modèle intégrant un critère d’optimisation dépendant des retards des véhicules (OLC). Afin de conserver un processus de décision local et “en ligne”, ces mécanismes sont appliqués seulement entre les véhicules en tête de files.

L’étude expérimentale a permis de montrer la capacité de régulation des conflits (des congestions) de ces comportements. L’approche simple OLC fait bien mieux qu’une alternance “stricte” et les trois variantes du critère d’optimisation donnent des comportements

de régulation assez distincts. Les phénomènes d’embouteillage, qui sont des phénomènes émergents indésirables, sont ici traités localement, donc indépendamment de tout système de planification extérieur, et en temps réel. L’introduction de communications locales des retards permet d’améliorer significativement la stratégie d’alternance utilisée couramment par les conducteurs humains.

Plus globalement, l’ensemble des résultats expérimentaux montre qu’une approche centrée sur des règles de coordination locale s’avère performante pour réguler le trafic en un lieu de conflit pour l’accès à un espace limité. Ce travail a donné lieu à deux publications internationales dans les conférences ICTAI et PAAMS [Tlig *et al.*, 2012, 2013b].

8.1.2 Perspectives

Nous envisageons d’intégrer les retards d’un plus grand nombre de véhicules présents dans les files pour améliorer encore la gestion du trafic, ce qui posera toutefois de nouveaux problèmes de complexité et de communication entre véhicules.

Nous souhaitons également poursuivre cette étude en généralisant les modèles proposés à un nombre quelconque de files.

8.2 Optimisation d’un réseau d’intersections

8.2.1 Synthèse des contributions

Synchronisation de flux de véhicules dans une intersection

Dans la deuxième partie de cette thèse, nous nous sommes concentrés sur les problèmes de congestion dans les espaces qui doivent être nécessairement partagés : les intersections.

Nous avons abordé le problème du croisement automatique des véhicules sur un réseau d’intersections. Pour cela, nous avons d’abord proposé une stratégie de contrôle local aux intersections. Elle consiste à espacer les véhicules de sorte qu’ils puissent se croiser sans être arrêtés en déphasant leur signal périodique de passage. Ainsi les véhicules ne subissent potentiellement qu’un léger ralentissement, si on les fait arriver au bon moment, ce qui est plus performant en temps et en consommation d’énergie que de les arrêter. Nous avons introduit sur ce principe un agent de contrôle en chaque intersection qui utilise uniquement ses perceptions locales du trafic. Il détermine le profil d’accélération pour le croisement de chaque véhicule en fonction de sa distance et de la priorité assignée à sa route. L’étude expérimentale a permis de montrer la capacité de régulation du trafic aux croisements, et le gain important en temps par rapport à un système classique de feux de circulation. Néanmoins, comme toute approche, cette synchronisation a ses limites. Par exemple, si trop de véhicules arrivent en même temps à une intersection, on ne peut plus assurer le passage alterné. En réalité, dans ce cas de figure, l’agent croisement devra arrêter tous les véhicules qui attendront alors leur tour pour passer un par un. Ce travail a été publié

dans la conférence nationale RJCIA [Tlig *et al.*, 2013a] (best paper) et dans la conférence internationale ICALT [Tlig *et al.*, 2014a].

Synchronisation d'intersections dans un réseau

Dans cette partie, nous nous sommes focalisés sur la synchronisation des intersections dans les réseaux de transports. Cet intérêt vient de la volonté de fluidifier le trafic au niveau du réseau en partant des intersections définies plus haut, et du fait que peu de travaux dans la littérature s'intéressent à ce problème dans le cadre des véhicules autonomes (chapitre 4).

Nous avons proposé une solution de type "optimisation distribuée" dont on a conçu d'abord une version simple (Hill-Climbing), déjà efficace mais avec certaines limites, puis une version stochastique distribuée (DSA) qui répond à ces problèmes et améliore nettement les résultats. Ces limites proviennent de la forme des fonctions périodiques de chaque intersection, forme qui favorise les optima locaux.

Les deux versions améliorent de façon significative les flux tout en utilisant simplement des intersections indépendantes sans arrêt de véhicules. Elles sont aussi meilleures qu'une stratégie plus opportuniste, premier arrivé premier servi, dans des conditions de trafic dense (comme proposé par Dresner *et Stone* [2005]).

Les algorithmes d'optimisation proposés ont pu être facilement modifiés pour s'exécuter en ligne et pour permettre l'exploitation des solutions. Chaque intersection doit estimer en permanence les conditions de circulation locales et optimiser sa phase, ce qui nécessite de communiquer seulement cette phase à ses voisins. Les versions en ligne ont montré leur capacité d'adaptation aux variations des flux de véhicules et leur compétitivité par rapport à l'approche FCFS²⁰. Les résultats de cette partie ont été publiés dans la conférence internationale ECAI [Tlig *et al.*, 2014b].

8.2.2 Perspectives

Les travaux réalisés dans la deuxième partie amènent des perspectives assez diverses. Voici les principaux sujets que nous envisageons de développer :

Amélioration de la version actuelle

D'autres expériences devront être menées sur des réseaux et des scénarios plus variés pour mieux identifier les avantages et limites de l'approche. Par ailleurs, des éléments de l'approche pourront être améliorés telles que l'estimation des conditions de trafic.

20. premier arrivé premier servi (introduite en section 7.4)

Traitement de scénarios plus complexes

Notre gestion des croisements aux intersections repose sur une stratégie d'alternance périodique (Alt) qui comporte deux limites principales : (1) à une intersection donnée, toutes les voies utilisent le même laps de temps (période), alors que les flux peuvent être très différents d'une voie à l'autre, et (2) les véhicules ne peuvent pas changer de voie (tourner).

Pour la première limitation, on peut simplement utiliser un modèle d'alternance n/m à chaque intersection plutôt que $1/1$, où n est le nombre de véhicules qui passe à une période pour une route, en alternance avec m pour l'autre route.

Comme dans la plupart des approches par feux de circulation (voir chapitre 4), la seconde limitation peut être traitée en utilisant, à chaque intersection, des cycles composés de quatre périodes au lieu de deux : deux périodes afin de laisser les véhicules de chaque route avancer tout droit ou tourner à droite, plus deux périodes pour laisser les véhicules de chaque route tourner à gauche. En outre, les voies doivent être dédoublées (huit voies entrantes au lieu de quatre) avant les intersections pour éviter les conflits entre les véhicules de la même voie qui veulent traverser vers des directions différentes.

Dans les deux cas, les calculs sont rendus plus complexes pour établir les différentes périodes de passage (correspondant aux quatre périodes), comme pour optimiser les phases (utilisation d'évaluations analytiques plus complexes).

Optimisation de FCFS

Comme observé précédemment, l'application des phases optimisées pour Alt tout en utilisant FCFS pour le contrôle à chaque intersection dégrade les résultats. Donc, il serait intéressant de savoir si les phases pourraient être optimisées spécifiquement pour la stratégie opportuniste FCFS. Dans ce cas, l'optimisation des déphasages doit être étudiée à nouveau.

Quand un véhicule quitte l'intersection i en direction de la prochaine intersection j , son passage par j dépend des conflits potentiels avec d'autres véhicules, et est donc difficile à prévoir. En conséquence, il est plus difficile de calculer analytiquement un critère de performance pour FCFS sur le segment $i \rightarrow j$ étant données $\phi(i)$ et $\phi(j)$. Cela implique (1) que les simulations peuvent être nécessaires pour évaluer un ensemble de phases $\vec{\phi}$, et (2) que l'optimisation distribuée va être complexe en raison des interactions complexes entre les intersections.

Utilisation combinée de Alt et FCFS

Un autre problème intéressant consiste à savoir comment tirer le meilleur des deux approches FCFS et Alt. Une première idée serait de passer de l'une à l'autre selon les conditions générales de la circulation. Mais on pourrait aussi imaginer des réseaux de

transport hétérogènes dans lesquels certaines des intersections utilisent un contrôleur Alt (trafic dense), et d'autres disposent d'un contrôleur FCFS (trafic faible).

Étudier d'autres algorithmes d'optimisation

Dans le chapitre 7, nous avons comparé l'optimisation des phases des intersections avec les algorithmes HC et DSA. L'étude expérimentale montre que DSA améliore considérablement (surtout pour la consommation d'énergie) les performances des véhicules dans le réseau de transport. Néanmoins, cette comparaison ne nous permet pas de prouver que cette solution est optimale. Pour cela, nous envisageons d'étudier d'autres algorithmes d'optimisation qui, peut-être, pourront optimiser les phases des intersections mieux que les solutions déjà proposées.

Perspectives expérimentales

Enfin, pour évaluer nos modèles dans des conditions routières réalistes, nous envisageons de connecter notre outil à un simulateur réaliste qui tient compte de la physique des véhicules, tel que SUMO (voir chapitre 5) ou ScannerStudio utilisé dans le projet InTraDE. Dans cette même problématique, une autre idée qui nous motive serait de porter nos études sur des robots réels pour vérifier la robustesse de nos propositions et pour comparer aussi les comportements obtenus par rapport aux résultats théoriques.

Annexe A

Détails de calcul de la synchronisation de flux de véhicules dans une intersection

A.1 Adaptation des vitesses

Dans cette première partie de l'annexe, nous détaillons les calculs du profil d'accélération effectués par chaque intersection lors de la détection d'un véhicule pour la première fois. D'après la figure A.1, nous avons le temps de parcours :

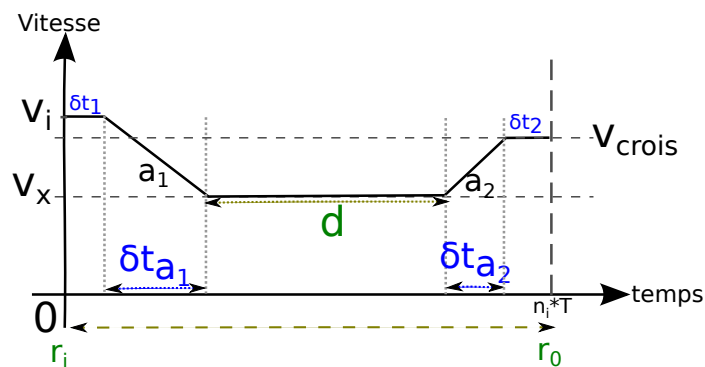


FIGURE A.1 – Profil d'accélération

$$\begin{aligned}
 t_p &= \delta t_1 + \delta t_2 + \delta t_{a_1} + \delta t_{a_2} + \frac{d}{v_x} \text{ où} \\
 \delta t_{a_1} &= \frac{v_x - v_i}{a_1}, \\
 \delta t_{a_2} &= \frac{v_{crois} - v_x}{a_2}, \\
 \delta t_1 &= \frac{l + \frac{\lambda}{2}}{v_{crois}}, \text{ et} \\
 \delta t_2 &= \frac{l + \frac{\lambda}{2}}{v_i}.
 \end{aligned}$$

Nous pouvons écrire la distance parcourue à faible vitesse d sous cette forme :

$$\begin{aligned}
 d &= (t_p - \delta t_1 - \delta t_2 - \delta t_{a_1} - \delta t_{a_2}) v_x \\
 &= v_x t_p - v_x \delta t_1 - v_x \delta t_2 - v_x \frac{v_x - v_i}{a_1} - v_x \frac{v_{crois} - v_x}{a_2} \\
 &= v_x t_p - v_x \delta t_1 - v_x \delta t_2 - \frac{v_x^2 - v_x v_i}{a_1} - \frac{v_x v_{crois} - v_x^2}{a_2} \\
 &= v_x t_p - v_x \delta t_1 - v_x \delta t_2 - \frac{a_2 v_x^2 - a_2 v_x v_i + a_1 v_x v_{crois} - a_1 v_x^2}{a_1 a_2} \\
 &= v_x t_p - v_x \delta t_1 - v_x \delta t_2 - \frac{a_2 v_x^2}{a_1 a_2} + \frac{a_2 v_x v_i}{a_1 a_2} - \frac{a_1 v_x v_{crois}}{a_1 a_2} + \frac{a_1 v_x^2}{a_1 a_2} \\
 &= v_x (t_p - \delta t_1 - \delta t_2 + \frac{a_2 v_i - a_1 v_{crois}}{a_1 a_2}) + v_x^2 (\frac{a_1 - a_2}{a_1 a_2}).
 \end{aligned}$$

D'autre part la distance r_p séparant r_i de r_0 peut être écrite sous la forme :

$$\begin{aligned}
 r_p &= 2l + \lambda + \int_0^{\delta t_{a_1}} (v_i + a_1 t) dt + \int_0^{\delta t_{a_2}} (v_{crois} - a_2 t) dt + d \\
 &= 2l + \lambda + \left[v_i \delta t_{a_1} + \frac{a_1 \delta t_{a_1}^2}{2} \right] + \left[v_{crois} \delta t_{a_2} - \frac{a_2 \delta t_{a_2}^2}{2} \right] + d.
 \end{aligned}$$

Donc en utilisant les expressions des distances d et r_p , nous avons :

$$\begin{aligned}
d &= r_p - 2l - \lambda - \left[v_i \delta t_{a_1} + \frac{a_1 \delta t_{a_1}^2}{2} \right] - \left[v_{crois} \delta t_{a_2} - \frac{a_2 \delta t_{a_2}^2}{2} \right] \\
&= r_p - 2l - \lambda - \frac{v_i(v_x - v_i)}{a_1} - \frac{a_1(v_x - v_i)^2}{2a_1^2} - \frac{v_{crois}(v_{crois} - v_x)}{a_2} + \frac{a_2(v_{crois} - v_x)^2}{2a_2^2} \\
&= r_p - 2l - \lambda - \frac{v_i(v_x - v_i)}{a_1} - \frac{(v_x - v_i)^2}{2a_1} - \frac{v_{crois}(v_{crois} - v_x)}{a_2} + \frac{(v_{crois} - v_x)^2}{2a_2} \\
&= r_p - 2l - \lambda - \frac{1}{2a_1 a_2} \left[2a_2 v_i(v_x - v_i) + a_2(v_x - v_i)^2 + \right. \\
&\quad \left. 2a_1 v_{crois}(v_{crois} - v_x) - a_1(v_{crois} - v_x)^2 \right] \\
&= r_p - 2l - \lambda - \frac{1}{2a_1 a_2} \left[2a_2 v_i v_x - 2a_2 v_i^2 + a_2 v_x^2 - 2a_2 v_x v_i + a_2 v_i^2 + \right. \\
&\quad \left. 2a_1 v_{crois}^2 - 2a_1 v_{crois} v_x - a_1 v_{crois}^2 + 2a_1 v_x v_{crois} - a_1 v_x^2 \right] \\
&= r_p - 2l - \lambda - \frac{-a_2 v_i^2 + a_2 v_x^2 + a_1 v_{crois}^2 - a_1 v_x^2}{2a_1 a_2} \\
&= r_p - 2l - \lambda - v_x^2 \frac{a_2 - a_1}{2a_1 a_2} + \frac{a_2 v_i^2 - a_1 v_{crois}^2}{2a_1 a_2}.
\end{aligned}$$

D'où l'équation

$$\begin{aligned}
&v_x(t_p - \delta t_1 - \delta t_2 + \frac{a_2 v_i - a_1 v_{crois}}{a_1 a_2}) + v_x^2 \left(\frac{a_1 - a_2}{a_1 a_2} \right) \\
&- (r_p - 2l - \lambda - v_x^2 \frac{a_2 - a_1}{2a_1 a_2} + \frac{a_2 v_i^2 - a_1 v_{crois}^2}{2a_1 a_2}) = 0
\end{aligned}$$

que l'on peut réécrire :

$$\begin{aligned}
&v_x^2 \left(\frac{a_1 - a_2}{2a_1 a_2} \right) + v_x(t_p - \delta t_1 - \delta t_2 + \frac{a_2 v_i - a_1 v_{crois}}{a_1 a_2}) \\
&- (r_p - 2l - \lambda + \frac{a_2 v_i^2 - a_1 v_{crois}^2}{2a_1 a_2}) = 0
\end{aligned}$$

A.2 Rayon minimum r_0

Considérons un véhicule qui entre dans une intersection (composée de deux routes à une seule voie), comme illustré dans la figure A.2.

Ce véhicule doit être à vitesse constante v_{crois} dès qu'il pénètre dans le losange. Parce que l'agent de contrôle de cette intersection utilise un cercle de rayon r_0 pour délimiter ce qui est à l'intérieur de l'intersection, un choix sûr serait de mettre r_0 égale à la plus longue demi-diagonale (d_1 ou d_2) de ce losange. Ainsi $r_0 = \max(d_1, d_2)$. Cette plus longue demi-diagonale dépend de l'angle de l'intersection θ .

A partir de la figure A.2, nous pouvons déterminer $d_1 = h \cos \frac{\theta}{2}$ et $d_2 = h \cos \frac{\pi - \theta}{2}$, où

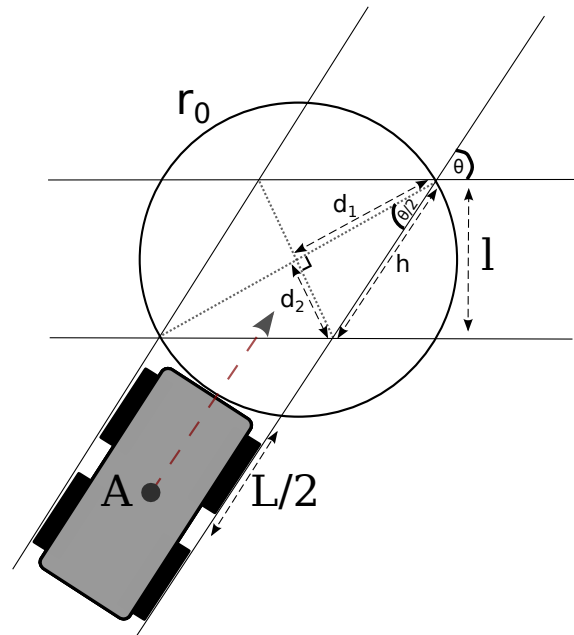


FIGURE A.2 – Le rayon minimum r_0 (pour deux routes à une seule voie chacune)

$h = \frac{l}{\sin \theta}$. La longueur $\frac{L}{2}$ est aussi ajoutée parce que le point de référence du véhicule est son centre de gravité. Le rayon minimum r_0 est donc :

$$r_0 = \frac{L}{2} + \frac{l \max(\cos \frac{\theta}{2}, \cos \frac{\pi-\theta}{2})}{\sin \theta}.$$

Pour adapter cette formule aux deux routes à deux voies, nous devons considérer toute la largeur de la route – notée $\Gamma = 2l + \lambda$ – au lieu de considérer uniquement la largeur d'un véhicule, ce qui conduit à la nouvelle formule du rayon minimum r_0 qui est :

$$r_0 = \frac{L}{2} + \frac{\Gamma \max(\cos \frac{\theta}{2}, \cos \frac{\pi-\theta}{2})}{\sin \theta}.$$

La figure A.3 montre la variation du rayon minimum r_0 en fonction de l'angle de croisement des routes (intersection) θ et de la largeur de la route (10m).

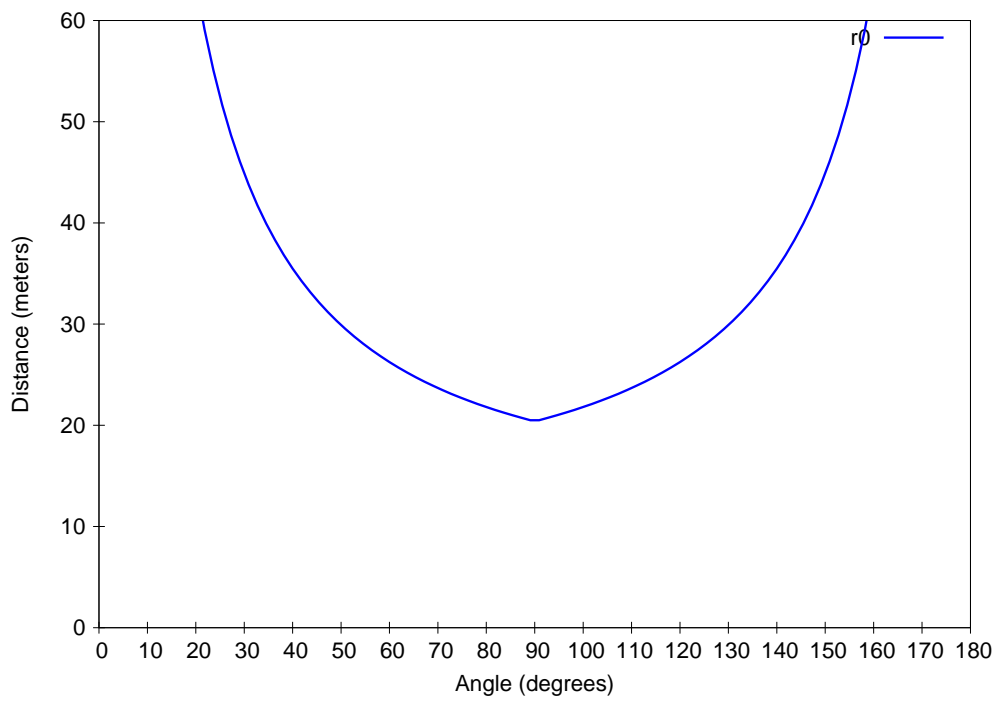


FIGURE A.3 – r_0 en fonction de θ pour deux routes à deux voies chacune et d'une largeur de $10m$

Annexe B

Exemple d'un fichier définissant un réseau de transport

```
//Définition des coordonnées de cinq intersections
```

```
<Agents>
```

```
<Object>
```

```
  lasttrafficsimulation.Agent
```

```
  <Methods>
```

```
    setX:=290
```

```
    setY:=290
```

```
<\Object>
```

```
<Object>
```

```
  lasttrafficsimulation.Agent
```

```
  <Methods>
```

```
    setX:=680
```

```
    setY:=350
```

```
<\Object>
```

```
<Object>
```

```
  lasttrafficsimulation.Agent
```

```
  <Methods>
```

```
    setX:=340
```

```
    setY:=650
```

```
<\Object>
```

```
<Object>
```

```
  lasttrafficsimulation.Agent
```

```
  <Methods>
```

```
    setX:=720
```

```
        setY:=690
<\Object>

<Object>
  lasttrafficsimulation.Agent
  <Methods>
    setX:=1000
    setY:=450
  <\Object>

<\Agents>

//Définition de trois routes qui passent par ces intersections

<Routes>

  <Object>
    lasttrafficsimulation.Route
    <Methods>
      setRoute:=1
      setRoute:=2
      setRoute:=4
      setTitle:=Paris
    <\Object>

  <Object>
    lasttrafficsimulation.Route
    <Methods>
      setRoute:=5
      setRoute:=2
      setRoute:=3
      setTitle:=Nancy
    <\Object>

  <Object>
    lasttrafficsimulation.Route
    <Methods>
      setRoute:=1
      setRoute:=3
      setRoute:=4
      setRoute:=5
      setTitle:=Lyon
    <\Object>

<\Routes>
```


Bibliographie

- AGV : URL http://fr.wikipedia.org/wiki/V%C3%A9hicule_%C3%A0_guidage_automatique.
- Y. AHADAR, R. BOUYEKHF et A. EL MOUDNI : minimisation des files d'attente d'une intersection isolée. *In 10ème Séminaire CONFERE, Belfort – France*, pages 361–368, 2003.
- S. ALBOUZE : Stratégie réactive pour le croisement sans arrêt de deux platoons de véhicules décentralisés. Stage master de recherche, Université Henri-Poincaré, Nancy, France, février 2010.
- L. ALEC MILLER : *Applied Queueing Theory*. MacMillan, 1966.
- G. BALAN et S. LUKE : History-based traffic control. *In Proceedings of the fifth International Conference on Autonomous Agents and Multiagent Systems*, pages 616–621. ACM, 2006.
- M. BAYKAL-GÜRSOY, Z. DUAN et H. XU : Stochastic models of traffic flow interrupted by incidents. *In IFAC Symposium on Transportation Systems*, 2009a.
- M. BAYKAL-GÜRSOY, W. XIAO, Z. DUAN et K. OZBAY : Delay estimation for traffic flow interrupted by incidents. *In 86th Annual Transportation Research Conf., Transportation Research Board, Washington, DC*, 2006.
- M. BAYKAL-GÜRSOY, W. XIAO et K. OZBAY : Modeling traffic flow interrupted by incidents. *European Journal of Operational Research*, 195(1):127 – 138, 2009b.
- A.L.C. BAZZAN : A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multi-Agent Systems*, 10(1):131–164, 2005.
- A.L.C. BAZZAN : Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, 18(3):342–375, 2009.
- M. BEHRISCH, L. BIEKER, J. ERDMANN et D. KRAJZEWICZ : Sumo-simulation of urban mobility-an overview. *In The Third International Conference on Advances in System Simulation*, pages 55–60, 2011.

- N. BHOURI, F. BALBO, S. PINSON et M. TLIG : Collaborative agents for modeling traffic regulation systems. *In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 2, pages 7–13, aug. 2011.
- O. BIHAM, A.A. MIDDLETON et D. LEVINE : Self-organization and a dynamical transition in traffic-flow models. *Physical Review A*, 46(10):R6124, 1992.
- A. BONHOMME, P. MATHIEU et S. PICAULT : A versatile description framework for modeling behaviors in traffic simulations. *In International Conference on Tools with Artificial Intelligence, (ICTAI'14)*, pages 937–944, 2014.
- L. BONTE, S. ESPIÉ et P. MATHIEU : Modélisation et simulation des usagers deux-roues motorisés dans archisim. *JFSMA*, 6:17, 2006.
- L. BONTE, S. ESPIÉ et P. MATHIEU : Virtual lanes interest for motorcycles simulation. *In Proceedings of the fifth European Workshop on Multi-Agent Systems (EUMAS'07)*, pages 580–596. ATIA, 2007.
- E. BROCKFELD, R. BARLOVIC, A. SCHADSCHNEIDER et M. SCHRECKENBERG : Optimizing traffic lights in a cellular automaton model for city traffic. *Physical Review E*, 64(5):056132, 2001.
- L. BULL, J. SHA'ABAN, A. TOMLINSON, J.D. ADDISON et B.G. HEYDECKER : Towards distributed adaptive control for road traffic junction signals using learning classifier systems. *In Applications of Learning Classifier Systems*, pages 276–299. Springer Berlin Heidelberg, 2004.
- E. CAMPONOGARA et W. KRAUS JR : Distributed learning agents in urban traffic control. *In Progress in Artificial Intelligence*, pages 324–335. Springer, 2003.
- E. CHEVALLIER et L. LECLERCQ : Do microscopic merging models reproduce the observed priority sharing ratio in congestion? *Transportation Research Part C : Emerging Technologies*, 17(3):328–336, 2009.
- K.F. CLANCEY et I. GOHBERG : *Factorization of matrix functions and singular integral operators*. Springer, 1981.
- J.W. COHEN : A two-queue, one-server model with priority for the longer queue. *Queueing Systems*, 2(3):261–283, 1987.
- B.C. DA SILVA, E.W. BASSO, A.L.C. BAZZAN et P.M. ENGEL : Dealing with non-stationary environments using context detection. *In Proceedings of the 23rd international conference on Machine learning*, pages 217–224. ACM, 2006.
- D. DE OLIVEIRA et A.L.C. BAZZAN : Traffic lights control with adaptive group formation based on swarm intelligence. *In Ant Colony Optimization and Swarm Intelligence*, pages 520–521. Springer, 2006.

-
- D. DE OLIVEIRA et A.L.C. BAZZAN : Swarm intelligence applied to traffic lights group formation. *Anais do VI Encontro Nacional de Inteligência Artificial (ENIA)*, pages 1003–1112, 2007.
- A. DONIEC, S. ESPIÉ, R. MANDIAU et S. PIECHOWIAK : Dealing with multi-agent coordination by anticipation : Application to the traffic simulation at junctions. *EUMAS*, 5:478–479, 2005.
- K. DRESNER et P. STONE : Multiagent traffic management : A reservation-based intersection control mechanism. *In Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2004.
- K. DRESNER et P. STONE : Multiagent traffic management : An improved intersection control mechanism. *In Proceedings of the Fourth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2005.
- S. ESPIÉ : Archisim, multi-actor parallel architecture for traffic simulation. *In Proceedings of the Second World Congress on Intelligent Transport Systems*, volume 4, 1995.
- J. FERBER : *Les Systèmes multi-agents*. InterEditions, 1995.
- J. FERBER : *Multi-Agent Systems. An introduction to Distributed Artificial Intelligence*. John Wiley and Sons Inc., New York, 1999.
- E.D. FERREIRA et P.K. KHOSLA : Multi agent collaboration using distributed value functions. *In Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 404–409. IEEE, 2000.
- R. E. FIKES et N. J. NILSSON : STRIPS : A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208, 1971.
- C.L. FOK, M. HANNA, S. GEE, T.C. AU, P. STONE, C. JULIEN et S. VISHWANATH : A platform for evaluating autonomous intersection management policies. *In Third International Conference on Cyber-Physical Systems (ICCPS)*, pages 87–96. IEEE, 2012.
- J. FRANCE et A.A. GHORBANI : A multiagent system for optimizing urban traffic. *In Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on*, pages 411–414. IEEE, 2003.
- F. GAILLARD, C. DINONT, M. SOULIGNAC et P. MATHIEU : Towards cognitive steering behaviours for two-wheeled robots. *In ICAART (2)*, pages 119–125, 2012.
- F. GAILLARD, M. SOULIGNAC, C. DINONT et P. MATHIEU : Deterministic kinodynamic planning with hardware demonstrations. *In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3519–3525. IEEE, 2011.
- F. GECHTER, J.M. CONTET, O. LAMOTTE, S. GALLAND et A. KOUKAM : Virtual intelligent vehicle urban simulator : Application to vehicle platoon evaluation. *Simulation Modelling Practice and Theory (SIMPAT)*, 24:103–114, mai 2012.

- C. GERSHENSON : Self-organizing traffic lights. *complex Systems*, 16(1):29–53, 2004.
- A. GLAD : *Etude de l'auto-organisation dans les algorithmes de patrouille multi-agent fondés sur les phéromones digitales*. Thèse de doctorat, Université Henri Poincaré, Nancy 1, novembre 2011. Laboratoire Lorrain de recherche en informatique et ses applications (LORIA).
- A. GLAD, O. SIMONIN, O. BUFFET et F. CHARPILLET : Influence of different execution models on patrolling ant behaviors : from agents to robots. *In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1173–1180, 2010.
- D.M. GORDON : The organization of work in social insect colonies. *Nature*, 380(6570):121–124, 1996.
- B.D. GREENSHIELDS : A study of traffic capacity. *In Proceedings of the Highway Research Board*, volume 14, pages 448–477, 1935.
- A.A. GUEBERT et G. SPARKS : Timing plan sensitivity to changes in platoon dispersion settings. *In Traffic Control Methods. Proceedings of the fifth ng foundation conference, Santa barbara, California*, 1990.
- P.A. HANCOCK : Is car following the real question—are equations the answer? *Transportation research part F : traffic psychology and behaviour*, 2(4):197–199, 1999.
- D. HELBING, S. LÄMMER et J.P. LEBACQUE : Self-organized control of irregular or perturbed network traffic. *In Optimal control and dynamic games*, pages 239–274. Springer, 2005.
- J.J. HENRY, J.L. FARGES et J. TUFFAL : The prodyn real time traffic algorithm. *In 4th IFAC/IFIP/IFORS Conference On Control*, 1984.
- P. HIDAS : Modelling lane changing and merging in microscopic traffic simulation. *Transportation Research Part C : Emerging Technologies*, 10(5-6):351–371, 2002.
- N. HOUNSELL et B. SHRESTHA : A new approach for co-operative bus priority at traffic signals. *Intelligent Transportation Systems*, 13(1):6–14, 2012.
- P. B. HUNT, D. I. ROBERTSON, R. D. BRETHERTON et R. I. WINTON : SCOOT - a traffic responsive method of coordinating signals. Rapport technique, TRRL, 1981.
- D.G. KENDALL : Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics*, 24(3):pp. 338–354, 1953.
- S. KOENIG, B. SZYMANSKI et Y. LIU : Efficient and inefficient ant coverage methods. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):41–76, 2001.

-
- E. KÖHLER, R.H. MÖHRING et G. WÜNSCH : Minimizing total delay in fixed-time controlled traffic networks. *In Operations Research Proceedings 2004*, pages 192–199. Springer Berlin Heidelberg, 2005.
- I. KOSONEN : Multi-agent fuzzy signal control based on real-time simulation. *Transportation Research Part C : Emerging Technologies*, 11(5):389–403, 2003.
- I. KOSONEN et M. PURSULA : A simulation tool for traffic signal control planning. *In Third International Conference on Road Traffic Control.*, pages 72 –76. IEEE, may 1990.
- F. KSONTINI : *Modèle d'agent fondé sur les affordances : application à la simulation de trafic routier*. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambresis, 2013.
- B. LACROIX et P. MATHIEU : Automated generation of various and consistent populations in multi-agent simulations. *In Advances on Practical Applications of Agents and Multi-Agent Systems*, pages 127–137. Springer, 2012.
- B. LACROIX, P. MATHIEU et A. KEMENY : A normative model for behavioral differentiation. *In Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, volume 2, pages 96–99. IEEE, 2008.
- B. LACROIX, P. MATHIEU et A. KEMENY : Formalizing the construction of populations in multi-agent simulations. *Engineering Applications of Artificial Intelligence*, 26(1):211–226, 2013.
- M. J. LIGHTHILL et G. B. WHITHAM : On kinematic waves. 2. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 229(1178):317–345, 1955.
- L.J. LIN : Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.
- P.R. LOWRIE : The sydney coordinated adaptive traffic system-principles, methodology, algorithms. *In International Conference on Road Traffic Signalling, 1982, London, United Kingdom*, 1982.
- W. G. MACREADY, A. G. SIAPAS et S. A. KAUFFMAN : Criticality and parallelism in combinatorial optimization. *Science*, 271(5245):56–59, 1996.
- S. MAMMAR : *Systèmes de Transport Intelligents, modélisation, information et contrôle*. Hermes science, Lavoisier, 2007.
- R. MANDIAU, A. CHAMPION, J.M. AUBERLET, S. ESPIÉ et C. KOLSKI : Behaviour based on decision matrices for a coordination between agents in a urban traffic simulation. *Applied Intelligence*, 28(2):121–138, 2008.

- D. MEIGNAN, O. SIMONIN et A. KOUKAM : Simulation and evaluation of urban bus-networks using a multiagent approach. *International Journal Simulation Modelling Practice and Theory (Elsevier)*, 15(6):659–671, 2007.
- R.H. MOHRING, K. NOKEL et G. WUNSCH : A model and fast optimization method for signal coordination in a network. *In Control in Transportation Systems*, volume 11, pages 73–78, 2006.
- P. MORRIS : The breakout method for escaping from local minima. *In AAAI*, volume 93, pages 40–45, 1993.
- K. NAGEL et M. SCHRECKENBERG : A cellular automaton model for freeway traffic. *Journal de physique I*, 2(12):2221–2229, 1992.
- R. NAUMANN et R. RASCHE : *Intersection collision avoidance by means of decentralized security and communication management of autonomous vehicles*. Univ.-GH, SFB 376, 1997.
- R. NAUMANN, R. RASCHE, J. TACKEN et C. TAHEDI : Validation and simulation of a decentralized intersection collision avoidance algorithm. *In Intelligent Transportation System (ITSC'97), IEEE*, pages 818–823, 1997.
- B. NEBEL et J. KOEHLER : Plan reuse versus plan generation : A theoretical and empirical analysis. *Artificial Intelligence*, 76:427–454, 1995.
- L. NUNES et E. OLIVEIRA : Learning from multiple sources. *In Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1106–1113. IEEE Computer Society, 2004.
- C. REYNOLDS : Flocks, herds and schools : A distributed behavioral model. *Computer Graphics*, 21(4):25–24, 1987.
- P.I. RICHARDS : Shock waves on the highway. *Operations research*, 4(1):42–51, 1956.
- S. RICHTER, D. ABERDEEN et J. YU : Natural actor-critic for road traffic optimization. In B. SCHÖLKOPF, J. PLATT et T. HOFMANN, éditeurs : *Advances in Neural Information Processing Systems 19*, Cambridge, MA, 2007. MIT Press.
- D.I. ROBERTSON : *TRANSYT : A Traffic Network Study Tool*. RRL report. Road Research Laboratory, 1969.
- D.I. ROBERTSON et R.D. BRETHERTON : Optimizing networks of traffic signals in real time-the scoot method. *Vehicular Technology, IEEE Transactions on*, 40(1):11–15, février 1991.
- G.E. ROBINSON : Regulation of division of labor in insect societies. *Annual review of entomology*, 37(1):637–665, 1992.

-
- F. ROCHNER, H. PROTHMANN, J. BRANKE, C. MÜLLER-SCHLOER et H. SCHMECK : An organic architecture for traffic light controllers. *In Proceedings of the Informatik 2006 –Informatik fürMenschen–I Jahrestagung (1)*, pages 120–127, 2006.
- D.E. RUMELHART, G.E. HINTON et R.J. WILLIAMS : Learning internal representations by error propagation. Rapport technique, DTIC Document, 1985.
- S. RUSSELL et P. NORVIG : *Artificial Intelligence : A Modern Approach*. Englewood Cliffs, NJ : prentice Hall, 1995.
- J.H.D. SÁNCHEZ et J.L. AGUIRRE : Traffic light control through agent-based coordination. *In Artificial Intelligence and Applications*, pages 163–168, 2007.
- A. SCHEUER, O. SIMONIN et F. CHARPILLET : Safe longitudinal platoons of vehicles without communication. *In Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 2835–2840, 2009.
- J. SCHNEIDER, W.K. WONG, A. MOORE et M. RIEDMILLER : Distributed value functions. *In Proceedings of the Sixteenth International Conference on Machine Learning*, pages 371–378, 1999.
- J.B. SHEU : Microscopic traffic behaviour modelling and simulation for lane-blocking arterial incidents. *Transportmetrica A : Transport Science*, 9(4):335–357, 2013.
- O. SIGAUD et O. BUFFET : *Markov decision processes in artificial intelligence*. John Wiley & Sons, 2010.
- P.M. SIMON et K. NAGEL : Simplified cellular automaton model for city traffic. *Phys. Rev. E*, 58(2):1286–1295, Aug 1998.
- O. SIMONIN : *Le modèle satisfaction-altruisme : coopération et résolution de conflits entre agents situés réactifs, application à la robotique*. Thèse de doctorat, Université de Montpellier 2/LIRMM, 2001.
- O. SIMONIN : *Contribution à la résolution collective de problème - Modèles d'auto-organisation par interactions directes et indirectes dans les SMA réactifs et robotiques*. Habilitation à diriger des recherches en informatique, Université Henri Poincaré (Nancy 1) - LORIA, Décembre 2010.
- O. SIMONIN et J. FERBER : Modeling self satisfaction and altruism to handle action selection and reactive cooperation. *In The Sixth International Conference on the Simulation of Adaptive Behavior (FROM ANIMALS TO ANIMATS 6)*, pages 314–323, Paris, France, 2000. in proceedings Supplement SAB 2000.
- M. STEINGROVER, R. SCHOUTEN, S. PEELLEN, E. NIJHUIS et B. BAKKER : Reinforcement learning of traffic light controllers adapting to traffic congestion. *In Proceedings of the Belgium-Netherlands Artificial Intelligence Conference (BNAIC)*, pages 216–223. Citeseer, 2005.

- J.C. TANNER : A problem of interference between two queues. *Biometrika*, 40(1/2):pp. 58–69, 1953.
- J.C. TANNER : A theoretical analysis of delays at an uncontrolled intersection. *Biometrika*, 49(1-2):163–170, 1962.
- M. Tlig, O. BUFFET et O. SIMONIN : Cooperative behaviors for the self-regulation of autonomous vehicles in space sharing conflicts. *In Proceedings of the twenty-fourth International Conference on Tools with Artificial Intelligence (ICTAI'12)*, 2012.
- M. Tlig, O. BUFFET et O. SIMONIN : Croisement synchronisé de flux de véhicules autonomes dans un réseau. *In actes des 11èmes Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA'13)*, 2013a.
- M. Tlig, O. BUFFET et O. SIMONIN : Reactive coordination rules for traffic optimization in road sharing problems. *In Proceedings of PAAMS'2013, AATMO'2013 Workshop Agent-based Approaches for the Transportation Modelling and Optimisation (PAAMS'13)*. CCIS Series of Springer, 2013b.
- M. Tlig, O. BUFFET et O. SIMONIN : Decentralized traffic management : A synchronization-based intersection control. *In Proc. of the International Conference on Advanced Logistics and Transport (ICALT'14)*, 2014a. <http://hal.inria.fr/hal-00960735>.
- M. Tlig, O. BUFFET et O. SIMONIN : Stop-free strategies for traffic networks : Decentralized on-line optimization. *In Proceedings of the European Conference on Artificial Intelligence (ECAI'14)*, 2014b.
- M. TREIBER et A. KESTING : An open-source microscopic traffic simulator. *Intelligent Transportation Systems Magazine, IEEE*, 2(3):6–13, 2010.
- J.P. TREUIL, A. DROGOUL et J.D. ZUCKER : *Modélisation et simulation à base d'agents : exemples commentés, outils informatiques et questions théoriques*. Dunod, 2008.
- C.J.C.H. WATKINS et P. DAYAN : Q-learning. *Machine learning*, 8(3-4):279–292, 1992.
- Z. WEIXIONG, W. GUANDONG, X. ZHAO et W. LARS : Distributed stochastic search and distributed breakout : properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1-2):55–87, 2005.
- M. WIERING : Multi-agent reinforcement learning for traffic light control. *In Proceedings of the International Conference on Machine Learning*, pages 1151–1158, 2000.
- K. ZEGHAL : A comparison of different approaches based on force fields for coordination among multiple mobiles. *In Intelligent Robots and Systems, IEEE/RSJ International Conference on*, volume 1, pages 273 –278 vol.1, oct 1998.

Résumé

Dans le cadre de cette thèse, nous nous intéressons à la coordination et l'optimisation du trafic aux intersections des réseaux routiers, avec la particularité de considérer des véhicules autonomes intelligents. Cette thèse est organisée en deux grandes parties. La première se concentre sur le problème du partage d'un espace de voie par deux files de véhicules évoluant en sens opposés. L'état de l'art montre le peu de travaux abordant cette question. Nous explorons deux approches par coordination réactive, en relation avec un critère de minimisation des retards. Les performances de ces approches ont été mesurées statistiquement en simulation. La deuxième partie de la thèse s'attaque au problème générique de la gestion du trafic au sein d'un réseau routier. Nous développons une approche originale à deux égards : d'une part elle explore un principe de passage en alternance des flux permettant de ne pas arrêter les véhicules aux intersections, et d'autre part, elle propose des algorithmes d'optimisation distribuée de ce passage alterné au niveau de chaque intersection et au niveau du réseau global. La thèse présente successivement les choix de modélisation, les algorithmes et l'étude en simulation de leurs performances comparées à des approches existantes.

Mots-clés: Systèmes multi-agents, Véhicules autonomes, Gestion du trafic, Intelligence artificielle.

Abstract

In this thesis, we focus on traffic coordination and optimization in road intersections, while accounting for intelligent autonomous vehicles. This thesis is organized in two parts. The first part focuses on the problem of sharing a one-lane road between two opposite flows of vehicles. The state of the art shows few studies addressing this issue. We propose two reactive coordination approaches that minimize vehicle delays and measure their performances statistically through simulations. The second part of the thesis addresses the problem of generic traffic management in a traffic network. We develop a stop-free approach that explores a principle alternating vehicles between flows at intersections, and it provides distributed algorithms optimizing this alternation at each intersection and in the overall network. We present the modeling choices, the algorithms and the simulation study of our approach and we compare its performances with existing approaches.

Keywords: Multi-agent systems, Autonomous vehicle, Traffic management, Artificial intelligence.