

THÈSE

de l'École Doctorale
Sciences Pour l'Ingénieur et Microtechniques

préparée à
l'Université de Technologie de Belfort-Montbéliard

pour obtenir le grade de
Docteur en Automatique

délivré conjointement par
l'Université de Technologie de Belfort-Montbéliard
et par
l'Université de Besançon

Présentée par

Aurélien CORRÉIA

MODÉLISATION DE CONFLITS DANS L'ALGÈBRE DES DIOÏDES APPLICATION À LA RÉGULATION DE TRAFIC DANS LES CARREFOURS

soutenue le mardi 18 décembre 2007

Président	Mohamed BENREJEB	Professeur	ENIT, Tunis, Tunisie
Rapporteurs	Jean-Louis BOIMOND Slim HAMMADI	Professeur Professeur	ISTIA, Angers, France École Centrale de Lille, France
Examineurs	Jean-Pierre QUADRAT Mohamed BENREJEB Rachid BOUYEKHF	Directeur de Recherche Professeur Maître de Conférences	INRIA, Rocquencourt, France ENIT, Tunis, Tunisie UTBM, Belfort, France
Directeur	Abdellah EL MOUDNI	Professeur	UTBM, Belfort, France
Co-directeur	Abdel-Jalil ABBAS-TURKI	Maître de Conférences	UTBM, Belfort, France

Laboratoire Systèmes et Transport
SeT, UTBM, site de Belfort
90010 Belfort Cedex

Aurélien CORRÉIA (aurelien.correia@utbm.fr)

Modélisation de conflits dans l'algèbre des dioïdes: application à la régulation de trafic dans les carrefours
172 pages.

Les rapports de recherche du Laboratoire Systèmes et Transport sont disponibles à l'adresse:

Research reports from Laboratoire Systèmes et Transport are available at:

<http://set.utbm.fr>

rapport_these.tex – Modélisation de conflits dans l'algèbre des dioïdes: application à la régulation de trafic dans les carrefours

Compilé le mardi 25 mars 2008 à 11h27 avec L^AT_EX₂_ε et la classe `kixam_thesis v. 1.0.0.1`, disponible à l'adresse

<http://aurelien.correia.free.fr>

Remerciements

Je remercie le Professeur Abdellah El Moudni pour m'avoir accordé toute sa confiance en me laissant une grande liberté dans ma recherche, tout en prodiguant des conseils avisés et opportuns. Je remercie infiniment le Docteur Abdel-Jalil Abbas-Turki pour sa disponibilité sans limite aucune et pour son aide des plus précieuses. Je remercie également les membres du jury les Professeurs Jean-Louis Boimond, Slim Hammadi, Jean-Pierre Quadrat et Mohamed Ben-rejeb, ainsi que le Docteur Rachid Bouyekhf, pour leurs remarques pertinentes qui ont permis d'améliorer sensiblement la qualité de cette thèse.

Je souhaite également remercier Jia Wu avec qui j'ai directement collaboré. J'adresse une pensée particulière à Lyamine Bouhafs, Salah-Eddine Merzouk, Marinette Morlot, Ariane Glatigny et tous les membres du laboratoire Systèmes et Transports qui m'ont aidé moralement et physiquement tout au long de ces trois années.

Je ne me serais pas engagé dans ce travail de longue haleine sans leurs connaissances et leurs encouragements: je remercie Sébastien Lahaye, Bertrand Cottenceau, Jean-Louis Ferrier et toute l'équipe du Laboratoire d'Ingénierie des Systèmes Automatisés d'Angers de m'avoir insufflé leur passion de la recherche.

Quels mots du vocabulaire académique si froid pourraient me permettre d'exprimer justement mes sentiments: je vous dirai oralement ma profonde gratitude que je ne peux écrire ici, à toi ma femme Mélanie et à vous mes trois enfants Émilie, Camille et Benjamin.

Vous pensez que vous n'avez pas contribué un tant soit peu à ce travail, pourtant je vous remercie d'avoir simplement permis et facilité son aboutissement: merci à Véronique et Didier Corrêa, mes parents, sans qui je n'existerais pas. Merci à Agnès Abbas-Turki pour sa patience et sa tolérance, sans limite aucune. Merci à vous que je n'ai pas cités ici et qui, pourtant, le méritez.

Table des matières

Liste des notations	7
Liste des acronymes	9
Introduction générale	11
1 Présentation du système	13
1.1 Généralités sur le trafic routier	15
1.1.1 Composantes du trafic	16
1.1.2 Instruments de mesure du trafic	16
1.1.3 Modélisation du trafic	17
1.1.4 Simulation du trafic	24
1.2 Régulation des carrefours urbains à feux	25
1.2.1 Carrefour urbain à feux	26
1.2.2 Plans de feux	27
1.2.3 Régulation adaptative	29
1.3 Approches émergentes de régulation	33
1.3.1 Approche basée sur la distribution en différé	33
1.3.2 Approche basée sur le PCLE	35
1.3.3 Approches basées sur les Réseaux de Petri	38
2 Étude d'un tronçon routier	43
2.1 Outils algébriques et applications	45
2.1.1 Dioïdes	46
2.1.2 Mise en équation des GETs	49
2.1.3 Bornes des temps de séjour et des marquages	53
2.2 Étude d'une route	58
2.2.1 Modélisation	58
2.2.2 Évaluation des performances	61
2.2.3 Validation	63
2.3 Régulation du trafic	67

3	Régulation de carrefour : un problème de partage de ressources	73
3.1	Gestion des conflits dans les dioïdes	75
3.1.1	Problème de ressources partagées et état de l'art	76
3.1.2	Modèle analytique	79
3.1.3	Écriture du modèle dans l'algèbre des dioïdes	82
3.1.4	Exemple : modélisation d'une gare de fret	82
3.2	Modélisation et commande d'un carrefour simplifié	91
3.2.1	Modèle d'un carrefour simplifié	91
3.2.2	Commande par fenêtres de temps	96
3.2.3	Commande optimale	98
3.3	Étude comparative	103
4	Carrefour contemporain	111
4.1	Présentation du problème et modélisation	113
4.1.1	Problème de ressources partagées avec temps de configuration	113
4.1.2	Modélisation analytique	115
4.1.3	Écriture du modèle dans l'algèbre des dioïdes	119
4.1.4	Application au trafic urbain	120
4.2	Commande	123
4.2.1	Commande en juste-à-temps	124
4.2.2	Commande optimale par programmation dynamique	126
4.3	Validation	130
4.3.1	Simulation d'un carrefour	130
4.3.2	Simulation de deux carrefours en cascade	134
	Conclusion générale	143
	Liste des tableaux	144
	Liste des figures	145
	Index	149
	Bibliographie	155
	Annexes	165

Liste des notations

\mathbb{N}	ensemble des entiers naturels
$\mathbb{N}^* = \mathbb{N} - \{0\}$	ensemble des entiers naturels sans 0
\mathbb{Z}	ensemble des entiers relatifs
$\mathbb{Z}^* = \mathbb{Z} - \{0\}$	ensemble des entiers relatifs sans 0
\mathbb{R}	ensemble des nombres réels
\mathbb{R}_+	ensemble des nombres réels positifs
X^T	transposée du vecteur X
$[X]_i$	i^{e} élément du vecteur X
$[X]_{ij}$	élément de la i^{e} ligne et de la j^{e} colonne de la matrice X
$\max_{x \in E} f(x)$	maximum de $f(\cdot)$ dans E
$\min_{x \in E} f(x)$	minimum de $f(\cdot)$ dans E
ε	élément neutre vis-à-vis de l'addition
e	élément neutre vis-à-vis de la multiplication
a^*	étoile de Kleene : $a^* = \bigoplus_{i \in \mathbb{N}} a^i$
$\overline{\mathbb{Z}}$	ensemble $\mathbb{Z} \cup \{-\infty, +\infty\}$
$\overline{\mathbb{Z}}_{\max}$	dioïde $(\overline{\mathbb{Z}}, \max, +)$ également appelé algèbre $(\max, +)$
$\overline{\mathbb{Z}}_{\min}$	dioïde $(\overline{\mathbb{Z}}, \min, +)$ également appelé algèbre $(\min, +)$
$\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$	dioïde des séries formelles à deux coefficients (γ et δ) et à exposants dans \mathbb{Z}
$\bigvee_{x \in E} f(x)$	plus petit majorant de $f(\cdot)$ dans E
$\bigwedge_{x \in E} f(x)$	plus grand minorant de $f(\cdot)$ dans E
$a \oslash b = \frac{b}{a}$	notation utilisée pour représenter $L_a^\sharp(b)$
$b \oslash a = \frac{b}{\overleftarrow{a}}$	notation utilisée pour représenter $R_a^\sharp(b)$

Liste des acronymes

AC Automate Cellulaire

AFNOR Association Française de NORmalisation

CRONOS ContROI of Networks by Optimization Switchovers

ELCP Extended Linear Complementary Problem

GCA Graphe à Choix Asymétriques — *Classe de RdP*

GCL Graphe à Choix Libres — *Classe de GCLE*

GCLE Graphe à Choix Libres Étendu — *Classe de GCA*

GERTRUDE Gestion Électronique de Régulation du Trafic Routier Défiant les Embouteillages

GET Graphe d'Événements Temporisés — *GM discret et temporisé*

GM Graphe Marqué — *Classe de GCL*

GPS Global Positioning System — *Système de positionnement mondial par satellite*

INRETS Institut National de REcherche sur les Transports et leur Sécurité

OPAC Optimization Policies for Adaptative Control

PCLE Problème de Complémentarité Linéaire Étendu — *en anglais : « Extended Linear Complementary Problem (ELCP) »*

PL Poids Lourd — *Véhicule de plus de 3500kg*

PRODYN PROgrammation DYNamique

PRPI Problème de Ressources Partagées Invariantes

PRPITC Problème de Ressources Partagées Invariantes avec Temps de Configuration

RdP Réseau de Petri — *Modèle graphique*

SCATS Sydney Coordinated Adaptative Traffic System

SCOOT Split Cycle and Offset Optimization Technique

SED Système dynamique à Événements Discrets — *Système dont l'évolution (discrète) dépend d'événements et non du temps*

SRPI Système à Ressources Partagées Invariantes

SRPITC Système à Ressources Partagées Invariantes avec Temps de Configuration

TIV Temps Inter-Véhiculaire

TUC Traffic-responsive Urban Control — *Stratégie de commande par supervision*

u.t. unité de temps

UTOPIA Urban Traffic OPTimization by Integrated Automation

u.v.p. unité de voiture particulière

VL Véhicule Léger — *Véhicule motorisé de 3500kg ou moins (plus de deux roues)*

Introduction générale

Les effets négatifs induits par l'usage excessif des véhicules individuels sont bien connus. Les principaux sont la pollution, l'insécurité et la congestion. Cependant, les tendances actuelles du développement économique laissent à penser que cet usage ne va pas baisser, tout au contraire. Malgré les tentatives d'aménagement des villes et de leurs infrastructures et l'émergence des technologies de prévision de la demande, nous restons face au dilemme opposant l'amélioration de la qualité de vie et la réduction des problèmes liés au trafic.

Les carrefours sont le point névralgique de la gestion du trafic urbain. En effet, c'est au niveau du carrefour que se décide une bonne partie du partage de l'espace alloué aux véhicules. Cependant, bien que plusieurs recherches dans le domaine soient réalisées, les efforts menés jusqu'à présent laissent encore le problème de la régulation des carrefours complètement ouvert. Ceci peut être dû au fait que les principes, voire même les moyens de régulation ne suivent pas une évolution aussi rapide que le trafic. Rappelons que, depuis plus de vingt ans, la majorité des feux de circulation sont pilotés à partir d'estimations du débit et, dans le meilleur des cas, de la file d'attente instantanée.

Ce constat pousse aujourd'hui plusieurs chercheurs à changer de vision sur le trafic. Ils proposent de nouvelles approches pour enrichir les techniques éprouvées et ainsi améliorer leurs résultats. Contrairement au modèle dominant qui considère le trafic comme un flot, il apparaît intéressant de traiter les véhicules d'une manière individuelle. Dans ce mémoire nous contribuons à l'exploration de cette piste.

Le fait de considérer les véhicules individuellement implique le recours à des modèles discrets. La présence de nombreuses synchronisations dans le système et la nécessité d'évaluer avec précision chaque arrivée de véhicule nous pousse naturellement à utiliser l'algèbre des dioïdes. Cette algèbre nous permet de dater et de compter les événements à travers des équations linéaires. Bien que l'algèbre des dioïdes soit idéale pour appréhender les phénomènes de synchronisation et de parallélisme, elle ne permet pas de représenter les conflits inhérents au fonctionnement du carrefour.

Le problème de conflit dans les dioïdes est au cœur de plusieurs recherches. Depuis ces dix dernières années, nous rencontrons dans la littérature des travaux qui tentent de s'affranchir de cette difficulté, soit en la contournant, soit en proposant une approche de modélisation

qui décompose le système en plusieurs modes de fonctionnement. D'une part, le contournement est réalisé en supposant une commande déjà établie, ce qui ne nous permet pas d'aborder l'ensemble des possibilités de régulation compatibles avec le système. D'autre part, bien que le découpage en modes de fonctionnement nous renseigne sur les configurations possibles du système, il déplace la difficulté de résolution du conflit vers l'étape de la modélisation. En effet, il faut autant de modèles que de solutions possibles.

Ainsi, dans ce mémoire, nous abordons les deux interrogations suivantes :

- Est-il possible d'obtenir un modèle de trafic discret, représentant fidèlement les comportements des véhicules pris individuellement, dans l'algèbre des dioïdes ?
- Comment palier le problème de la modélisation des conflits, inhérents aux carrefours, dans l'algèbre des dioïdes ?

Dans une première partie, nous rappelons le vocabulaire de l'ingénierie du trafic. Ceci nous permet d'introduire les principes fondamentaux qui régissent les systèmes de régulation actuellement en place dans nos villes. Plus précisément, nous mettons en évidence les avantages et les inconvénients des différents outils mis à la disposition de l'ingénieur aujourd'hui. Nous présentons notamment plusieurs approches récentes de la littérature.

Après avoir présenté brièvement l'algèbre des dioïdes, nous développons, dans le deuxième chapitre, un nouveau modèle de trafic basé sur cette algèbre. Puisqu'elle est souvent associée aux Réseaux de Petri, nous introduisons également cet outil de modélisation graphique. Ainsi, nous pouvons illustrer notre modèle mathématique par une représentation plus lisible. L'évaluation des performances du trafic se fait à partir des équations décrivant le comportement du système.

Le troisième chapitre tente de modéliser une partie des conflits par l'algèbre des dioïdes. Cette contribution théorique est nécessaire au développement d'un nouveau modèle de carrefour simplifié. Plus précisément, nous supposons que les accélérations des véhicules sont négligeables. Dans ce contexte, nous proposons une approche pour décrire les comportements du système de trafic en dehors de toute solution particulière.

Enfin, dans le dernier chapitre, nous nous attaquons à la modélisation et la régulation de carrefours contemporains plus réalistes. Autrement dit, nous prenons en compte le fait que les véhicules, après s'être arrêtés à un feu, mettent un temps non négligeable pour atteindre leur vitesse moyenne. Nous considérons également, à chaque changement de feu, une période intermédiaire contraignante, pendant laquelle tous les feux sont rouges.

Chapitre 1

Présentation du système

1.1	Généralités sur le trafic routier	15
1.1.1	Composantes du trafic	16
1.1.2	Instruments de mesure du trafic	16
1.1.3	Modélisation du trafic	17
1.1.4	Simulation du trafic	24
1.2	Régulation des carrefours urbains à feux	25
1.2.1	Carrefour urbain à feux	26
1.2.2	Plans de feux	27
1.2.3	Régulation adaptative	29
1.3	Approches émergentes de régulation	33
1.3.1	Approche basée sur la distribution en différé	33
1.3.2	Approche basée sur le PCLE	35
1.3.3	Approches basées sur les Réseaux de Petri	38

Introduction

CHACQUE année, nous pouvons constater une augmentation du trafic routier, due à l'utilisation excessive de véhicules particuliers et la généralisation du fret par camion. Les villes se trouvent ainsi gravement atteintes par les problèmes de pollutions atmosphérique et sonore, de sécurité et de congestion inhérents à ces modes de transport. Depuis quelques dizaines d'années, la recherche fournit de nouveaux moyens pour réguler le trafic, afin de le fluidifier et de mieux gérer les incidents. Des outils permettent aujourd'hui de maîtriser les flux de véhicules sur plusieurs échelles, depuis un réseau autoroutier complexe jusqu'au moindre carrefour à feux. Les feux de carrefour représentent d'ailleurs un dispositif central dans l'effort de régulation, puisqu'ils permettent d'agir directement sur les flux en conflit à chaque intersection, en distribuant successivement la priorité à chacun d'entre eux de manière à réduire les temps de parcours et à maîtriser le trafic.

Cependant, les modes de commande des feux d'une intersection isolée reposent parfois sur des modèles pensés d'une manière qui les rend inadéquats au traitement des différentes situations rencontrées au niveau local. Il en découle que la commande des feux est souvent approximative et prend en compte des valeurs moyennes ou *a priori* des mesures du trafic. Autrement dit, la commande calculée est idéale pour une situation passée ou supposée, mais ne l'est pas pour la situation réelle et présente. Ainsi, il reste encore beaucoup à faire pour améliorer la régulation par les feux et la synchronisation des intersections à l'échelle locale. Pour cela, il est nécessaire de développer de nouveaux modèles, spécifiques au problème local qu'est la planification des feux. Des efforts doivent être fournis pour obtenir une commande plus précise, c'est-à-dire mieux adaptée aux situations réellement présentes.

Dans ce chapitre, nous présentons les éléments qui composent le trafic urbain, ainsi que son environnement. Nous rappelons également les deux visions du trafic présentes dans la littérature et les objectifs auxquels répondent chacune d'entre elles. Ensuite, nous nous attardons sur les méthodes éprouvées de régulation du trafic. Enfin, nous introduisons quelques méthodes émergentes pour la gestion des carrefours.

1.1 Généralités sur le trafic routier

Avec la démocratisation des véhicules particuliers, les problèmes de sécurité, de nuisance et d'efficacité d'usage des infrastructures n'ont fait qu'accroître le besoin de rationaliser les déplacements. Ainsi, de nombreux scientifiques ont conçu des outils méthodologiques pour évaluer, prévoir et réguler le trafic. Ceci se traduit aujourd'hui par des techniques appliquées et éprouvées. Quelle que soit la méthode d'analyse du trafic, l'objectif demeure l'explication des phénomènes rencontrés quotidiennement dans les réseaux de transport. L'accroissement ininterrompu du trafic et parallèlement des technologies le concernant, rend nécessaire de l'étudier d'une manière mathématique précise, à travers des modèles capables de décrire sa dynamique dans l'espace et dans le temps. Du fait de la diversité de ces modèles, ceux-ci sont classés en deux catégories : microscopique et macroscopique [[Transportation Research Board, 2000](#)].

Dans cette section, nous présentons les composantes du trafic, les instruments de mesure

puis les modèles de trafic. Diverses applications sont envisageables, notamment la simulation et la régulation du trafic. Comme nous consacrons la prochaine section à la régulation du trafic, nous n'aborderons dans cette section que la simulation.

1.1.1 Composantes du trafic

Le *trafic* est constitué de deux composantes majeures : l'infrastructure et les mobiles.

L'*infrastructure* est composée, entre autres, de trottoirs, de bâtiments et de mobiliers urbains (bancs publics, arrêts de bus, *etc.*) qui sont généralement négligés par la modélisation du trafic. En revanche, la chaussée et les signalisations, qui en font également partie, sont les principaux éléments considérés.

Un *mobile* est défini comme étant un *usager* qui conduit son *véhicule* au sein de l'infrastructure. Ils sont au cœur de notre considération. Par contre, les *piétons* (vaquant à pied) ne sont pas considérés car leurs mouvements sont supposés dépendre de celui des mobiles.

Les véhicules se déclinent en quatre *modes*, à savoir les deux-roues, les camions ou *Poids Lourds* (PLs), les transports en commun et les voitures ou *Véhicules Légers* (VLs). Dans la suite de ce mémoire, nous considérons ces modes comme un tout, par l'entremise de l'*unité de voiture particulière* (u.v.p.) [Cohen, 1993] :

- un VL ou une camionnette vaut 1 u.v.p. ;
- un camion ou un bus vaut 2 u.v.p.s ;
- un cycle vaut 0.2 u.v.p.

Notons que les véhicules à l'arrêt ou en stationnement, au sens défini par le code de la route, sont considérés comme faisant partie de l'infrastructure et ne sont généralement pas pris en compte. Les obstacles (*e.g.* travaux, accidents, ...) sont habituellement assimilés à un rétrécissement de la chaussée. En d'autres termes, seuls les mobiles en mouvement sur la chaussée et en interaction avec la signalisation sont considérés. Un *mouvement* est défini par son origine et sa destination. Une voie n'est pas allouée à chaque type de mouvement, mais ceux-ci sont au contraire regroupés en courants sur une même voie.

1.1.2 Instruments de mesure du trafic

Une évolution rapide de la technologie résulte en des capteurs permettant la mesure des variables du trafic. Nous présentons ici les types de capteurs les plus couramment utilisés [Kamata et Oda, 1991].

1.1.2.A Capteurs pneumatiques

Ce sont les capteurs les plus utilisés pour le comptage des véhicules. Un capteur pneumatique est constitué par un câble en caoutchouc relié à son extrémité à un compteur. Le câble tendu traverse la chaussée. Le passage d'un véhicule provoque une surpression : un manomètre actionne alors un relais qui génère une impulsion électrique. Le comptage des impulsions est exprimé en u.v.p..

1.1.2.B Capteurs à ultrasons et à effet Doppler

Ce capteur est constitué d'une antenne fixée sur un portique, lequel est positionné dans l'axe de la voie de circulation. Cette antenne émet constamment des ondes ultra-sonores, qui se propagent à une vitesse constante et connue. Lors du passage d'un véhicule, l'onde émise rencontre un obstacle et est réfléchi. À partir de la différence entre les fréquences de l'onde émise et de celle reçue, la vitesse du véhicule peut être calculée avec une précision d'environ 90%.

1.1.2.C Boucles électromagnétiques

Ces capteurs sont utilisés, en général, dans la commande des feux de circulation en temps réel. Le capteur, noyé dans le revêtement de la chaussée, est constitué d'une boucle inductive. Lors du passage d'un véhicule, une variation du champ électromagnétique survient. Cette variation génère, aux bornes de la boucle inductive, une tension de forme créneau dont la période est liée à la longueur du véhicule et à son temps de passage.

Bien évidemment, cette liste de capteurs n'est pas exhaustive. Notons que l'utilisation de technologies alternatives, telles que les caméras, les ondes radio et le *Global Positioning System* (GPS), font émerger de nouvelles pistes de mesure du trafic.

1.1.3 Modélisation du trafic

Comme nous l'avons mentionné précédemment, nous pouvons classer les modèles de référence en deux grandes parties, à savoir, les modèles microscopiques et macroscopiques. La distinction des deux familles de modèles s'effectue à travers le niveau de détail pris en considération.

1.1.3.A Niveau microscopique

Les grandeurs *microscopiques* mesurent les moindres détails du comportement de chaque véhicule pris individuellement, ainsi que son environnement direct (interactions avec le véhicule précédent). Cette partie rappelle les paramètres les plus couramment rencontrés dans ce niveau d'étude [Cohen, 1993].

L'écart de temps inter-véhiculaire – *L'écart de temps inter-véhiculaire* mesure le temps écoulé entre les passages de deux véhicules successifs. Cette variable est surtout utilisée dans le cadre d'études sur la sécurité routière ou de simulation de trafic. Il existe également des algorithmes en temps réel de commande de feux qui s'appuient sur cette grandeur.

La présence de poids lourds, de caravanes ou d'autocars modifie sensiblement le niveau de service offert à l'utilisateur. L'effet de ces divers types de véhicules est quantifié par l'introduction du concept de *coefficient d'équivalence*, exprimé en u.v.p.s. Ce coefficient désigne le nombre e de voitures particulières que représente chaque catégorie dans les conditions de circulation

données. Le *Temps Inter-Véhiculaire (TIV)* d'une catégorie donnée équivaut à e fois celui d'un **VL**. Formellement, ce coefficient s'exprime par :

$$e = \frac{\text{TIV}(\text{catégorie})}{\text{TIV}(\text{VL})} .$$

La *régulation d'écart de TIV* est très répandue dans l'exploitation des autoroutes. Ce paramètre permet d'aborder les effets de la composition du trafic (types de véhicules) sur les conditions d'écoulement de la circulation.

L'écoulement du trafic sur la route est étroitement lié à de nombreux facteurs : les uns concernent la géométrie de la route, les autres son environnement, d'autres encore caractérisent la nature de la circulation. Dans cette dernière catégorie, la *composition du trafic* joue un rôle prépondérant.

L'espace entre véhicules – C'est la distance (spatiale), notée s , qui sépare à un instant donné l'avant des véhicules successifs sur une même voie de circulation. Ainsi, cette distance ne tient pas compte de la longueur des véhicules concernés, ou plutôt elle l'intègre sans la distinguer de l'espace (vide) entre les véhicules.

La vitesse – Un modèle microscopique trouve son principal intérêt dans l'étude de la vitesse de chaque véhicule pris indépendamment des autres. Le premier concept introduit est celui de *vitesse instantanée*. Pour un même véhicule, l'enregistrement des vitesses instantanées permet de caractériser le profil temporel de la vitesse. Cet histogramme est utile à l'obtention de divers paramètres. Il est révélateur de la qualité du trafic sur un itinéraire et s'emploie fréquemment dans les calculs relatifs à la consommation d'énergie des véhicules.

Sur un parcours de durée T , la *vitesse moyenne* v_{moy} d'un véhicule est définie par l'intégrale des vitesses instantanées sur la période considérée T :

$$v_{moy} = \frac{1}{T} \int_0^T v(t) dt \quad , \quad (1.1)$$

où $v(t)$ désigne la vitesse instantanée du véhicule à l'instant t .

L'accélération – L'accélération d'un véhicule suit les variations (augmentation, ou diminution si elle est négative) de la vitesse instantanée du véhicule. Sur un parcours de durée T , l'*accélération moyenne* a_{moy} d'un véhicule est définie par :

$$a_{moy} = \frac{1}{T} \int_0^T a(t) dt = \frac{v(T) - v(0)}{T} \quad ,$$

où $v(t)$ et $a(t)$ désignent respectivement la vitesse instantanée et l'accélération instantanée à l'instant t . Si les vitesses initiale et finale sont identiques, $a_{moy} = 0$.

Modèles de poursuite – Le courant de circulation est envisagé comme un mélange de véhicules *libres* et de véhicules *gênés* ou *suiveurs*. Les véhicules libres sont à même d’atteindre leur vitesse désirée, ce qui ne pose pas de problème de modélisation. Les véhicules gênés voient au contraire leur vitesse limitée par celle du véhicule qui les devancent. Ainsi, les *modèles de poursuite*, aussi appelés modèles *véhicule-suiveur*, sont utilisés pour décrire et expliquer ce mécanisme. Ils décrivent les interactions que chaque véhicule, pris individuellement, entretient avec le mouvement de l’ensemble des autres véhicules et l’infrastructure. Ces modèles considèrent le comportement de l’ensemble conducteur-véhicule comme un système dynamique qui interagit avec son environnement par l’entremise de ses accélérations et freinages. Ces actions sont traduites au moyen d’une équation différentielle décrivant la variation de vitesse du véhicule considéré, en respectant le véhicule qui le précède. Les modèles véhicule-suiveur ont été initiés par [Reuschel, 1950, Pipes, 1953] et plus amplement développés par [Herman et Potts, 1959].

Le phénomène de poursuite lie la réaction du conducteur à sa sensibilité et à l’intensité du stimulus définie par la différence de vitesse entre le véhicule considéré et son prédécesseur. La relation peut se résumer par l’équation suivante :

$$\text{réaction}(t + \tau) = \text{sensibilité} \times \text{stimulus}(t) \quad ,$$

où τ est le temps de réaction de l’ensemble conducteur-véhicule. Plus formellement, cette équation a d’abord revêtu une forme linéaire [Chandler *et al.*, 1958], puis non-linéaire [Gazis *et al.*, 1961].

Modèle de particules – Du fait de leur simplicité, les *Automates Cellulaires (ACs)* sont devenus un outil très répandu dans la modélisation microscopique du trafic. Leur utilisation consiste à considérer les véhicules comme des particules interagissant entre elles. La route est découpée en un nombre fini de *sites* ou *cellules* de longueur $7.5m$, ce qui correspond à la longueur moyenne des véhicules augmentée de l’espacement inter-véhiculaire. Chaque site peut être occupé ou vide. L’état de la cellule représente la vitesse du véhicule qui l’occupe, ou -1 si elle est vide. À chaque itération, l’état de l’ensemble du système est mis à jour en suivant des règles appliquées simultanément sur tous les véhicules. Les règles satisfont au moins les conditions suivantes :

- tous les véhicules veulent s’établir à la vitesse maximale ;
- les véhicules évitent les collisions entre eux.

Les modèles de particules du trafic apparaissent avec les travaux de [Gerlough, 1956] puis sont étendus par [Cremer et Ludwig, 1986, Biham et Middleton, 1992]. Un aspect stochastique a été ajouté dans [Fukui et Ishibashi, 1996].

1.1.3.B Niveau macroscopique

Les variables *macroscopiques* du trafic permettent de décrire globalement des *flots* de véhicules sur une section de route [Cohen, 1993]. Nous passons en revue les paramètres utilisés dans ce niveau d’étude.

Le débit – Le *débit* correspond à la répartition des véhicules dans le temps. Tout d'abord le débit moyen $q(t_1, t_2, x)$ au point d'abscisse x entre les instants t_1 et t_2 est défini par le rapport :

$$q(t_1, t_2, x) = \frac{n(t_1, t_2, x)}{t_2 - t_1} ,$$

dans lequel $n(t_1, t_2, x)$ désigne le nombre de véhicules passés en x entre les deux instants. Expérimentalement, le débit peut être déterminé par de simples comptages sur la route.

Dans certaines théories, le flot de véhicules est parfois considéré comme un milieu continu. Le débit $q(x, t)$ au point x et à l'instant t est alors défini par :

$$q(x, t) = \lim_{\Delta t \rightarrow 0} q\left(t - \frac{\Delta t}{2}, t + \frac{\Delta t}{2}, x\right) .$$

Une telle définition ne s'applique pas en ces termes à une théorie discrète de l'écoulement de la circulation, car cette limite vaudrait l'infini ou zéro, selon qu'un véhicule passerait ou non à l'instant t . Il s'agit donc d'une schématisation mathématique. En fait, nous considérons qu'il y a identité entre $q(x, t)$ et $q(t - 0.5\Delta t, t + 0.5\Delta t, x)$ pour de petites valeurs de Δt , par exemple de l'ordre de 20 secondes.

Soulignons au passage le lien avec l'approche microscopique : le débit moyen q est l'inverse de l'écart de temps inter-véhiculaire moyen h , pour un flot *stationnaire* (*i.e.* dont le débit ne varie pas beaucoup autour de sa moyenne) : $q = 1/h$. En effet, si N désigne le nombre de temps inter-véhiculaires observés pendant une période de temps T , au passage en un point donné, alors :

$$T = \sum_{i=1}^N h_i = N \times h_{moy} ,$$

où h_{moy} désigne l'intervalle de temps moyen. Le débit mesuré q vaut donc :

$$q = \frac{N}{T} = \frac{1}{h_{moy}} .$$

Par expérience, il est fréquemment admis dans la littérature que le débit maximum le plus réaliste est de l'ordre de 0.5 veh.s^{-1} .

La densité – La *densité* (ou *concentration*) décrit la répartition des véhicules dans l'espace. La concentration moyenne $k(x_1, x_2, t)$ à l'instant t sur une section de route délimitée par les points d'abscisse x_1 et x_2 correspond au rapport :

$$k(x_1, x_2, t) = \frac{n(x_1, x_2, t)}{x_2 - x_1} ,$$

dans lequel $n(x_1, x_2, t)$ désigne le nombre de véhicules présents sur la section à l'instant t . Une telle grandeur est directement observable par photographie aérienne ou encore par caméra vidéo.

Les théories continues définissent la densité $k(x, t)$ au point x et à l'instant t par :

$$k(x, t) = \lim_{\Delta x \rightarrow 0} k\left(x - \frac{\Delta x}{2}, x + \frac{\Delta x}{2}, t\right) .$$

À l'instar du débit, si Δx est de l'ordre de 50 à 100 mètres, il y a pratiquement identité entre les valeurs de $k(x, t)$ et $k(x - 0.5\Delta x, x + 0.5\Delta x, t)$.

Les approximations précédentes, relatives au débit et à la densité, ont des conséquences qui n'ont pas de sens physique. En effet, sur une petite section de route ou pendant un temps très court, il peut passer une certaine quantité de véhicules, par exemple un centième de véhicule. En pratique, dans les observations et les analyses, des longueurs et des durées qui définissent de telles quantités ne sont pas sujets à interprétation.

Si nous observons le passage de N véhicules, nous pouvons définir les densités individuelles k_i par : $k_i = 1/s_i$, où s_i est l'espacement entre le véhicule i et le véhicule qui le précède. La concentration moyenne k sur le segment vaut :

$$k = \frac{1}{\frac{1}{N} \sum_{i=1}^N \frac{1}{k_i}} .$$

Comme précédemment pour les débits, la densité moyenne est la moyenne harmonique des débits individuels. Remarquons enfin que concentration moyenne est l'inverse de l'espacement moyen, pour un courant de circulation dit *stationnaire* (*i.e.* lorsque le débit, la densité et la vitesse varient peu autour de leurs moyennes respectives) : $k = 1/s$.

Le taux d'occupation – Le *taux d'occupation* est aujourd'hui très couramment employé dans le domaine de l'exploitation, notamment parce qu'il peut être mesuré à l'aide de boucles magnétiques. Grandeur sans dimension, défini par la proportion de temps durant laquelle la boucle est occupée, le taux d'occupation τ est directement lié à la densité k par la relation : $\tau = (L + l)k$, où L et l désignent respectivement la longueur moyenne des véhicules et celle du capteur.

Les vitesses moyennes – Le fait de s'intéresser non plus à un seul véhicule isolé comme dans l'approche microscopique, mais à un flot de véhicules, autorise plusieurs définitions de la vitesse moyenne ; chacune d'entre elles répond au mieux à un objectif propre. Nous distinguons alors les définitions suivantes.

La vitesse moyenne temporelle – En un point fixé de la route, la *vitesse moyenne temporelle* u_t est la moyenne arithmétique des vitesses instantanées u_i des N véhicules passant pendant un intervalle de temps donné :

$$u_t = \frac{\sum_{i=1}^N \frac{d_i}{t}}{N} ,$$

où d_i est la distance parcourue par le i^{e} véhicule pendant la période t .

La vitesse moyenne spatiale – La notion de *vitesse moyenne spatiale* u_s , définie dans [Wardrop, 1952], s'avère plus utile en pratique. Ainsi, sur une section de route de longueur fixée, la moyenne arithmétique des vitesses des N véhicules à un instant donné est :

$$u_s = \frac{d}{\frac{\sum_{i=1}^N t_i}{N}} ,$$

où t_i est le temps nécessaire au i^{e} véhicule pour parcourir la distance d .

Les deux concepts de vitesse ainsi définis sont différents. Pour un flot de circulation à l'état stationnaire, nous établissons la relation suivante [Wardrop, 1952] : $u_t = u_s + \sigma^2 s / u_s$, où σ désigne l'écart-type de la distribution des vitesses spatiales. Nous observons donc l'inégalité $u_t \geq u_s$. D'autre part, l'espacement moyen s , l'écart de temps moyen h et la vitesse moyenne u_s sont liés par : $h = s / u_s$.

Modèles classiques – La première formulation macroscopique de l'écoulement du trafic a été proposée par [Lighthill et Whitham, 1955]. Le modèle proposé exploite les concepts de base définissant un fluide compressible. Une conséquence de la représentation continue est que la dimension des véhicules est négligée : n'importe quelle portion de route, si petite soit-elle, doit contenir une quantité de véhicule(s) (*i.e.* éventuellement moins de 1). Certains travaux proposent des modèles à *temps discrétisé* (ou *échantillonné*) : les mesures ont lieu à intervalles réguliers [Buisson *et al.*, 1995]. D'autres introduisent des approches [Payne, 1971, Papageorgiou et Schmidt, 1991, Aw et Rascle, 2000] capables d'améliorer respectivement la mise en œuvre et la précision des modèles initiaux.

Les modèles macroscopiques du trafic les plus courants assimilent le trafic à un fluide. Ceci permet d'exploiter les résultats de la mécanique des fluides. Ces modèles sont des corrélations entre les différentes grandeurs qui décrivent les tronçons étudiés selon la relation fondamentale. Quelques corrélations, tirées de [Transportation Research Board, 2000], sont données dans le tableau 1.1. La relation fondamentale permet de caractériser les tronçons à travers le diagramme fondamental qui en résulte.

Hypothèse	Équation	Taux critique	Capacité
Greenshields	$\frac{q}{\tau} = a\tau + b$	$-\frac{b}{2a}$	$-\frac{b^2}{4a}$
Drew	$\frac{q}{\tau} = a\sqrt{\tau} + b$	$-\frac{4b^2}{9a^2}$	$-\frac{4b^3}{27a^2}$
Greenberg	$\frac{q}{\tau} = a \ln(\tau) + b$	$\exp^{-1-\frac{b}{a}}$	$-a \exp^{-1-\frac{b}{a}}$
Underwood	$\frac{q}{\tau} = a \exp^{-b\tau}$	$\frac{1}{b}$	$\frac{a}{b \exp}$
May	$\frac{q}{\tau} = a \exp^{-b\tau^2}$	$\frac{a}{\sqrt{2b}}$	$\frac{a}{\sqrt{2b} \exp}$

TABLEAU 1.1 – Quelques exemples de fonctions permettant d’obtenir le diagramme fondamental. a et b sont des paramètres à définir.

La relation fondamentale – Dans un courant de circulation *isovélocique* (*i.e.* tous les mobiles ont la même vitesse), la vitesse commune est appelée *vitesse moyenne du flot* u . Dans ce cas simple, il est aisé d’établir la *relation fondamentale* : $q(x, t) = u \times \tau(x, t)$. Dans un cas quelconque, la vitesse du flot $u(x, t)$ correspond à la vitesse moyenne spatiale u_s . Pour un flot stationnaire, $u(x, t)$ est la moyenne harmonique des vitesses des véhicules passant au point x pendant une durée Δt .

Le diagramme fondamental – Le flot est supposé homogène et stationnaire. À faible densité de véhicules, Nous observons une vitesse moyenne u du flot généralement élevée, appelée *vitesse libre* ou *vitesse désirée* u_{\max} . À l’inverse, lorsque τ augmente, les interactions entre les véhicules deviennent plus importantes et u diminue. Ce constat suggère que u est une fonction monotone décroissante de τ . L’hypothèse $u = u(\tau)$ s’appelle hypothèse du *diagramme fondamental*. La forme habituellement observée est présentée par la figure 1.1.

Compte tenu de la relation liant les trois variables, la loi d’écoulement du trafic sur une section de route peut donc être formulée par une fonction liant deux des trois variables de débit, taux d’occupation et vitesse.

Le diagramme fondamental est sensible à de nombreux facteurs comme la géométrie de la route, la nature et la composition du trafic, les conditions météorologiques, les mesures d’exploitation, *etc.* Outil pratique, bien adapté à divers problèmes de circulation, il joue un rôle précieux dans la conception et l’exploitation des infrastructures routières.

Le schéma de la figure 1.1 est la représentation la plus courante pour étudier un segment de route car il permet d’extraire visuellement les caractéristiques les plus importantes de l’objet de l’étude. En effet, il est possible d’identifier d’une part la capacité du segment donnée par le débit maximal et d’autre part le taux d’occupation critique qu’il ne faut pas dépasser pour éviter la congestion. En plus de ses qualités visuelles, la fonction identifiée du diagramme fondamental permet non seulement de calculer quelques indicateurs du trafic tels que les temps de parcours

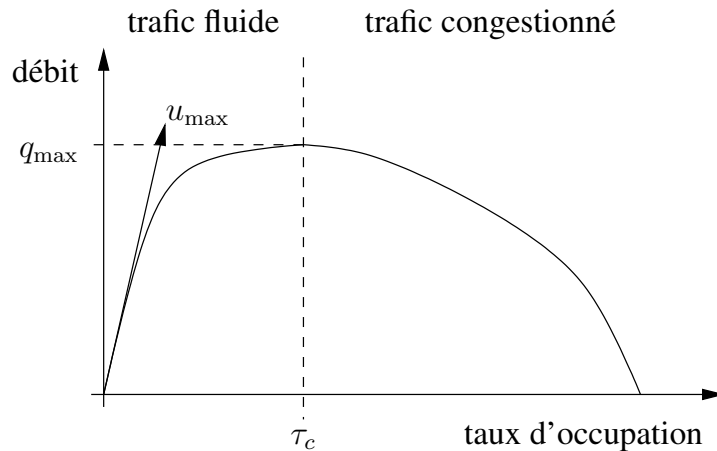


FIGURE 1.1 – Le diagramme fondamental donne le débit en fonction du taux d'occupation

et la détection d'incidents, mais aussi elle est la base du modèle de poursuite utilisé au niveau microscopique pour la simulation.

1.1.4 Simulation du trafic

La plupart des simulateurs de trafic professionnels récents utilisent des modèles microscopiques *multimodaux*, c'est-à-dire qui prennent en considération plusieurs modes de transport, typiquement les voitures et les transports en commun, parfois aussi les piétons et les bicyclettes [Bhouri, 2002]. Les aspects pris en compte dans tous les modèles utilisés sont :

- la circulation sur la voie courante,
- le changement de voies,
- le dépassement,
- les croisements et les carrefours signalisés ou non,
- les opérations de transport public.

D'autres aspects de la simulation microscopique sont souvent également inclus, notamment :

- les incidents,
- les rond-points,
- les piétons ; les cyclistes et autres deux-roues,
- la recherche d'un lieu de stationnement,
- les conditions atmosphériques.

Les *modèles continus* représentent le changement d'état du système continuellement dans le temps. Cependant, étant donné que les ordinateurs sont numériques par nature, tous les modèles de simulation de trafic opérationnels sont discrets. Ainsi, deux types de discrétisation sont utilisées :

- La *discrétisation temporelle* consiste à partager le temps en intervalles réguliers et finis, appelés *périodes* ou *pas d'échantillonnage*. Les changements affectant l'état du système sont calculés pour chacun de ces intervalles.

- La *discrétisation événementielle* considère que le changement d'état est dû à un événement, comme le changement de couleur des feux par exemple. Cette méthode de modélisation permet de réaliser une économie de temps de calcul lorsque le système ne change pas souvent d'état.

Les modèles à discrétisation temporelle sont plus faciles à élaborer, mais sont plus coûteux en temps de calcul. En effet, il est plus simple de chercher les caractéristiques d'un carrefour à chaque changement de l'état des feux plutôt qu'à chaque seconde. En revanche, le suivi des files d'attente est plus précis s'il est réalisé par un tel modèle.

Le tableau 1.2 présente un classement succinct des outils de simulation les plus connus [Bhourri, 2002].

Modèle	Origine	Discrétisation	Piétons	Bicyclettes
CORSIM	États-Unis	temporelle	✓	✓
DRACULA	Angleterre	temporelle	✓	✓
FLEXYT II	Pays-Bas	événementielle	✓	✓
HUTSIM	Finlande	temporelle objet	✓	✓
MICSTRAN II	Japon	temporelle	✓	✓
NEMIS	Italie	temporelle	✓	
PLANSIM-T	Allemagne	temporelle objet		
SIGSIM	Angleterre	événementielle		
SITRA B+	France	temporelle		
VISSIM	Allemagne	temporelle	✓	

TABLEAU 1.2 – Modèles de simulation microscopiques multimodaux

1.2 Régulation des carrefours urbains à feux

La régulation du trafic est un domaine vaste dans lequel plusieurs techniques et formes de signalisation sont utilisés pour augmenter la sécurité des usagers, réduire toute sorte de nuisances et rationaliser l'exploitation des infrastructures routières. Ainsi, la régulation du trafic agit sur divers aspects tels que la limitation des vitesses ou l'orientation en temps réel des conducteurs à travers des panneaux lumineux pour éviter les congestions.

Parmi les sujets les plus importants de la régulation du trafic, nous trouvons la gestion des carrefours. En effet, ces derniers sont omniprésents, et engendrent un problème de conflit entre leurs utilisateurs. Plusieurs problématiques sont ainsi posées : la régulation des rampes d'accès aux autoroutes, l'ergonomie des carrefours ou encore la sécurité de ces lieux conflictuels. En revanche, notre étude ne porte sur aucune de ces problématiques : elle se focalise sur la régulation des carrefours urbains. Plus précisément, nous nous penchons sur le partage de la surface conflictuelle formée par le croisement de plusieurs rues. Ce partage peut être arbitraire s'il est laissé à l'interprétation que font les usagers des règles définies par le code de la route [Anonyme, 2004], telles que la priorité à droite ou les panneaux statiques. Toutefois, notre intérêt porte au contraire sur les cas où il est déterminé par les feux de circulation.

Dans nos villes, deux modes de régulation des carrefours par les feux cohabitent : plan de feux et régulation adaptative. Le premier est plus ancien et moins coûteux, tandis que le deuxième est plus complexe et performant. C'est pour cette raison que nous présentons les plans de feux en premier. Cependant, avant de traiter de la régulation des feux, nous allons définir le carrefour urbain à feux.

1.2.1 Carrefour urbain à feux

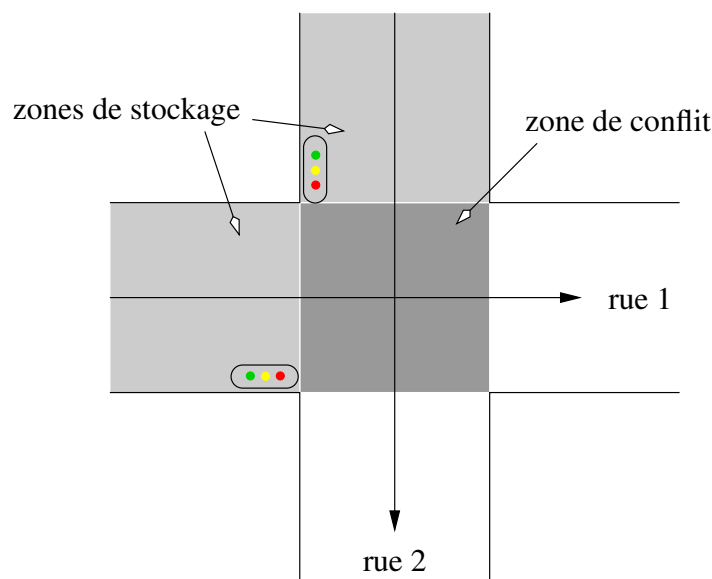


FIGURE 1.2 – L'intersection : un carrefour simple, croisement de deux rues à sens unique

Nous reprenons dans cette partie un certain nombre de définitions du guide de l'*Association Française de NORMALISATION (AFNOR)* [Anonyme, 1991]. Comme annoncé précédemment, nous étudions le trafic routier dans une catégorie particulière de son environnement, à savoir les carrefours à feux.

- Les *carrefours* sont les lieux de croisement d'au moins deux routes. Un tel croisement peut revêtir de nombreuses formes, tant dans l'infrastructure que dans les règles de franchissement. Une *route* est caractérisée par sa longueur, son nombre de voies, ainsi que par le sens de circulation et la vitesse maximum autorisée sur les voies. La figure 1.2 représente un carrefour simple, composé de deux routes à sens unique : ce type de carrefour sera appelé *intersection* par la suite. Cette figure montre également les zones fonctionnelles : la *zone de conflit*, commune aux deux routes et les *zones de stockage* en amont de la zone de conflit. Un carrefour peut sembler très complexe, mais souvent il peut être décomposé en plusieurs intersections plus simples, suffisamment rapprochées voire imbriquées pour apparaître aux yeux de l'utilisateur comme un seul carrefour.

L'accès au *carrefour à feux* est contrôlé par des signaux lumineux situés à la frontière entre la zone de stockage et la zone de conflit. Nous nous intéressons par la suite essentiellement aux *signaux tricolores*, appelés aussi *feux tricolores* ou *de circulation*. Un feu

tricolore est caractérisé par la couleur du signal allumé (vert, orange, rouge), un éventuel pictogramme désignant les catégories de mobiles concernées et le fait qu'il clignote ou non. La façon de gérer les feux de circulation constitue la *méthode* ou *stratégie de régulation* du carrefour et est discutée dans la section 1.2 page 25.

- Un *mouvement* est défini par son origine et sa destination. Une voie n'est pas allouée à chaque type de mouvement, mais ceux-ci sont au contraire regroupés en courants sur une même voie mixte. Un *courant de mobiles* est donc l'ensemble des mouvements des véhicules qui proviennent d'une même branche d'accès à un carrefour et qui sont admis simultanément au moyen d'un même groupe de signaux.
- Un *point !conflictuel* résulte de la rencontre, en un même lieu, de mouvements antagonistes de mobiles. Certains trouvent une solution dans le respect du code de la route. Cependant, ils doivent être examinés avec soin en fonction de la visibilité et de l'importance respective des mouvements (*cf.* la réglementation en vigueur dans [Anonyme, 2002]). D'autres doivent être systématiquement éliminés dès lors que l'on implante une signalisation lumineuse d'intersection. La zone de conflit d'un carrefour quelconque est constituée de l'ensemble de ses points conflictuels. Chacun relève d'un degré d'importance :
 - *primaire* s'il concerne deux mouvements directement sécants. Pour des raisons de sécurité, il est nécessaire de n'autoriser l'accès à un tel point de conflit qu'à un seul des mouvements concernés à la fois ;
 - *secondaire* lorsqu'un mouvement est gêné par un autre, par exemple lorsque les véhicules tournent à gauche. Ce type de conflit est résolu, la plupart du temps, par les règles du code de la route. Toutefois, pour des raisons de fluidité ou de sécurité, il arrive que les mouvements concernés ne soient pas autorisés à accéder à la zone de conflit simultanément ;
 - *tertiaire* s'il est dû à la géométrie d'un tronçon acceptant plusieurs modes, comme par exemple dans le cas d'une rue à double sens mais étroite, ou encore une voie d'insertion trop courte.

Les feux tricolores apportent beaucoup à la régulation du trafic. Tout d'abord, ils suppriment certains points conflictuels, car ils permettent de découper le temps d'utilisation des zones de conflits pour les différents groupes de courants antagonistes. Le choix des durées de chaque état des feux et la synchronisation de ceux-ci entre eux interviennent dans la gestion de l'écoulement du trafic.

1.2.2 Plans de feux

La durée pendant laquelle un ou plusieurs courants sont admis simultanément dans le carrefour constitue une *phase*. Dans le cadre d'une signalisation lumineuse d'intersection, l'enchaînement des différentes phases constitue le *cycle* des feux. Le *temps de cycle* désigne la durée entre deux passages successifs de l'ensemble des feux d'un carrefour par la même phase, dans le cas où ces phases ne sont pas escamotées. Dans le cas contraire, la stratégie est dite *acyclique*.

Le *plan de feux* consiste à construire et agencer des phases de feux. Celles-ci sont définies par leur durée et par l'attribution d'une couleur de feu pour chaque courant de mobile impliqué dans le carrefour considéré. Le plan de feu le plus simple consiste à répéter

indéfiniment la même séquence de phases de durées fixes, toujours agencées dans le même ordre, de manière à constituer un *cycle fixe*. Le calcul de la durée totale de ce cycle et des phases qui le compose a été l'un des premiers problèmes posés à l'ingénierie du trafic [[Transportation Research Board, 2000](#)].

Une des difficultés tient à la variabilité de la demande du trafic. En effet, sur une entrée de carrefour, cette demande fluctue de façon importante au cours d'une même journée ; elle est différente selon qu'il s'agit d'un jour ouvrable, de fin de semaine ou de vacances par exemple. Elle peut aussi fluctuer en fonction d'événements exceptionnels. Deux types de variations du trafic se dessinent : certaines sont régulières et planifiables quand d'autres sont inattendues et aléatoires. Ainsi, les migrations pendulaires des jours ouvrables (matin et soir) sont à l'origine de changements prévisibles : des manifestations, des travaux ou des accidents, impliquent souvent des augmentations inopinées du trafic.

Cependant, un plan de feux fixe (*i.e.* cycle et phases invariables) ne permet pas d'offrir un bon niveau de service aux usagers. Il faut donc chercher à adapter l'offre de l'infrastructure à la demande du trafic. Une multitude de méthodes et outils existants s'ajoutent au plan de feu fixe. Celui-ci sert généralement de base aux systèmes de régulation présentés dans cette sous-section ; s'y ajoutent différentes approches destinées à tenir compte des fluctuations du trafic [[Braban et Boillot, 2003](#)], parmi lesquelles nous retrouvons :

- la *programmation horaire*, également appelée *multiprogrammation à temps fixe* entre dans le cadre des variations régulières. Cette méthode consiste à disposer de plusieurs plans de feux, selon des situations-types prédéfinies, puis à les mettre en œuvre selon un calendrier programmé. Dans la plupart des configurations, trois plans de feux suffisent : un pour les heures de pointe (matin, midi et soir), un pour les heures creuses du jour et un pour celles de la nuit. Une méthode classique de calcul des plans de feux est décrite dans [[Cohen, 1993](#)] ;
- la *micro-régulation* est utilisée pour les variations exceptionnelles et aléatoires. Elle prend en compte l'analyse instantanée de la demande pour modifier la durée, l'occurrence et/ou l'ordre des phases dans le déroulement d'un plan de feux. Elle agit uniquement à l'échelle d'un carrefour ou d'un petit ensemble de carrefours ;
- la *macro-régulation* convient pour tous les types de variations, puisqu'elle agit à l'échelle d'un groupe de carrefours sur des durées plus longues, pour coordonner voire synchroniser les carrefours. L'*Institut National de REcherche sur les Transports et leur Sécurité (INRETS)* a, par exemple, développé un programme baptisé CLAIRE [[Scemama, 2006](#)] qui peut être couplé à un système de micro-régulation. Ce logiciel permet de détecter les congestions et fournit de nombreuses statistiques sur le trafic. Il recommande également des comportements (favoriser ou retenir le trafic) pour chaque axe.

Ces méthodes ne s'opposent pas et doivent au contraire être utilisées en parallèle. Ensemble, elles assurent au système de gestion des plans de feux une capacité d'adaptation au trafic non négligeable ainsi qu'une efficacité réelle dans de larges plages de fonctionnement, dans la mesure où un suivi et une maintenance du système sont assurés.

Nous pouvons classer quelques uns des outils de gestion de trafic les plus connus en fonction des objectifs qu'ils permettent d'atteindre :

- Certains outils favorisent la technologie embarquée et individuelle pour chaque véhicule,

ceci afin de conseiller au mieux l'utilisateur quelque soit les possibilités topologiques et interactives de l'infrastructure. Une fois entrés les points d'origine et de destination, ALI-SCOUT, ULYSSE ou AUTOGUIDE optimisent le trajet à suivre en fonction des informations du trafic et de l'état de l'environnement.

- Jusqu'à aujourd'hui, les plans de feux à cycle fixe sont encore fréquemment rencontrés. Des outils tels TRANSYT proposent de calculer la séquence et les durées de phases qui conviendront dans la majorité des cas.
- Il peut s'avérer souhaitable d'avoir un système un peu plus souple, sans toutefois augmenter les dépenses : le logiciel GERTRUDE [Morrish, 1980] offre un panel de plans de feux pré-calculés, prêts à être mis en place en fonction des heures et des événements prévus.
- La synchronisation de plusieurs carrefours successifs qui concernent, par exemple, un axe important, doit être réalisée. En effet, il est souhaitable que les véhicules évoluant sur cet axe rencontrent des feux verts de manière continue : ceci est appelé l'*onde verte*. ICARE, TALON (anciennement ACHILLE) ou CASIMIR permettent de calculer les cycles (fixes) à appliquer à chaque intersection pour obtenir un tel résultat dans les meilleures conditions.
- TACOS et SAGE [Foraste et Scemama, 1987] s'attaquent à une autre situation particulière, à savoir, les intersections saturées. Leur utilisation fournit la démarche à suivre pour évacuer les véhicules au plus vite.

Depuis une vingtaine d'années est apparue une nouvelle famille de systèmes de régulation de trafic, dits *adaptatifs* ou *temps réel*. Ces systèmes tendent à se démarquer des traditionnels plans de feux en offrant davantage de souplesse dans le choix des états de feux. Ceci est rendu possible par une adaptation permanente et continue de la commande en fonction de l'évolution du trafic, ce qui permet en outre de s'affranchir des mises à jour au fil des années et des événements. La majorité des logiciels cités précédemment ont évolué et proposent désormais de la régulation adaptative.

1.2.3 Régulation adaptative

Les nombreux systèmes de régulation adaptative diffèrent principalement sur cinq points, que nous parcourons dans cette sous-section.

1.2.3.A Horizon de planification

Les systèmes de trafic adaptatifs emploient un modèle pour estimer les diverses informations du trafic jusqu'à un certain point dans le temps. La durée entre ce point et l'instant présent est appelée *horizon de planification*. La longueur de cet horizon ainsi que la façon dont il est découpé en intervalles successifs diffèrent d'un système adaptatif à l'autre. Cependant, la plupart du temps, l'horizon a une taille fixe (de une à deux minutes) et est découpé en intervalles fixes. OPAC, PRODYN, SPPORT et ALLONS-D ont utilisé un pas de cinq secondes.

Dans le cas où l'horizon est choisi trop court et que l'algorithme d'optimisation est confronté au choix, soit de servir complètement une phase qui a un taux de dispersion faible, soit de pré-

empter cette phase de manière à permuter vers une phase qui a un taux de dispersion plus important, il choisirait, contrairement à ce que l'intuition laisserait supposer, la deuxième possibilité. C'est pourquoi beaucoup de systèmes adaptatifs qui emploient des horizons plus courts ont introduit des coûts terminaux, afin de pénaliser les files d'attente résiduelles à la fin de l'horizon [Newell, 1998, Shelby, 2004].

L'algorithme d'ALLONS-D a une approche différente puisqu'il considère que la longueur de l'horizon est directement lié à l'état courant du trafic. Ainsi, il élargit l'horizon jusqu'à ce qu'il trouve une solution dans laquelle toutes les arrivées prévues sont traitées. Bien que l'idée d'un horizon qui puisse se comprimer ou s'étirer en suivant les aléas du trafic soit attirante, il n'en n'est pas ainsi dans le cas d'ALLONS-D. Dans des conditions saturées (*i.e.* lorsqu'il y a beaucoup d'arrivées prévues), l'horizon peut devenir si grand que la méthode d'optimisation utilisée peut ne pas délivrer une réponse à temps.

1.2.3.B Méthode d'optimisation

L'objectif de la commande est d'agir sur le système nominal de manière à optimiser ses performances couvrant l'horizon de planification. Le *coût* est un indice de performance calculé à partir des grandeurs mesurées ou estimées du système. Il augmente avec l'allongement des temps de parcours, des files d'attente, ou du temps de présence des véhicules dans le système. Ainsi, la commande consiste à déterminer la séquence des actions $[u_0, u_1, \dots, u_T]$, appelée *politique* ou *trajectoire de commande* de telle sorte que le coût attendu soit minimisé. Dans le cas d'un horizon de planification infini, il est d'usage d'appliquer un facteur de réduction $\gamma < 1$ aux coûts futurs afin d'obtenir une estimation finie du coût à partir de l'état courant i , dénoté $f(i)$. La valeur du coût optimal, dénoté $f^*(i)$, est une fonction du coût immédiat de l'application de la commande, ajoutée au coût estimé du prochain état. Cette relation est contenue dans la relation récursive suivante, appelée *équation de Bellman* :

$$f^*(i) = \min_{u \in U(i)} \left\{ c_i(u) + \gamma \sum_{j \in S} p_{i,j}(u) f^*(j) \right\}, \quad (1.2)$$

où, partant de l'état i pris dans l'ensemble des états possibles S , $U(i)$ est l'ensemble des actions disponibles, $c_i(u)$ le coût de l'action u , $p_{i,j}(u)$ la probabilité d'entrer dans l'état j par l'action u .

Étant donné que l'espace de décision est structurée sous forme d'*arbre*, la recherche de la séquence optimale consiste en la construction de cet arbre. Une recherche exhaustive de l'espace de décision résulte en la construction d'un arbre complet. Puisque l'espace de recherche croît de manière exponentielle avec la taille du problème, il n'est pas possible d'explorer toutes les possibilités, sauf pour les plus petits problèmes. Le seul moyen de s'en sortir est de ne pas chercher l'espace entier. L'efficacité de la recherche de l'espace de décision dépend du degré auquel l'arbre ne devra pas être construit pour trouver le chemin optimal. Dans [Shelby, 2004], plusieurs algorithmes bien connus sont évalués sur la base de leurs complexités et de la qualité de leurs résultats (en terme de retards).

Bien que l'état de trafic résultant d'une décision dépend d'une loi probabiliste (voir équation (1.2)), la plupart des systèmes adaptatifs dont nous parlons dans ce mémoire utilisent un

modèle déterministe. Du fait de cette simplification, les diagrammes d'arbres de décision correspondant au problème de commande des signaux de trafic négligent généralement l'usage de nœuds d'actions explicites. En effet, chaque action résulte en un seul état possible.

La programmation dynamique (**PRODYN**, **OPAC**, **SPPORT**) et les algorithmes de séparation et évaluation progressive (**ALLONS-D**, **SPOT**), ainsi que leur combinaison (**RHODES**), sont les techniques prédominantes dans les systèmes de trafic adaptatifs. Toutefois, d'autres travaux proposent des méthodes d'optimisation alternatives, notamment les algorithmes génétiques [Park *et al.*, 2000], la logique floue [Lee et Lee-Kwang, 1999, Niittymäki, 2001], les réseaux de neurones [Hua et Faghri, 1995, Chin *et al.*, 1999] et même une combinaison de ces deux dernières méthodes [Bingham, 2001].

1.2.3.C Liberté d'action

La largeur de l'arbre à explorer est dépendante du nombre de décisions que peut prendre le système de commande à chaque instant. Dans sa forme la plus épurée, le choix disponible est celui d'étendre la phase courante ou commuter vers la phase suivante. Ceci est l'approche choisie par **OPAC**, **PRODYN**, **ALLONS-D** et **SPPORT**. Bien que cette approche diminue significativement le nombre d'options considérées, il ne permet pas de construire une séquence de phases arbitraire. Dans sa forme la plus élaborée, le choix est celui de la prochaine phase. Cette approche libère l'arrangement des phases, au prix d'une extension de l'arbre de décision. C'est l'approche choisie par **UTOPIA**. Les deux approches sont schématisées dans la figure 1.3.

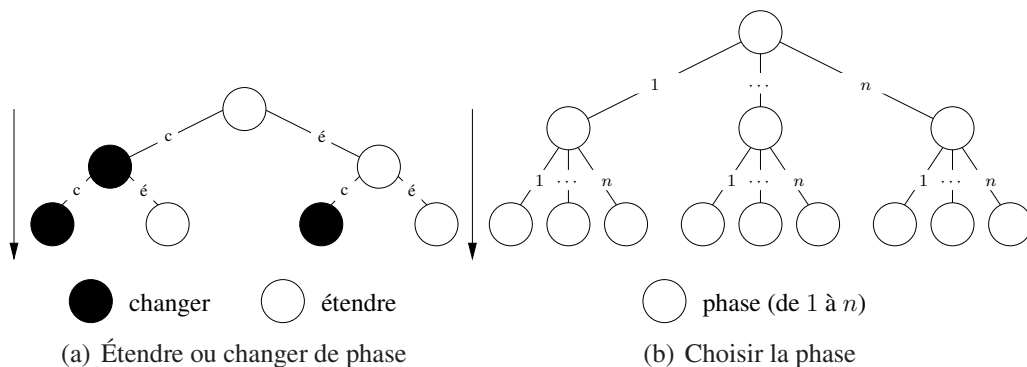


FIGURE 1.3 – Différentes conceptions des actions à considérer

Un compromis entre ces deux extrêmes consiste à considérer le saut de phases. Lorsque le saut de phases est autorisé, toutes les séquences peuvent être atteintes. Cette approche, choisie par **COP**, est montrée par la figure 1.4. L'inconvénient est que si la séquence de phases est mal construite au départ, le gain en terme de largeur de l'arbre est contrecarré par un accroissement de sa profondeur. Cependant, il est en principe permis de choisir la prochaine phase de manière dynamique.

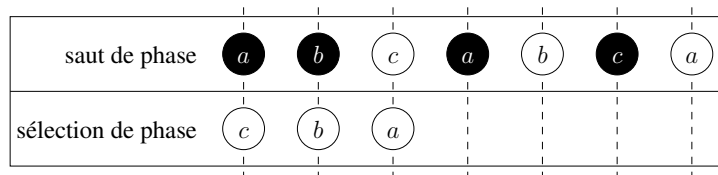


FIGURE 1.4 – Comparaison entre le saut et la sélection de phase

1.2.3.D Fréquence de rafraîchissement

Les systèmes adaptatifs de trafic se basent sur des arrivées prévues de véhicules. L'augmentation de la distance à partir de laquelle ces arrivées sont prévues fait souvent décroître la pertinence de ces prédictions. C'est pourquoi un *horizon glissant* est souvent utilisé. Le concept d'horizon glissant trouve ses origines dans le domaine de la recherche opérationnelle et est utilisé pour déterminer une politique à court terme basée sur une analyse à plus long terme. Tous les systèmes adaptatifs comparés dans cette section qui dépendent de prédictions emploient le concept d'horizon glissant. Ces algorithmes implémentent seulement les (quelques) premières actions de la planification, après quoi une nouvelle optimisation est calculée.

1.2.3.E Modèle de temps

Comme nous l'avons vu, les modèles microscopiques sont idéaux pour simuler le trafic, notamment le comportement des usagers. Ils permettent de préparer les différents intervenants aux procédures à suivre et aux délais lors de situations d'urgences (accidents, travaux, *etc.*). *A contrario*, les modèles macroscopiques peuvent être utilisés pour la supervision en temps réel de très grands systèmes routiers, ainsi que pour des analyses statistiques *a priori* ou *a posteriori* du trafic. Ceci facilite le dimensionnement des infrastructures en place ou en phase de conception.

En revanche, aucune de ces approches n'est appropriée à la commande en temps réel d'un carrefour à feux. En effet, dans le premier cas, la prise en compte d'un trop grand nombre de paramètres rend le temps de calcul de l'état du système incompatible avec une application temps réel. Dans la deuxième approche, la mesure des variables doit être effectuée sur une longue période afin d'obtenir des valeurs moyennes qui finalement sont le reflet d'un état passé au moment où la commande peut être calculée. L'utilisation de moyennes en elle-même rend d'ailleurs la commande imprécise et ne tient pas compte de chaque véhicule individuellement.

Ainsi, les premières versions des logiciels de commande de trafic utilisent des modèles de files d'attente verticales. Moins précis que la simulation, ces modèles offrent un bon compromis pour l'ajustement de la commande en temps réel. Cependant, ils ne permettent qu'une optimisation locale, sans interconnexions entre les carrefours. C'est pour cela que la seconde génération d'outils est basée sur des modèles de files d'attente horizontales, de telle sorte que les files des carrefours en amont alimentent directement celles des intersections en aval.

En résumé, les outils tels que [SCOOT](#) [Bretherton, 1989, Bretherton et Bowen, 1990], [SCATS](#) [Sims et Finlay, 1984], [SPPORT](#) [Dion et Hellinga, 2001, Dion et Hellinga, 2002] ou encore [ALLONS-D](#) [Porche *et al.*, 1996, Porche, 1998] peuvent apporter des modifications des

temps de cycle, de phases et d'escamotage de phases en fonction des fluctuations mesurées en temps réel.

Soulignons aussi que certains outils gèrent les feux d'une manière moins séquentielle. Par exemple, **PRODYN** [Henry *et al.*, 1983, Barriere *et al.*, 1986, Henry et Farges, 1989], **OPAC** [Gartner, 1983, Gartner *et al.*, 1995, Gartner *et al.*, 1999], **UTOPIA** [Di Taranto et Mauro, 1989] et **COP** appliquent des règles de changement de phase en fonction de la situation présente ou estimée du trafic. **RHODES** [Sen et Head, 1997, Mirchandani et Head, 2001] conjugue quant à lui les deux aspects ci-dessus, (changement de l'ordre et des durées des phases).

Enfin, il existe des approches inspirées de domaines distincts de l'ingénierie du trafic, comme par exemple l'informatique ou l'industrie. Ainsi, **CRONOS** et **ANTARES** sont basés sur un algorithme d'optimisation des files d'attente. La notion de cycle de feux disparaît alors, au profit d'une prise en compte individuelle de chaque véhicule.

Bien que nos villes soient dotées de plus en plus d'outils modernes de régulation des carrefours, elles restent toujours victimes des problèmes de congestion, de sécurité et de nuisance liés au trafic. Ceci peut être expliqué par la croissance continue de la demande de déplacement qui fait atteindre ses limites aux modes de régulation actuellement en place. Ainsi, le besoin se fait sentir aujourd'hui de changer radicalement la manière d'analyser le trafic. Plus précisément, les modèles à développer doivent partir d'un nouveau point de vue qui va au-delà du traitement actuel des mobiles sous forme de lots. Celui-ci peut être encouragé par les nouveaux moyens dont disposent les ingénieurs du trafic, à savoir des dispositifs embarqués et couplés au **GPS**, ou bien des caméras, qui offrent des possibilités bien plus grandes que les boucles magnétiques ou les capteurs pneumatiques.

1.3 Approches émergentes de régulation

Comme nous l'avons vu, les méthodologies de régulation des feux de circulation actuellement implantés dépendent fortement d'approches dont les limites ont été atteintes. Les approches microscopiques et macroscopiques au sens classique ont en effet été développées pour exploiter des capteurs tels que les boucles magnétiques, qui ont une capacité d'information limitée. Par exemple, le débit est estimé à partir de mesures passées et ne reflète pas l'état du trafic (*i.e.* fluide ou congestionné). En plus de l'estimation du débit, des détecteurs plus récents tels que les caméras vidéos, le **GPS** ou les ondes radio offrent des possibilités supplémentaires que ces modèles ne permettent pas d'exploiter.

Dans cette section, nous survolons quelques nouvelles approches de modélisation et de commande en temps réel des feux de circulation. Nous présentons plus en détail certaines commandes innovantes, puisant leurs origines dans des domaines variés.

1.3.1 Approche basée sur la distribution en différé

Dans [Dinopoulou *et al.*, 2000], les auteurs présentent la stratégie de commande du trafic « *Traffic-responsive Urban Control (TUC)* ». Son but est de fournir, à intervalles réguliers défi-

nis de manière appropriée, les paramètres de feux en fonction du trafic, en prenant en compte la situation du trafic dans son ensemble.

1.3.1.A Modélisation du système

Le modèle d'écoulement en différé utilisé dans [Dinopoulou *et al.*, 2000] est basé sur un modèle de *distribution en différé* (« *stock and forward* » en anglais) inspiré des problèmes de réseaux informatiques et présenté dans [Diakaki, 1999]. Le principe est simple : chaque intersection est modélisée par un nœud. Les voies reliant les intersections sont schématisées par des arcs orientés. Un exemple de réseau urbain modélisé par cette technique est donné par la figure 1.5.

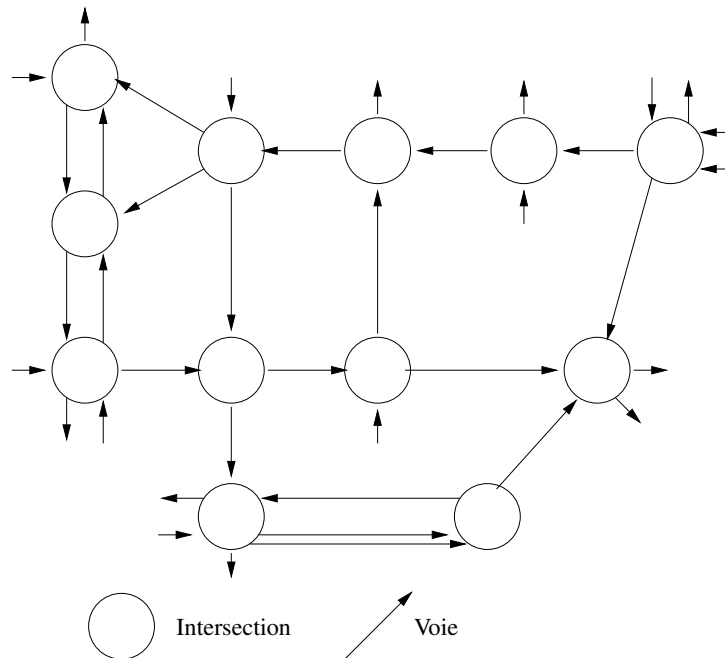


FIGURE 1.5 – Modélisation d'un réseau urbain par distribution en différé

1.3.1.B Critères d'optimisation

Les critères à optimiser sont :

- le temps d'attente aux entrées du réseau considéré,
- le temps de parcours total,
- le temps total passé dans le système,
- la consommation de carburant.

Le système à modéliser doit respecter les hypothèses suivantes :

- les temps de cycle et les temps perdus aux jonctions sont fixes ;
- le début de la phase principale de chaque feu est fixé ;
- la commande des feux de chaque intersection est basée sur un nombre fixe de phases ;

- les flux de saturation et les itinéraires (séquence de nœuds) sont fixes et connus.

1.3.1.C Stratégie de commande

Dans la stratégie de supervision présentée dans [Dinopoulou *et al.*, 2000], les véhicules sont regroupés en « paquets », définis par un nœud d'origine et un nœud de destination. Durant leurs trajets, les paquets sont stockés (*stock*) à chaque nœud, puis transmis (*forward*) au nœud suivant.

1.3.1.D Commentaires et critiques

Les auteurs observent, à travers des simulations, une forte amélioration des critères à optimiser. Cependant, ils constatent également que le modèle utilisé et la structure du régulateur rendent l'approche rigide, au sens où la moindre modification du système considéré entraîne de larges ajustements du modèle et un nouveau calcul de la commande. De plus, les intervalles de commande ne peuvent être plus courts que les temps de cycle des jonctions signalisées considérées. Autrement dit, les décisions ne peuvent être prises qu'à la fin du plus grand cycle employé. Enfin, l'effet de l'extension du vert aux carrefours consécutifs ne peut pas être directement constaté.

1.3.2 Approche basée sur le PCLE

Le travail réalisé par [De Schutter et De Moor, 1997] est basé essentiellement sur la commande optimale des feux de signalisation dans une intersection élémentaire. Une fois que le taux des arrivées et le taux maximal des départs des véhicules dans cette intersection sont connus, il est possible de calculer les commutations des feux de signalisation qui minimisent plusieurs critères.

Anticipant sur la présentation qui va suivre, nous pouvons dire que cette méthode, appliquée aux carrefours à feux, tire profit des propriétés bien établies d'une méthode connue dans la littérature sous le terme « *Problème de Complémentarité Linéaire Étendu (PCLE)* ».

1.3.2.A Modélisation du système

Les auteurs ont modélisé l'évolution des files d'attente dans un carrefour à feux tel que schématisé par la figure 1.2 page 26. Celui-ci est considéré comme étant un système de quatre lignes L_1 , L_2 , L_3 et L_4 . Chaque ligne représente une file de véhicules. À chaque coin de rue, un feu est installé : ils sont repérés par T_1 , T_2 , T_3 et T_4 .

Afin de pouvoir présenter cette approche, un certain nombre de notations doivent être introduites. Soit $i = 1, 2, 3, 4$:

- $l_i(t)$ est la longueur de la file d'attente dans la ligne L_i à l'instant t ;
- λ_i est le taux moyen des arrivées de véhicules dans la ligne L_i ;
- μ_i est le taux moyen des départs de véhicules dans la ligne L_i ;
- t_0, t_1, t_2, \dots sont les instants de commutation des feux de signalisation. Ils définissent les phases, comme le montre le tableau 1.3 ;
- $\delta_k = t_{k+1} - t_k$, $k \in \mathbb{N}$ est la durée entre deux instants de commutation consécutifs.

	T_1	T_2	T_3	T_4
$t_1 - t_0$	rouge	vert	rouge	vert
$t_2 - t_1$	vert	rouge	vert	rouge
$t_3 - t_2$	rouge	vert	rouge	vert
\vdots	\vdots	\vdots	\vdots	\vdots

TABLEAU 1.3 – Commutations des feux de signalisation.

Le comportement dynamique de la ligne L_1 est modélisé, pour tout $k \in \mathbb{N}$, par :

$$\begin{cases} l_1(t_{2k+1}) = \max(l_1(t_{2k}) + \lambda_1 \delta_{2k}, 0), & \text{si } T_1 \text{ est rouge,} \\ l_1(t_{2k+2}) = \max(l_1(t_{2k+1}) + (\lambda_1 - \mu_1) \delta_{2k+1}, 0), & \text{si } T_1 \text{ est vert.} \end{cases}$$

De la même manière, nous pouvons modéliser le comportement dynamique des autres lignes. Ainsi, si nous posons :

$$\begin{aligned} x_k &= [l_1(t_k) \quad l_2(t_k) \quad l_3(t_k) \quad l_4(t_k)]^T, \\ b_1 &= [\lambda_1 \quad \lambda_2 - \mu_2 \quad \lambda_3 \quad \lambda_4 - \mu_4]^T, \\ b_2 &= [\lambda_1 - \mu_1 \quad \lambda_2 \quad \lambda_3 - \mu_3 \quad \lambda_4]^T, \end{aligned}$$

nous obtenons pour tout $k \in \mathbb{N}$ le système suivant :

$$\begin{cases} x_{2k+1} = \max(x_{2k} + b_1 \delta_{2k}, 0), & \text{si } T_i \text{ est rouge,} \\ x_{2k+2} = \max(x_{2k+1} + b_2 \delta_{2k+1}, 0), & \text{si } T_i \text{ est vert.} \end{cases}$$

1.3.2.B Critères d'optimisation

Les critères utilisés dans [De Schutter et De Moor, 1997] pour définir le problème d'optimisation sont, sur la période délimitée par les instants de commutation t_0 et t_N :

$$\begin{aligned} J_1 &= \sum_{i=1}^4 \frac{w_i}{t_N - t_0} \int_{t_0}^{t_N} l_i(t) dt, \\ J_2 &= \max_{i \in [1,4]} \left(\frac{w_i}{t_N - t_0} \int_{t_0}^{t_N} l_i(t) dt \right), \\ J_3 &= \max_{i \in [1,4]; t \in [t_0, t_N]} (w_i l_i(t)), \\ J_4 &= \sum_{i=1}^4 \frac{w_i}{\lambda_i(t_N - t_0)} \int_{t_0}^{t_N} l_i(t) dt, \\ J_5 &= \max_{i \in [1,4]} \left(w_i \frac{1}{\lambda_i(t_N - t_0)} \int_{t_0}^{t_N} t_N l(i) dt \right), \end{aligned}$$

où J_1, J_2, J_3, J_4 et J_5 représentent respectivement la moyenne des longueurs de files d'attente, la longueur moyenne de la plus grande file d'attente, la plus grande longueur instantanée parmi les files d'attente, le temps d'attente moyen de toutes les files d'attente et le plus grand temps moyen d'attente. Ces critères sont tous pondérés par un facteur $w_i > 0, i \in [1, 4]$, dont le choix dépend de l'importance de chaque ligne au sein de l'intersection.

1.3.2.C Stratégie de commande

Des contraintes sur la commande impliquent que les périodes de feu rouge δ_{2k} et de feu vert δ_{2k+1} ne doivent pas dépasser certaines bornes fixées préalablement par l'ingénieur. Une contrainte supplémentaire sur l'état traduit d'une part le caractère positif des files d'attente et exclut d'autre part l'existence de trajectoires de longueur infinie.

La stratégie d'optimisation consiste à calculer, pour un horizon N donné, une séquence de commutations optimales t_0, t_1, \dots, t_N telle que l'un des critères $J_i, i \in [1, 5]$ prenne une valeur minimale sous les contraintes.

Pour atteindre cet objectif, les auteurs ont formalisé le problème de l'optimisation sous la forme d'un **PCLE**. Ce problème est défini comme suit :

« Soient les matrices $A \in \mathbb{R}^{p \times n}$ et $B \in \mathbb{R}^{q \times n}$, les vecteurs $c \in \mathbb{R}^p$ et $d \in \mathbb{R}^q$ et m sous-ensembles $\phi_1, \phi_2, \dots, \phi_m$ de $[1, 2, \dots, p]$. Trouver un vecteur $x \in \mathbb{R}^n$ tel que :

$$\sum_{j=1}^m \prod_{i \in \phi_j} (Ax - c)_i = 0 \quad ,$$

sous la contrainte $Ax \geq c$ et $Bx = d$, ou bien montrer qu'aucun vecteur x n'existe. »

En résumé, le problème de commande revient à minimiser $J_i, i \in [1, 5]$ sous les contraintes suivantes :

$$\begin{aligned} AX + B\Delta + c &\geq 0 \quad , \\ X &\geq 0 \quad , \\ EX + D\Delta + f &\geq 0 \quad , \\ (AX + B\Delta + c)^T X &= 0 \quad , \end{aligned}$$

avec

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \quad \text{et} \quad \Delta = \begin{bmatrix} \delta_0 \\ \vdots \\ \delta_{N-1} \end{bmatrix} \quad .$$

Il est clair que ce système est un cas particulier de **PCLE**. Les auteurs ont ainsi développé un algorithme qui évoque une description paramétrique de l'ensemble des solutions du problème **PCLE** et calcule les valeurs des paramètres qui emmènent à une minimisation globale du critère choisi $J_i, i \in [1, 5]$.

1.3.2.D Commentaires et critiques

Les auteurs notent que leur démarche considère uniquement le cas où les intersections ne subissent pas une énorme intensité de flux de véhicules. Or généralement les intersections en cas de non-saturation ne posent pas de problème de congestion. Cela représente, à notre sens, un élément critiquable. De plus, dans le cas où l'horizon d'étude N est très grand, la solution proposée doit se poser autrement. En effet, il faut alors diviser le champ défini par l'horizon en plusieurs champs N_s puis appliquer la procédure présentée. Cette démarche s'appelle « multi-PCLE sous-optimal ». En pratique, le choix de cette division n'est pas évident. Cette approche pose alors un problème au niveau de l'optimisation : cela est dû à la nature de la méthode du PCLE qui est en général un problème NP -difficile.

1.3.3 Approches basées sur les Réseaux de Petri

Comme nous l'avons vu dans la sous-section précédente, la plupart des commandes de trafic récentes émergent d'analogies avec des problèmes issus de disciplines *a priori* distinctes de l'ingénierie du trafic. Cependant, l'objectif a été jusqu'à présent de simplifier les calculs afin de pouvoir fournir une commande de feux en temps réel, parfois au détriment de la précision offerte par le modèle.

Il s'agit ici des modèles qui considèrent le trafic au niveau d'un carrefour comme étant un *Système dynamique à Événements Discrets (SED)*. Les SEDs se distinguent des systèmes dynamiques continus par le fait que les changements d'état sont observés seulement en des instants discrets du temps XXX . Plus précisément, ils sont associés à des événements. Le trafic urbain peut être assimilé à cette catégorie de systèmes, puisque les véhicules sont des entités entières et indivisibles, donc discrètes par nature. En outre, le caractère événementiel des SEDs se traduit par l'arrivée et le départ des véhicules dans le système étudié, ainsi que les changements de feux.

Les SEDs peuvent être aisément modélisés par un outil graphique puissant : les *Réseaux de Petri (RdPs)* [Petri, 1962, Murata, 1989, David et Alla, 1992]. Plusieurs travaux traitent des systèmes de transport, à l'aide des RdPs. Nous présentons ici trois travaux portant sur la planification des feux par les RdPs, qui nous paraissent représentatifs :

1. le travail de [List et Cetin, 2004],
2. le travail de [Mancinelli *et al.*, 2001],
3. le travail de [Tolba, 2004].

1.3.3.A Modélisation du système

1. le modèle proposé par [List et Cetin, 2004] décrit le fonctionnement d'un mode de carrefour en faisant abstraction des rues. Ainsi le modèle se focalise sur la représentation des phases. Le modèle se veut modulaire où la modélisation d'un cycle s'effectue à travers une agrégation des différentes phases le composant. Celles-ci peuvent être ainsi escamotables ;

2. le modèle présenté dans [Mancinelli *et al.*, 2001] reproduit le fonctionnement des feux de croisement, pour le cas d'une ville régulière. Le système considéré par cette approche est constitué de carrefours identiques reliés entre eux par des rues ayant strictement les mêmes propriétés (longueur, vitesse moyenne, *etc.*). L'intérêt principal de ce travail réside en la possibilité d'écrire les équations d'état de ce RdP dans l'algèbre des dioïdes, ce qui le rend facilement exploitable. Celui-ci se distingue du premier par la modélisation des rues et des mobiles qui sont représentés par des variables réelles ;
3. le modèle RdP proposé dans [Tolba, 2004] est modulaire. Les routes sont découpées en sections de propriétés distinctes (longueur, vitesse, *etc.*) ; chaque tronçon ainsi défini est modélisé par un RdP, présenté dans [Tolba *et al.*, 2001]. Les intersections sont également modélisées par un RdP, dont la description est faite dans [Tolba *et al.*, 2003]. Comme le modèle présenté dans [Mancinelli *et al.*, 2001], le trafic est assimilé à un fluide, ce qui est une reproduction du modèle macroscopique des voies considérées.

1.3.3.B Critères d'optimisation

Les critères d'optimisation des trois approches sont sensiblement les mêmes que ceux présentés en sous-section 1.2.3.B page 30, à savoir la moyenne des longueurs de files d'attente, la longueur moyenne de la plus grande file d'attente, la plus grande longueur instantanée parmi les files d'attente, le temps d'attente moyen de toutes les files d'attente et le plus grand temps moyen d'attente.

1.3.3.C Stratégie de commande

Les stratégies de commande sont, en revanche, complètement différentes :

1. l'approche de [List et Cetin, 2004] permet d'implémenter et simuler de manière systématique *a priori* n'importe quelle stratégie de commande de feux, dont celles présentées en sous-section 1.2.3.B page 30. Un exemple de régulation avec phases escamotables est proposé ;
2. la forme particulière des équations d'état de [Mancinelli *et al.*, 2001] permet d'utiliser certains outils algébriques puissants pour réaliser la régulation du trafic. La stratégie de régulation proposée consiste en la réalisation d'ondes vertes orthogonales ;
3. chaque carrefour est optimisé localement : la durée du cycle et des phases est calculée de manière appropriée. La commande de l'ensemble du système est ensuite résumée par un automate [Tolba *et al.*, 2004].

1.3.3.D Commentaires et critiques

En résumé, nous pouvons énumérer les avantages et inconvénients de chaque approche :

1. l'approche de [List et Cetin, 2004] profite pleinement des facilités de modélisation offertes par l'outil graphique que forment les RdPs, ainsi que des possibilités de ceux-ci à pouvoir systématiquement et automatiquement être transposés en code machine prêt

à être implémenté. Cependant, elle ne permet de synthétiser que des modèles basés sur des phases de feux. De plus, la description précise de toutes les possibilités de phases et de séquences augmente la taille du modèle de manière exponentielle lorsque le domaine d'étude considéré est élargi. Le fait de modéliser les phases en lui-même amène à une commande rigide et une réalisation du modèle fastidieuse ;

2. Le modèle de [Mancinelli *et al.*, 2001], de par sa nature algébrique, ouvre des perspectives prometteuses. Ses capacités à modéliser des villes régulières de type américain lui confère de nombreuses opportunités d'application. En revanche, il souffre des restrictions suivantes :
 - il n'est valide que lorsqu'il n'y a pas de congestion ;
 - il ne permet de modéliser que des villes très régulières, chaque rue étant de longueur identique ;
 - les flux de voitures sont supposés fixes à chaque intersection.

Cette piste est explorée en profondeur dans de récents travaux et un modèle algébrique plus formel émerge d'une importante contribution [Farhi *et al.*, 2005, Farhi *et al.*, 2007]. Nous comparons les résultats obtenus par cette nouvelle approche avec ceux produits par notre démarche dans le chapitre 3 ;

3. la distinction entre modèle et commande est bien réalisée. Les simulations du modèle montrent de grandes similitudes avec des mesures effectuées dans le cadre de cas réels et la commande proposée contribue à l'amélioration du trafic. Cependant, ce travail ne s'affranchit pas de la rigueur des séquences de phases prédéterminées. De plus, comme nous l'avons vu, le choix d'une approche macroscopique n'est pas adapté à la commande en temps réel d'une intersection isolée ni même de quelques intersections contiguës.

Conclusion

Le problème de la régulation du trafic est tout à fait contemporain et reste ouvert de nos jours. En effet, les infrastructures urbaines arrivent à leurs limites de capacité et la maîtrise des flux est la clé de voûte pour garantir un écoulement fluide et une sécurité accrue, résultant en une diminution des pollutions et des temps de trajet. Les feux de circulation, conçus au départ pour régler les conflits inhérents aux intersections, forment désormais l'un des principaux instruments de la régulation pour résoudre les problèmes contemporains liés au trafic urbain.

Les premiers modèles de trafic sont apparus avec pour objectif de mieux cerner les mécanismes régissant le trafic urbain. Les approches microscopiques permettent l'étude des interactions entre véhicules quand les méthodes macroscopiques appréhendent le système dans son ensemble. Cependant, ces modèles ne permettent pas aujourd'hui à eux seuls de palier les problèmes contemporains du trafic.

Ainsi, depuis peu, de nouvelles approches originales émergent de la littérature. Ces modèles ne sont pas des extensions ou des améliorations des approches classiques mais sont plutôt basées sur des analogies avec des problèmes *a priori* distincts de l'ingénierie du trafic. Ces modèles sont validés par des simulations et par des comparaisons avec des mesures sur le terrain. Cependant, du fait que leur conception est récente, ils présentent encore quelques limitations.

Plus précisément, ils ne prennent pas en considération les arrivées de véhicules de manière individuelle et précise, ce qui émerge pourtant grâce aux nouvelles technologies de communication et de géopositionnement.

La réputation des **RdPs** dans le domaine de la régulation et du pilotage n'est plus à faire, pas plus que la démonstration de l'analogie de ces systèmes avec les problèmes de modélisation du trafic. Comme nous l'avons vu, plusieurs modèles basés sur ce puissant outil graphique ont été validés. Cependant, quelques limitations demeurent, ce qui laisse encore un champ d'investigation large dans l'exploration de cette piste.

Dans le chapitre suivant, nous traitons de la modélisation du trafic à l'aide des **RdPs** discrets et temporisés. Nous tirons pleinement avantage du fait que le comportement d'une classe de **RdPs** peut être décrit par des équations linéaires dans une algèbre particulière, l'algèbre des dioïdes.

Chapitre 2

Étude d'un tronçon routier

2.1 Outils algébriques et applications	45
2.1.1 Dioïdes	46
2.1.2 Mise en équation des GETs	49
2.1.3 Bornes des temps de séjour et des marquages	53
2.2 Étude d'une route	58
2.2.1 Modélisation	58
2.2.2 Évaluation des performances	61
2.2.3 Validation	63
2.3 Régulation du trafic	67

Introduction

HISTORIQUEMENT, les modèles macroscopiques ont été conçus pour avoir une vue d'ensemble du trafic urbain. Les nombreuses statistiques qu'ils fournissent permettent de superviser de grands réseaux routiers et autoroutiers. Néanmoins, ils sont inadaptés à l'étude d'un système de dimension plus modeste, tel qu'un carrefour isolé, ou même quelques intersections mitoyennes. Les modèles microscopiques sont, au contraire, proposés pour prévoir les moindres variations dans les trajectoires individuelles des véhicules. En revanche, le trop grand nombre de données prises en considération alourdit la procédure de régulation sur l'échelle d'un carrefour.

Depuis quelques années, plusieurs nouvelles méthodologies de modélisation et de régulation du trafic émergent de la littérature. Parmi ces outils, les **RdPs** nous sont apparus comme particulièrement intéressants. D'un point de vue modélisation, cet outil graphique apporte de grandes facilités. De plus, les modèles **RdP** du trafic proposés dans la littérature sont accompagnés de méthodes de régulation.

L'un des atouts principaux de l'utilisation des **RdPs** est la formulation mathématique du système. Plusieurs approches analytiques peuvent être rattachées aux **RdPs**, parmi lesquelles nous trouvons l'algèbre des dioïdes. L'application de cette algèbre à la régulation peut s'avérer très intéressante car celle-ci permet de regrouper les événements du trafic à la fois dans les domaines temporels et événementiels. Plus précisément, nous obtenons non seulement le nombre de passages instantané des véhicules, mesurant ainsi le débit, mais aussi la date de passage de chaque véhicule, ce qui nous permet de calculer des temps d'attente individuels.

Cependant avant d'appliquer le modèle à la régulation, il nous incombe de le valider à travers la modélisation d'une route. En effet, pour proposer une régulation adéquate, il est important de bien connaître le comportement des véhicules au sein des zones de stockage situées en amont de la zone de conflit. À la lumière de ces remarques, nous avons décidé de concevoir un modèle mesoscopique, permettant de représenter l'évolution du trafic, basé sur l'algèbre des dioïdes.

Ce chapitre débute par l'introduction des notions de base de cette algèbre qui servent pour la modélisation. Ainsi, nous faisons le lien entre l'algèbre des dioïdes et le comportement d'une classe de **RdPs**. La seconde partie présente un modèle de route et décrit les différentes mesures du trafic que le modèle est capable de fournir. Enfin, la dernière section montre les résultats issus de certains outils de régulation de l'algèbre des dioïdes.

2.1 Outils algébriques et applications

Dans cette section, nous présentons les outils mathématiques nécessaires à l'étude algébrique d'une classe de **RdPs**. Nous rappelons également la manière avec laquelle ces outils algébriques sont utilisés. Plusieurs propriétés intéressantes découlent de leur combinaison avec les **RdPs**. Elles sont particulièrement utiles pour réaliser une analyse, ainsi qu'une maîtrise du système à travers son modèle graphique ou algébrique.

2.1.1 Dioïdes

Avant d'aborder la mise en équation des RdPs, nous présentons les fondements de base de l'algèbre des dioïdes en introduisant les définitions et les propriétés les plus importantes à retenir. Pour une étude plus approfondie, nous recommandons au lecteur les ouvrages [Cunninghame-Green, 1979, Baccelli *et al.*, 1992].

Rappelons quelques notions de la théorie des ensembles. Un ensemble E est dit *ordonné* s'il existe une relation binaire \preceq telle que, pour tout $x, y, z \in E$, cette relation est :

- *réflexive* : tout élément est en relation avec lui-même ($x \preceq x$);
- *antisymétrique* : si $x \preceq y$ et $y \preceq x$, alors $x = y$;
- *transitive* : si $x \preceq y$ et $y \preceq z$, alors $x \preceq z$.

Un ensemble E est *fermé* pour une opération *ssi* pour tout $a, b \in E$, le résultat de cette opération entre a et b est inclus dans E .

Soit (A, \preceq) un ensemble ordonné, $B \subset A$ un sous-ensemble non vide de A et $a, b \in B$:

- un élément $x \in A$ satisfaisant $\forall a \in B, x \preceq a$ est appelé *minorant* de B ;
- un élément $y \in A$ satisfaisant $\forall b \in B, y \succeq b$ est appelé *majorant* de B .

En particulier, si la *borne supérieure* (i.e. le plus petit majorant) et/ou la *borne inférieure* (i.e. le plus grand minorant) de l'ensemble $[a, b]$ existent, elles sont notées $a \vee b$ et $a \wedge b$ respectivement.

Soient (A, \preceq_A) et (B, \preceq_B) des ensembles ordonnés et soit f une application définie par $f : A \rightarrow B$.

- f est *injective* si $\forall a, b \in A, f(a) = f(b) \iff a = b$;
- f est *surjective* si $\forall b \in B, \exists a \in A \mid f(a) = b$;
- f est *bijective* si elle est à la fois injective et surjective ;
- f est *isotone* si $\forall a, b \in A, a \preceq_A b \implies f(a) \preceq_B f(b)$.

Parmi les ensembles ordonnés, il est une classe particulièrement riche de propriétés, qui permet d'établir un lien entre les ensembles ordonnés et les structures algébriques. Cette classe est introduite par la définition suivante.

Soit (E, \preceq) un ensemble ordonné non vide.

- Si $x \wedge y$ existe dans E pour tout couple $(x, y) \in E$, alors (E, \preceq) est un *demi-treillis inférieur*;
- Si $x \vee y$ existe dans E pour tout couple $(x, y) \in E$, alors (E, \preceq) est un *demi-treillis supérieur*;
- Si (E, \preceq) est à la fois un demi-treillis inférieur et supérieur, alors c'est également un *treillis* [Birkhoff, 1967, Davey et Priestley, 1990].

Il est maintenant possible d'introduire les dioïdes. Notons que les treillis sont des ensembles ordonnés par nature. Cependant, lorsqu'ils sont munis d'opérations jouissant de certaines propriétés classiques, ils peuvent être manipulés comme des structures algébriques, ainsi que le montre la définition suivante.

Définition 2.1 (Dioïde [Cunninghame-Green, 1979]) *Un dioïde $(\mathcal{D}, \oplus, \otimes)$ est un ensemble ordonné \mathcal{D} muni de deux opérations, \oplus et \otimes , qui satisfont les axiomes suivants pour tous $a, b, c \in \mathcal{D}$:*

- (i) l'addition \oplus est associative (i.e. $(a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus b \oplus c$);
- (ii) l'addition est commutative (i.e. $a \oplus b = b \oplus a$);
- (iii) l'addition est idempotente (i.e. $a \oplus a = a$);
- (iv) la multiplication \otimes est associative;
- (v) la multiplication est distributive vis-à-vis de l'addition (i.e. $a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$);
- (vi) l'ensemble \mathcal{D} inclut un élément neutre ε tel que $a \oplus \varepsilon = a$ (ε est absorbant pour la multiplication, i.e. $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$);
- (vii) l'ensemble \mathcal{D} inclut un élément identité e tel que $a \otimes e = e \otimes a = a$.

À l'instar des simplifications d'écriture de l'algèbre classique, le symbole « \otimes » est parfois omis dans la suite de ce mémoire. De plus, la notation $(\mathcal{D}, \oplus, \otimes)$ peut être raccourcie en \mathcal{D} lorsque cela n'entraîne pas de confusion.

En corollaire à sa définition, un dioïde \mathcal{D} est un ensemble ordonné pour la relation \preceq définie pour tout $a, b \in \mathcal{D}$ par $a \preceq b \Leftrightarrow a \oplus b = b$. Autrement dit, $a \oplus b$ est la borne supérieure de l'ensemble $\{a, b\}$.

Soit un dioïde $(\mathcal{D}, \oplus, \otimes)$. L'ensemble $\mathcal{C} \subset \mathcal{D}$ muni des mêmes opérations \oplus et \otimes est un sous-dioïde de \mathcal{D} ssi $\varepsilon, e \in \mathcal{C}$ et \mathcal{C} est fermé pour \oplus et \otimes [Baccelli *et al.*, 1992, p. 156].

Un dioïde est *complet* s'il est fermé pour les sommes infinies et si l'axiome (v) de la définition 2.1 s'applique aux sommes infinies. Dans un tel dioïde, le plus grand élément, noté \top , existe et est égal à la \oplus -somme de tous les éléments de \mathcal{D} .

Un dioïde est *commutatif* si la multiplication est commutative. Par exemple, le dioïde complet et commutatif $\overline{\mathbb{Z}}_{\min}$, communément appelé *algèbre* $(\min, +)$, est l'ensemble $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, +\infty\}$ doté des opérations suivantes :

$$\begin{aligned} a \oplus b &\equiv \min(a, b) \quad , \\ a \otimes b &\equiv a + b \quad , \end{aligned}$$

pour les « scalaires » $a, b \in \overline{\mathbb{Z}}$ et

$$\begin{aligned} [A \oplus B]_{ij} &= [A]_{ij} \oplus [B]_{ij} \equiv \min([A]_{ij}, [B]_{ij}) \\ [A \otimes C]_{ij} &= \bigoplus_{k=1}^n [A]_{ik} \otimes [C]_{kj} \equiv \min_{k=1, \dots, n} ([A]_{ik} + [C]_{kj}) \quad , \end{aligned}$$

pour les matrices $A, B \in \overline{\mathbb{Z}}^{m \times n}$ et $C \in \overline{\mathbb{Z}}^{n \times p}$. Dans ce dioïde, $\varepsilon = +\infty$, $e = 0$ et $\top = -\infty$.

Le dioïde $\overline{\mathbb{Z}}_{\max}$, également dénommé *algèbre* $(\max, +)$, est obtenu en remplaçant \min par \max dans la définition de $\overline{\mathbb{Z}}_{\min}$. Dans ce dioïde, $\varepsilon = -\infty$, $e = 0$ et $\top = +\infty$.

La \otimes -multiplication dans un dioïde est rarement inversible [Blyth et Janowitz, 1972]. Si l'application f n'est pas surjective, $f(x) = b$ peut n'avoir aucune solution; si elle n'est pas injective, cette équation peut avoir plusieurs solutions. La *théorie de la résiduation* apporte une réponse à ce problème d'inversion d'application. Cette théorie permet notamment de calculer, lorsqu'elle existe, la plus grande solution à l'inégalité $f(x) \preceq b$.

Définition 2.2 (Résiduation, résiduée) Soit une application $f : A \rightarrow B$, avec (A, \preceq_A) et (B, \preceq_B) des ensembles ordonnés. f est résiduable si pour tout $y \in B$, la borne supérieure de l'ensemble $\{x \in A \mid f(x) \preceq y\}$ existe et est incluse dans cet ensemble. Cette borne est alors dénotée $f^\sharp(y)$; l'application f^\sharp est appelée la résiduée de f .

En l'occurrence, l'application $L_a : x \mapsto a \otimes x$ définie sur un dioïde complet \mathcal{D} est résiduable. Sa résiduée est l'application définie par :

$$L_a^\sharp(y) \stackrel{\text{déf}}{=} a \oslash y = \frac{y}{a} = \bigwedge \{x \in \mathcal{D} \mid a \otimes x \preceq y\} \quad .$$

De même, l'application $R_a : x \mapsto x \otimes a$ définie sur un dioïde complet \mathcal{D} est résiduable. Sa résiduée est l'application définie par :

$$R_a^\sharp(y) \stackrel{\text{déf}}{=} y \oslash a = \frac{y}{a} = \bigwedge \{x \in \mathcal{D} \mid x \otimes a \preceq y\} \quad .$$

Les propriétés suivantes, inhérentes à $\overline{\mathbb{Z}}_{\min}$, sont démontrées dans [Cohen *et al.*, 1991] :

$$\frac{x}{a} \otimes a \preceq x \quad , \quad (2.1)$$

$$\frac{x \otimes a}{a} \succeq x \quad , \quad (2.2)$$

$$x \preceq \frac{ax}{a} \quad , \quad (2.3)$$

$$\frac{x \oslash a}{b} = \frac{x}{ba} \quad , \quad (2.4)$$

$$\frac{b}{a} \equiv b - a \quad \text{si } a \text{ et } b \text{ sont finis,} \quad (2.5)$$

$$\frac{\top}{a} = \top \quad \text{pour tout } a,$$

$$\frac{\varepsilon}{a} = \varepsilon \quad \text{si } a \neq \varepsilon,$$

$$\frac{a}{\varepsilon} = \top \quad \text{pour tout } a,$$

$$\frac{a}{\top} = \varepsilon \quad \text{si } a \neq \top.$$

En particulier, nous soulignons que $\varepsilon \otimes \top = \varepsilon$ (autrement dit $+\infty - \infty = +\infty$) tandis que $\varepsilon \oslash \varepsilon = \top$ (ou encore $+\infty - \infty = -\infty$). Ceci montre que la notation $a - b$ est ambiguë pour des valeurs infinies de a et b , ce qui constitue la principale difficulté pour calculer le marquage instantané des places. En effet, rappelons que, par convention, le compteur $u(t) = \varepsilon$ ($= +\infty$) lorsque la transition u n'a pas été tirée à la date t . Autrement dit, nous ne sommes pas en mesure de calculer les marquages d'une place encadrée par les transitions u et v tant que ces transitions

ne sont pas tirées au moins une fois. Pour autant que nous sachions, les travaux publiés jusqu'à maintenant se limitent au calcul des bornes des marquages.

Dans la sous-section suivante, nous allons voir comment utiliser les dioïdes pour décrire le comportement d'une classe de **RdPs**.

2.1.2 Mise en équation des **GETs**

Comme nous allons le voir, les **GETs** forment une classe de **RdPs**.

2.1.2.A Définitions

Un *Réseau de Petri* (**RdP**) est un graphe doté d'un état initial appelé *marquage initial* et noté \mathcal{M}_0 [Petri, 1962, Murata, 1989]. Le graphe sous-jacent N d'un **RdP** est orienté, pondéré et *bipartite* puisqu'il est composé de deux sortes de nœuds, appelés *places* et *transitions*. Dans la représentation graphique, les places sont dessinées comme des cercles et les transitions comme des barres ou des rectangles. Les *arcs* (flèches) vont d'une place vers une transition ou d'une transition vers une place. Le poids d'un arc, qui est un entier, est noté au-dessus de celui-ci ; un arc de poids k peut être interprété comme k arcs parallèles et les poids de 1 sont habituellement omis.

Un *marquage* assigne à chaque place un entier positif. Si un marquage assigne un entier k à une place \mathcal{P} , alors \mathcal{P} est marquée avec k *marques* ou *jetons* : graphiquement, k points noirs (jetons) sont dessinés dans la place \mathcal{P} . Un marquage \mathcal{M} est un vecteur qui a pour dimension le nombre de places. Le \mathcal{P}^e élément de \mathcal{M} , noté $\mathcal{M}(\mathcal{P})$, est le nombre de jetons dans la place \mathcal{P} . Par exemple, dans la figure 2.1(a), la place \mathcal{P}_2 est marquée de 3 jetons ; l'arc allant de \mathcal{P}_2 à \mathcal{T}_3 a un poids de 2.



FIGURE 2.1 – Exemple de **RdP** : la transition \mathcal{T}_3 est validée et peut être franchie

En terme de modélisation, les places représentent des conditions et les transitions des événements. Une transition a un certain nombre de *places amont* et *aval* représentant respectivement les conditions préalables et induites de l'événement.

D'une manière plus formelle, un **RdP** est un quintuple $R = (P, T, F, W, \mathcal{M}_0)$ dans lequel :

- $P = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_m\}$ est un ensemble fini de places ;
- $T = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}$ est un ensemble fini de transitions ;
- $F \subseteq (P \times T) \cup (T \times P)$ est un ensemble d'arcs (relations de flux) ;
- $W : F \rightarrow \mathbb{N}^*$ est une fonction de pondération ;

- $\mathcal{M}_0 : P \rightarrow \mathbb{N}$ est le marquage initial ;
- $P \cap T = \emptyset$ et $P \cup T \neq \emptyset$.

Une structure de **RdP** sans marquage initial spécifique est noté $N = (P, T, F, W)$. Un **RdP** accompagné de son marquage initial est noté (N, \mathcal{M}_0) .

Les comportements de beaucoup de systèmes peuvent être décrits en termes d'états du système et leurs changements. Afin de simuler le comportement dynamique d'un système, un état ou marquage d'un **RdP** est changé suivant la loi suivante, dite de *tir de transition* :

1. une transition \mathcal{T} est dite *validée* si chacune de ses places amont \mathcal{P}_{in} est marquée d'au moins $w(\mathcal{P}_{in}, \mathcal{T})$ jetons, avec $w(\mathcal{P}_{in}, \mathcal{T})$ le poids de l'arc allant de \mathcal{P}_{in} à \mathcal{T} . Dans ce cas, l'événement correspondant à \mathcal{T} peut avoir lieu ;
2. une transition validée peut être (ou ne pas être) *tirée* ou *franchie*, selon que l'événement lui correspondant a lieu ou pas ;
3. un tir de la transition \mathcal{T} retire $w(\mathcal{P}_{in}, \mathcal{T})$ jetons de chaque place amont \mathcal{P}_{in} et génère $w(\mathcal{T}, \mathcal{P}_{out})$ jetons dans chaque place aval \mathcal{P}_{out} .

Par exemple, dans la figure 2.1(a), la transition \mathcal{T}_3 est validée. Son franchissement (figure 2.1(b)) consomme un jeton de la place \mathcal{P}_1 et deux de la place \mathcal{P}_2 ; il génère un jeton dans la place \mathcal{P}_3 .

Une transition sans place amont est dite *transition source* ; elle est validée sans condition. Une transition sans place aval est dite *transition puits* ; elle consomme des jetons mais n'en génère aucun.

Un **RdP** est dit *ordinaire* si tous ses arcs ont un poids de 1.

Une *séquence de tirs* ou *d'occurrences* est notée $\sigma = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}$.

Un *invariant de place* est un ensemble de places dans lequel le nombre de jetons est constant, quelque soit la séquence de tirs choisie.

Un **RdP** (N, \mathcal{M}_0) est *vivant* (ou, de manière équivalente, \mathcal{M}_0 est un *marquage vivant* pour N) si, quelque soit le marquage atteint depuis \mathcal{M}_0 , il est toujours possible de tirer n'importe quelle transition en suivant une séquence de tirs donnée. Autrement dit, un **RdP** vivant garantit qu'il n'y aura pas de blocage, quelque soit la séquence de tirs choisie. À l'inverse, une transition \mathcal{T} est dite *morte* si elle ne peut jamais être tirée, quelque soit la séquence de tirs choisie. Par extension, un **RdP** est *mort* si aucune de ses transitions ne peut jamais être tirée. Il existe de nombreux cas de **RdPs** qui ne sont ni vivants, ni morts.

Il est possible d'ajouter un aspect temporel aux **RdPs**, en associant une *temporisation* soit aux places (*\mathcal{P} -temporisé*), soit aux transitions (*\mathcal{T} -temporisé*). Dans le premier cas, les jetons doivent séjourner au moins le temps associé à la place \mathcal{P} pour valider les transitions en aval. Dans le second cas, la transition \mathcal{T} ne génère les jetons dans les places aval qu'une fois le temps associé écoulé, à compter du début du tir.

Un **RdP** temporisé dont les temporisations sont constantes et connues est dit *déterministe*. A contrario, un **RdP** *stochastique* est un **RdP** dans lequel :

- le temps de franchissement des transitions est généré par des variables aléatoires suivant une distribution exponentielle, à support dans \mathbb{N} ou \mathbb{R}^+ ;
- une politique d'exécution est associée pour gérer les conflits.

Un *Graphe Marqué (GM)* est un **RdP** ordinaire tel que chaque place a exactement une transition amont et une transition aval. Un *Graphe d'Événements Temporisés (GET)* est un **GM** temporisé. Il est possible d'écrire les équations d'état $(\min, +)$ - ou $(\max, +)$ -linéaires qui décrivent l'évolution des tirs des transitions d'un **GET** ordinaire, lorsque nous considérons un *fonctionnement au plus tôt* (i.e. toute transition est franchie aussitôt qu'elle est validée).

2.1.2.B Modèle linéaire dans un dioïde

Pour obtenir un modèle linéaire, nous pouvons associer à chaque transition x , soit une *fonction compteur* (ou simplement *compteur*), soit une *fonction dateur* (ou simplement *dateur*). Un compteur $x(\cdot)$ est une application de \mathbb{Z} dans $\overline{\mathbb{Z}}_{\min}$ tel que $x(t)$ est le nombre de tirs de la transition x à l'instant t (par convention, $x(t) = \varepsilon$ si x n'a pas encore été tirée à la date t). Un dateur $x(\cdot)$ est une application de \mathbb{Z} dans $\overline{\mathbb{Z}}_{\max}$ telle que $x(k)$ est la date du k^{e} tir de la transition x . Nous illustrons ces définitions par l'exemple suivant.

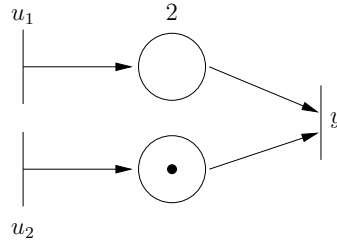


FIGURE 2.2 – Un **GET** simple

Considérons le **GET** de la figure 2.2. Le comportement de la transition y peut être décrit comme suit :

- le compteur associé est

$$\begin{aligned} y(t) &= \min(u_1(t-2), u_2(t)+1) \quad , \\ &= u_1(t-2) \oplus 1 \otimes u_2(t) \text{ dans } \overline{\mathbb{Z}}_{\min}; \end{aligned}$$

- le dateur associé est

$$\begin{aligned} y(k) &= \max(u_1(k)+2, u_2(k-1)) \quad , \\ &= 2 \otimes u_1(k) \oplus u_2(k-1) \text{ dans } \overline{\mathbb{Z}}_{\max}. \end{aligned}$$

À l'instar de la transformée en z des séries de l'algèbre classique, les *transformées en γ* et *en δ* permettent de traduire respectivement des dateurs et des compteurs en séries formelles, avec :

- γ l'*opérateur de décalage événementiel* (abusivement, $x(k-1) \xrightarrow{\gamma} \gamma x(\gamma)$);
- δ l'*opérateur de décalage temporel* (abusivement, $x(t-1) \xrightarrow{\delta} \delta x(\delta)$).

Ainsi, le comportement d'un **GET** dans son ensemble peut être décrit par des équations d'état linéaires dans un dioïde de séries formelles à deux variables commutatives γ et δ , à exposants dans \mathbb{Z} et à coefficients booléens [Cohen et al., 1989]. Ce dioïde, dénoté $\mathcal{M}_{in}^{ax}[\gamma, \delta]$,

permet d'étudier un système simultanément dans les domaines temporel et événementiel. Un élément $s \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ est défini par $s(\gamma, \delta) = \bigoplus_{i \in \mathbb{Z}} f(k_i, t_i) \gamma^{k_i} \delta^{t_i}$ avec $f : \mathbb{Z}^2 \rightarrow \mathbb{B}$ une application booléenne. Lorsqu'un élément de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ est utilisé pour représenter les informations concernant une transition d'un **GET**, chacun de ses monômes $\gamma^k \delta^t$ peut être interprété comme : « le k^{e} événement a lieu au plus tard à la date t ». Par exemple, le comportement de la transition y de la figure 2.2 est décrit par l'équation suivante dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$:

$$y(\gamma, \delta) = \delta^2 u_1(\gamma, \delta) \oplus \gamma^1 u_2(\gamma, \delta) \quad .$$

Autrement dit, pour que le k^{e} tir de y ait lieu, il faut que 2 unités de temps (**u.t.s**) se soient écoulées après le k^{e} tir de u_1 (*i.e.* $\delta^2 u_1$), mais également que le $(k - 1)^{\text{e}}$ tir de u_2 (*i.e.* $\gamma^1 u_2$) ait eu lieu.

Dans la littérature, il est d'usage de regrouper les séries décrivant les tirs des

- n transitions source dans le vecteur d'entrée U ,
- p transitions puits dans le vecteur de sortie Y ,
- m autres transitions dans le vecteur d'état X .

Ceci permet d'écrire les équations d'état des **GETs** sous la forme matricielle suivante :

$$\begin{cases} X(\gamma, \delta) = A \otimes X(\gamma, \delta) \oplus B \otimes U(\gamma, \delta) \quad , \\ Y(\gamma, \delta) = C \otimes X(\gamma, \delta) \quad . \end{cases}$$

De la même manière qu'en automatique classique, il est possible de calculer une *matrice de transfert* H depuis ces équations.

Soit l'opérateur « $*$ » appelé *étoile de Kleene* et défini par $s^* = \bigoplus_{i \in \mathbb{N}} s^i$. La matrice de transfert est alors donnée par $H = CA^*B$ [Baccelli *et al.*, 1992]. Elle représente la relation entrée(s)-sortie(s) du système dans le cadre d'un fonctionnement au plus tôt.

2.1.2.C Autres notions

La *restriction* d'une application correspond à la réduction de son domaine (*i.e.* ensemble de définition) et/ou codomaine (*i.e.* ensemble d'arrivée). Cette notion est introduite dans [Wonham, 1979] et reprise notamment dans [Lhommeau, 2003, pp. 13 à 15]. En particulier, la restriction suivante est présentée.

Définition 2.3 (Injection canonique) Soit B un ensemble et soit $A \subset B$. L'application I_A définie sur $A \rightarrow B$ par $x \mapsto x$ est appelée injection canonique de A dans B .

Définition 2.4 (Causalité) Une série $s \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$ est dite causale si $s = \varepsilon$ ou si chacun de ses monômes est à exposants dans \mathbb{N} . Une matrice est dite causale si ses éléments sont causaux.

Il est possible d'extraire la partie causale d'une série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ à l'aide de la *projection causale*, notée Pr_+ . Pratiquement, cette application ne conserve que les monômes à exposants positifs ou nuls. Formellement, cette application est définie comme suit [Cottenceau *et al.*, 2001] : soit $\mathcal{M}_{in}^{ax}[\gamma, \delta]^+ \subset \mathcal{M}_{in}^{ax}[\gamma, \delta]$ le sous-dioïde complet des séries causales de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. L'injection canonique $I : \mathcal{M}_{in}^{ax}[\gamma, \delta]^+ \rightarrow \mathcal{M}_{in}^{ax}[\gamma, \delta]$ est résiduable. Sa résiduée, notée Pr_+ , est une *projection*, c'est-à-dire : $\text{Pr}_+ \circ \text{Pr}_+ = \text{Pr}_+$.

Définitions 2.5 (Applications sur les dioïdes) Soient (A, \oplus_A, \otimes_A) et (B, \oplus_B, \otimes_B) deux dioïdes et soit $f : A \rightarrow B$ une application.

- f est un \oplus -morphisme si $\forall a, b \in A, f(a \oplus_A b) = f(a) \oplus_B f(b)$ et $f(\varepsilon_A) = \varepsilon_B$;
- f est un \otimes -morphisme si $\forall a, b \in A, f(a \otimes_A b) = f(a) \otimes_B f(b)$ et $f(\mathbf{e}_A) = \mathbf{e}_B$;
- f est un homomorphisme si elle est à la fois un \oplus -morphisme et un \otimes -morphisme ;
- f est un isomorphisme si elle est à la fois un homomorphisme et bijective.

Les séries décrivant le comportement des GETs forment un sous-dioïde de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$. En effet, la réponse impulsionnelle d'un GET à une entrée et une sortie est ultimement périodique. Les notions suivantes sont tirées de [Baccelli *et al.*, 1992, pp. 254 à 256].

Définition 2.6 (Périodicité) Une série $s \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ est dite périodique si elle peut s'écrire sous la forme $s = p \oplus qr^*$ où p et q sont des polynômes de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ et r un monôme de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$. Une matrice est dite périodique si ses éléments sont périodiques.

Définition 2.7 (Rationalité) Une série $s \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ est dite rationnelle si elle peut s'écrire avec un nombre fini d'opérations \oplus, \otimes et $*$ avec les opérands $\varepsilon, \mathbf{e}, \gamma$ et δ . Une matrice est dite rationnelle si ses éléments sont rationnels.

Définition 2.8 (Réalisation) Soient C et B des matrices de dimension $n \times m$ et $p \times n$ respectivement, ayant tous leurs éléments égaux à ε ou \mathbf{e} . Une matrice $H \in (\mathcal{M}_{in}^{ax}[[\gamma, \delta]])^{p \times m}$ est réalisable si elle peut s'écrire sous la forme $H = C(\gamma A_1 \oplus \delta A_2)^* B$, où A_1 et A_2 sont des matrices de dimension $n \times n$ telles que n un est nombre entier arbitraire mais fini (dépendant de H).

Pour toute matrice $H \in (\mathcal{M}_{in}^{ax}[[\gamma, \delta]])^{p \times m}$, les propositions suivantes sont équivalentes [Baccelli *et al.*, 1992, p. 256] :

- (i) H est réalisable,
- (ii) H est périodique et causale,
- (iii) H est rationnelle.

2.1.3 Bornes des temps de séjour et des marquages

Dans [Cohen *et al.*, 1991], les auteurs montrent comment utiliser la résiduation de séries formelles de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ pour calculer le minimum/le maximum des temps de séjour/des marquages de certains chemins d'un GET, dans le cadre d'un fonctionnement au plus tôt et pour des tirs impulsionnels d'une entrée (*i.e.* l'entrée est tirée une infinité de fois à la date $t = 0$).

2.1.3.A Théorie du second ordre

Soit une place \mathcal{P} ayant u comme transition amont et v comme transition aval. En représentant les trajectoires de tir des transitions u et v par des fonctions compteurs, nous relierons le

marquage (ou stock) de la place \mathcal{P} évalué à la date t , noté $\bar{S}_{uv}(t)$, le marquage initial $\bar{S}_{uv}(0)$, et les compteurs $u(t)$ et $v(t)$ par la relation suivante (dans l'algèbre usuelle) :

$$\underbrace{\bar{S}_{uv}(t)}_{\text{stock à la date } t} = \underbrace{\bar{S}_{uv}(0)}_{\text{stock initial}} + \underbrace{u(t)}_{\text{jetons entrés}} - \underbrace{v(t)}_{\text{jetons sortis}} .$$

De la même manière, en représentant les trajectoires de tir des transitions u et v par des fonctions dateurs, le temps de séjour du k^{e} jeton dans la place \mathcal{P} , noté $\bar{T}_{uv}(k)$, est donné par la relation suivante dans l'algèbre usuelle, si $\mathcal{M}_0(\mathcal{P}) = 0$:

$$\underbrace{\bar{T}_{uv}(k)}_{\text{temps de séjour du } k^{\text{e}} \text{ jeton}} = \underbrace{v(k)}_{\text{date de sortie}} - \underbrace{u(k)}_{\text{date d'entrée}} .$$

La variable $S_{uv}(t) = \bar{S}_{uv}(t) - \bar{S}_{uv}(0)$ décrit alors l'évolution du stock de la place (ou des chemins) située entre u et v de la date 0 à la date t .

Soit Σ_c l'ensemble des fonctions compteurs $\mathbb{Z} \rightarrow \bar{\mathbb{Z}}_{\min}$, c'est-à-dire l'ensemble des applications non décroissantes de \mathbb{Z} dans $\bar{\mathbb{Z}}_{\min}$, que nous munissons du min comme \oplus -addition et du produit de convolution \star comme \otimes -produit, défini par :

$$u(\cdot), v(\cdot) \in \Sigma_c, (u \star v)(t) \stackrel{\text{déf}}{=} \bigoplus_{\tau \in \mathbb{Z}} u(t - \tau) \otimes v(\tau) .$$

$(\Sigma_c, \oplus, \star)$ a alors une structure de dioïde complet.

Puisque Σ_c est complet, le produit de convolution est par conséquent résiduable [Baccelli *et al.*, 1992, p. 172]. L'expression de la résiduée du produit de Σ_c est la suivante, avec $u(\cdot)$ et $v(\cdot)$ deux applications de Σ_c :

$$\begin{aligned} (u \oslash v)(t) &\stackrel{\text{déf}}{=} \bigoplus \{c(t) \in \Sigma_c \mid (c \star v)(t) \preceq u(t)\} , \\ &= \bigwedge_{\tau \in \mathbb{Z}} (u(\tau) - v(\tau - t)) \quad \left(= \max_{\tau \in \mathbb{Z}} [u(\tau) - v(\tau - t)] \right) . \end{aligned} \quad (2.6)$$

De même, en représentation dateur, le produit de convolution de deux fonctions dateurs $u(\cdot)$ et $v(\cdot)$ de Σ_d (dioïde des fonctions dateurs) est :

$$\begin{aligned} y(k) &= (u \star v)(k) , \\ &\stackrel{\text{déf}}{=} \bigoplus_{\nu \in \mathbb{Z}} u(k - \nu) \otimes v(\nu) \quad \left(= \max_{\nu \in \mathbb{Z}} [u(k - \nu) + v(\nu)] \right) , \end{aligned}$$

et la résiduée du produit de convolution de deux dateurs $u(\cdot), v(\cdot) \in \Sigma_d$ est donnée par :

$$\begin{aligned} (u \oslash v)(k) &\stackrel{\text{déf}}{=} \bigoplus \{d(k) \in \Sigma_d \mid (d \star v)(k) \preceq u(k)\} , \\ &= \bigwedge_{\nu \in \mathbb{Z}} (u(\nu) - v(\nu - k)) \quad \left(= \min_{\nu \in \mathbb{Z}} [u(\nu) - v(\nu - k)] \right) . \end{aligned} \quad (2.7)$$

Ces expressions de la résiduée du produit de convolution conduisent aux résultats suivants [Cohen *et al.*, 1991, théorème 4.2] :

1. soient $u(\cdot), v(\cdot) \in \Sigma_c$ deux fonctions compteurs. Alors,

$$\forall t, -(v \wp u)(0) \leq S_{uv}(t) \leq (u \wp v)(0) \quad ; \quad (2.8)$$

2. soient $u(\cdot), v(\cdot) \in \Sigma_d$ deux fonctions dateurs. Alors,

$$\forall k, (v \wp u)(0) \leq \bar{T}_{uv}(k) \leq -(u \wp v)(0) \quad . \quad (2.9)$$

Remarque 2.1 *Il faut noter qu'il existe $t \in \mathbb{Z}$ tel que $S_{uv}(t) = (u \wp v)(0)$. C'est-à-dire que pour deux compteurs $u(t)$ et $v(t)$, la borne du stock fournie par le calcul de $(u \wp v)(0)$ est atteinte pour une date t . De même pour l'autre borne.*

En codant les fonctions dateur ou compteur par des séries formelles de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$, le comportement entrée-sortie d'un GET, qui apparaît comme un produit de convolution sur Σ_c ou Σ_d , s'exprime alors comme un simple produit de séries formelles dans $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$. Les dioides Σ_c , Σ_d et $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ sont isomorphes. Par exemple, l'isomorphisme entre Σ_c et $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ s'exprime de la manière suivante¹.

Soit s une série de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$. $\mathcal{C}_s(t)$ dénote la fonction compteur de Σ_c , définie par :

$$s = \bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{C}_s(t)} \delta^t \quad .$$

L'application $s \mapsto \mathcal{C}_s$ définit un isomorphisme de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ dans Σ_c . Notamment, pour $u, v \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$:

$$(\mathcal{C}_u \wp \mathcal{C}_v)(t) = \mathcal{C}_{(u \wp v)}(t) \quad ,$$

où \wp désigne la résiduée du produit de Σ_c dans le membre de gauche et la résiduée du produit de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ dans le membre de droite.

De même, en notant l'application $\mathcal{D}_s(k)$ la fonction dateur de Σ_d définie par :

$$s = \bigoplus_{k \in \mathbb{Z}} \gamma^k \delta^{\mathcal{D}_s(k)} \quad ,$$

alors, $s \mapsto \mathcal{D}_s$ définit un isomorphisme de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ dans Σ_d . Notamment, pour $u, v \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$:

$$(\mathcal{D}_u \wp \mathcal{D}_v)(k) = \mathcal{D}_{(u \wp v)}(k) \quad .$$

¹pour plus de détails concernant cet isomorphisme de Σ_c , Σ_d et $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$, nous renvoyons le lecteur à [Gaubert, 1992, p. 185].

Les résultats (2.8) et (2.9) se réécrivent donc de la manière suivante :

$$\begin{aligned} \forall t, -\mathcal{C}_{(v\emptyset u)}(0) &\leq S_{\mathcal{C}_v \mathcal{C}_u}(t) \leq \mathcal{C}_{(u\emptyset v)}(0) \quad , \\ \forall k, \mathcal{D}_{(v\emptyset u)}(0) &\leq \overline{T}_{\mathcal{D}_v \mathcal{D}_u}(k) \leq -\mathcal{D}_{(u\emptyset v)}(0) \quad . \end{aligned}$$

Pratiquement, qu'il s'agisse d'une seule place entre u et v ou bien d'un chemin du GET, nous avons :

- l'évaluation pour $k = 0$ de la fonction dateur associée à la série $v\emptyset u$ donne une borne inférieure aux temps de séjour possibles entre les transitions u et v ,
- l'évaluation pour $k = 0$ de la fonction dateur associée à la série $u\emptyset v$ donne une borne supérieure aux temps de séjour possibles entre les transitions u et v ,
- l'évaluation pour $t = 0$ de la fonction compteur associée à la série $v\emptyset u$ donne une borne inférieure aux marquages possibles entre les transitions u et v ,
- l'évaluation pour $t = 0$ de la fonction compteur associée à la série $u\emptyset v$ donne une borne supérieure aux marquages possibles entre les transitions u et v .

À ce niveau, nous pouvons supposer que les bornes peuvent changer avec la série u de la transition amont. Cependant, comme il est expliqué dans la suite, cette supposition n'est pas fondée.

2.1.3.B Abstraction du vecteur d'entrées et interprétation

Nous présentons ici une extension de la théorie du second ordre introduite par [Cohen *et al.*, 1991]. Cette extension est détaillée plus formellement dans [Corréia, 2004, Lahaye *et al.*, 2004, Corréia *et al.*, 2006c].

L'évaluation des bornes de temps de séjour des jetons dans un chemin d'un GET à une entrée et une sortie se fait grâce aux résiduées des multiplications à droite des deux séries représentant les trajectoires de tirs de la transition d'entrée et de la transition de sortie du chemin.

Nous avons vu en effet que l'évaluation de l'exposant de δ à l'événement 0 des deux trajectoires représentées par les résiduées de ces séries nous donnent les bornes inférieure et supérieure du temps de séjour entre lesdites transitions.

Soit un GET avec une transition d'entrée u et deux transitions internes x_α et x_ω représentant respectivement l'entrée et la sortie du chemin dont nous voulons évaluer les bornes de temps de séjour des jetons. La théorie du second ordre appliquée à notre cas suggère d'évaluer les monômes $\gamma^0 \delta^i$ des résiduées

$$\frac{x_\alpha u}{x_\omega u} \quad \text{et} \quad \frac{x_\omega u}{x_\alpha u} \quad . \quad (2.10)$$

La valeur $-i$ de l'expression de gauche et la valeur i de celle de droite nous donnent respectivement les bornes supérieure et inférieure du temps de séjour des jetons entre les transitions x_α et x_ω .

Soient \mathcal{D} un dioïde complet et $a, b, c \in \mathcal{D}$. Alors d'après les propriétés (2.3) et (2.4) page 48,

$$\forall c, \frac{ac}{bc} = \frac{(ac)\emptyset c}{b} \succeq \frac{a}{b} \quad .$$

Pratiquement, la constatation précédente préconise d'évaluer les séries

$$\frac{x_\alpha}{x_\omega} \quad \text{et} \quad \frac{x_\omega}{x_\alpha}$$

pour définir les bornes de temps de séjour des jetons entre les transitions x_α et x_ω qui ne font pas partie du marquage initial. En effet, ces séries étant inférieures respectivement aux séries (2.10),

- $x_\alpha \oslash x_\omega \preceq x_\alpha u \oslash x_\omega u$ signifie que pour tout u , la première série minore la seconde. Notamment pour le monôme $\gamma^0 \delta^i$ de chaque série, i est inférieur dans la série $x_\alpha \oslash x_\omega$. Autrement dit, le temps de séjour maximum, donné par $-i$, est supérieur si nous évaluons la première série plutôt que la deuxième.
- De même, $x_\omega \oslash x_\alpha \preceq x_\omega u \oslash x_\alpha u$ signifie que pour tout u , la première série minore la seconde. Notamment pour le monôme $\gamma^0 \delta^i$ de chaque série, i est inférieur dans la série $x_\omega \oslash x_\alpha$. Autrement dit, le temps de séjour minimum, donné par i , est inférieur si nous évaluons la première série plutôt que la deuxième.

2.1.3.C Expression d'une contrainte de temps

Soient deux transitions internes x_α et x_ω d'un GET, entre lesquelles existe un chemin marqué initialement de κ jetons. L'objectif est de limiter le temps de séjour des jetons qui ne font pas partie du marquage initial dans ce chemin [Lahaye *et al.*, 2004, Corrêia *et al.*, 2006c].

Soit le temps de séjour τ des jetons dans les places entre x_α et x_ω , à ne pas dépasser. Alors le monôme de $x_\alpha \oslash x_\omega$ en l'événement 0 doit être tel que $\gamma^0 \delta^i \succeq \gamma^{-\kappa} \delta^{-\tau}$. Nous rappelons que le temps de séjour maximum entre x_α et x_ω est donné par $-i$.

x_α et x_ω sont des séries représentant les tirs de transitions, donc $x_\alpha, x_\omega \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$. Ainsi, $x_\alpha \oslash x_\omega$ est, par définition de la résiduation de R_a , une série de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, elle est donc monotone croissante.

La transition x_ω est en aval de la transition x_α , donc $x_\omega \succeq x_\alpha$. De plus, les séries x_α et x_ω représentent des trajectoires de tir d'un GET, donc elles n'ont pas de monôme dont l'exposant de γ serait négatif. En conséquence, il en est de même pour $x_\alpha \oslash x_\omega$.

En résumé, $\gamma^{-\kappa} \delta^{-\tau} \preceq \gamma^0 \delta^i$ pour le monôme $\gamma^0 \delta^i$ de $x_\alpha \oslash x_\omega$ ssi $\gamma^{-\kappa} \delta^{-\tau} \preceq x_\alpha \oslash x_\omega$. Ou encore, $\gamma^{-\kappa} \delta^{-\tau} \in \{d \mid d \otimes x_\omega \preceq x_\alpha\}$, autrement dit $\gamma^{-\kappa} \delta^{-\tau} x_\omega \preceq x_\alpha$.

Concrètement, lorsque la contrainte τ de temps de séjour des jetons qui ne font pas partie du marquage initial dans le chemin $x_\alpha \rightarrow x_\omega$, dont le marquage initial vaut κ , est respectée, les séries associées aux dateurs des transitions d'entrée et de sortie du chemin sont telles que

$$\gamma^{-\kappa} \delta^{-\tau} x_\omega \preceq x_\alpha \iff x_\omega \preceq \gamma^\kappa \delta^\tau x_\alpha \quad .$$

2.2 Étude d'une route

Afin de déterminer la meilleure gestion des feux de circulation en temps réel, nous devons avant tout convenir d'un modèle exploitable. Nous proposons ainsi une approche modulaire, en distinguant les rues de leurs intersections. Ces modèles pourront par la suite être associés par concaténation, afin de modéliser le plus grand nombre de cas. Nous présentons dans cette section notre nouvelle méthode de modélisation des rues, ainsi que les outils d'interprétation et d'analyse du trafic qui en découlent. Ces résultats nouveaux, ainsi que les propositions associées, sont développés dans [Corréia *et al.*, 2005].

2.2.1 Modélisation

Le système étudié dans ce chapitre est une rue découpée en sections dont les voies parallèles ont des propriétés identiques. À la lumière des travaux de [Tolba, 2004], nous supposons que les véhicules sont distribués de façon homogène entre les voies d'une section et que le premier véhicule entré dans chaque section est aussi le premier véhicule à la quitter. Enfin, les véhicules sont supposés de la même longueur : plus précisément notre modèle compte les véhicules en quantités entières d'**u.v.p.s.**

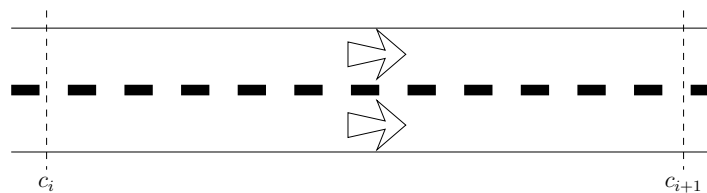


FIGURE 2.3 – Une section de rue.

Afin de modéliser une rue complète par un **GET**, chacune de ses directions est traitée séparément. Les rues sont discrétisées en segments dont les variables caractéristiques du flux dépendent seulement du temps et non pas de la position des véhicules. Chaque segment est considéré comme un système de trafic élémentaire délimité par un couple de lignes noté $[c_i, c_{i+1}]$. En effet, comme il sera montré dans la suite, une fois le modèle d'un segment défini, il est possible de l'étendre par concaténation, afin de traiter une rue complète.

Nous prenons en considération les paramètres suivants, qui restent constants dans chaque segment :

- la longueur du segment,
- la vitesse moyenne des véhicules dans le segment,
- le temps minimum nécessaire pour traverser le segment (calculé à partir des deux paramètres précédents),
- la densité maximum,
- le débit maximum ($0.5veh.s^{-1}$, en accord avec la littérature [Cohen, 1993]),
- le nombre de voies,
- le nombre initial de véhicules.

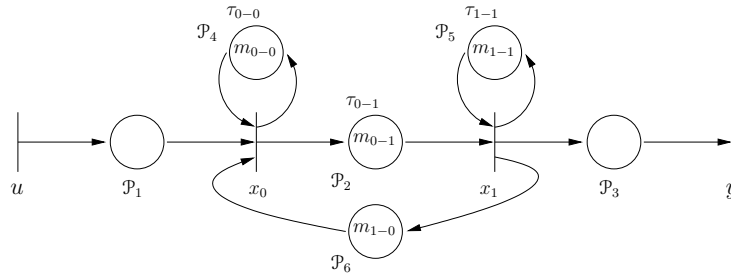


FIGURE 2.4 – Modèle GET d'une section de rue.

Le comportement du système est modélisé par le GET de la figure 2.4. Nous considérons ici un GET ordinaire et \mathcal{P} -temporisé fonctionnant sous la règle « premier entré, premier sorti » (voir [Murata, 1989]). Ce modèle se compose en deux parties. La première partie représente l'évolution des véhicules dans le segment. Le marquage de la place \mathcal{P}_2 correspond au nombre de véhicules présents dans le segment étudié. Les marquages des places \mathcal{P}_1 et \mathcal{P}_3 représentent respectivement les véhicules en amont et en aval de ce même segment. La deuxième partie caractérise le segment par sa capacité et son taux d'occupation admissible. Les marquages des places \mathcal{P}_4 et \mathcal{P}_5 comptent respectivement les voies au début (c_0) et à la fin (c_1) du segment. Puisqu'un segment ne peut contenir qu'un nombre fini de véhicules, le marquage de la place \mathcal{P}_6 limite celui de la place \mathcal{P}_2 . La signification des paramètres est détaillée dans le tableau 2.1. Les transitions rattachent les deux parties. Leurs franchissements reflètent les arrivées (u et x_0) et les départs (x_1 et y) des véhicules.

Paramètre	Signification
$\tau_{i-(i+1)}$	Temps minimum nécessaire pour traverser le segment
τ_{i-i}	Temps inter-véhiculaire minimum
m_{i-i}	Nombre de voies
$m_{i-(i+1)}$	Nombre initial de véhicules dans le segment
$m_{i-(i+1)} + m_{(i+1)-i}$	Nombre maximum de véhicules que peut contenir le segment

TABLEAU 2.1 – Paramètres du modèle d'une section de rue.

Le tableau 2.2 donne plus de détails sur le calcul de ces paramètres.

Paramètre	Calcul	
Capacité maximum	longueur \times densité maximum \times nombre de voies	
$\tau_{i-(i+1)}$	longueur	vitesse
$m_{i-(i+1)}$	nombre de véhicules initialement présents	
$m_{(i+1)-i}$	capacité maximum $- m_{i-(i+1)}$	
τ_{i-i}	1 / débit maximum	

TABLEAU 2.2 – Calcul des paramètres du modèle d'une section de rue.

Il est important de comprendre que la simulation doit avoir une date de début, dénotée $t_0 = 0$. Cependant, avant t_0 , il est impossible d'étudier le comportement du système. Ceci conduit à supposer que les jetons du marquage initial sont disponibles depuis un temps infini, ce qui correspond à des conditions initiales canoniques. Une explication formelle est discutée dans [Baccelli *et al.*, 1992, §2.6.3.4 p.95].

Le principe de l'analyse d'un **GET** est tel que les transitions validées avant t_0 sont tirées, jusqu'à obtenir un état de stabilité (aucune transition validée) qui est considéré comme étant l'état initial à t_0 . Cette convention est motivée par les raisons suivantes :

- puisque le **GET** est étudié dans le cadre d'un fonctionnement au plus tôt, une transition validée est aussitôt franchie ;
- les jetons du marquage initial sont considérés comme étant disponibles depuis un temps infini, car la causalité est nécessaire à l'étude du comportement des **GETs** à l'aide des dioïdes.

C'est pourquoi le modèle de la figure 2.4 ne permet pas de considérer un nombre initial (*i.e.* à t_0) de véhicules non nul et doit donc être complété. Par exemple, si nous prenons $m_{0-1} = 5$ dans le **GET** de la figure 2.4, ces cinq jetons sont disponibles depuis $t = -\infty$ et la transition x_1 est validée et tirée cinq fois avant la date t_0 . En conséquence de quoi, la place \mathcal{P}_2 est vide à la date $t_0 = 0$. Autrement dit le modèle ne reflète pas les conditions initiales du système.

Néanmoins, d'un point de vue pratique, nous pouvons avoir besoin d'un marquage initial non nul, dans le but de modéliser une rue non vide à la date de début de simulation t_0 . Pour pallier cet inconvénient, une entrée virtuelle *init* est connectée à chaque sortie de segment, comme le décrit la figure 2.5. Cette entrée virtuelle empêche x_1 d'être tirée avant la date t_0 et « gèle » le marquage initial jusqu'à cette date [Baccelli *et al.*, 1992]. Afin d'empêcher toute interférence avec le comportement du système, cette transition est tirée un nombre infini de fois à la date t_0 .

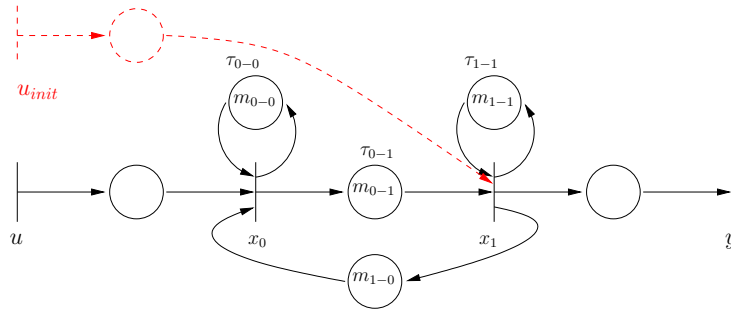


FIGURE 2.5 – Modèle **GET** d'une section de rue éventuellement non vide au début de la simulation.

Le comportement du modèle **RdP** d'un segment de route de la figure 2.5 est donné par :

$$\begin{cases} X &= \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} \gamma^{m_{0-0}} \delta^{\tau_{0-0}} & \gamma^{m_{1-0}} \\ \gamma^{m_{0-1}} \delta^{\tau_{0-1}} & \gamma^{m_{1-1}} \delta^{\tau_{1-1}} \end{pmatrix} X \oplus \begin{pmatrix} \mathbf{e} & \varepsilon \\ \varepsilon & \mathbf{e} \end{pmatrix} \begin{pmatrix} u \\ u_{init} \end{pmatrix} \\ Y &= \begin{pmatrix} \varepsilon & \mathbf{e} \end{pmatrix} X \end{cases} \quad , \quad (2.11)$$

2.2.2 Évaluation des performances

Nous pouvons retrouver, à partir du modèle **GET** du système élémentaire de trafic, les mesures de performance suivantes :

- les bornes des temps de séjour des véhicules, ce qui permet d'évaluer la capacité d'évacuation du tronçon ;
- le nombre instantané de véhicules présents dans chaque segment de rue, afin d'estimer l'état du trafic (*i.e.* fluide ou congestionné) ;
- le diagramme fondamental, pour valider notre modèle vis-à-vis de l'approche macroscopique.

2.2.2.A Bornes des temps de parcours

À partir du modèle algébrique d'un segment de rue (2.11), il est possible de calculer six bornes de temps de parcours des véhicules arrivés dans le segment à partir de $t = 0$ (*i.e.* dont les arrivées sont décrites par u).

Si nous prenons en compte un marquage initial et une séquence d'entrée u donnés, alors :

- la borne supérieure des temps de parcours des véhicules relative au marquage initial et à la séquence d'entrée u est donnée par $\mathcal{D}_{x_0 u \varnothing (\gamma^{-m_{0-1} x_1 u})}(0)$;
- la borne inférieure des temps de parcours des véhicules relative au marquage initial et à la séquence d'entrée u est donnée par $\mathcal{D}_{(\gamma^{-m_{0-1} x_1 u}) \varnothing x_0 u}(0)$.

Ainsi, pour avoir les bornes de temps de parcours quelque soit le marquage initial et la séquence d'entrée, il suffit de reprendre les formules précédentes avec des valeurs particulières de m_{0-1} , m_{1-0} et u . Plus précisément :

- la borne inférieure absolue des temps de parcours des véhicules est obtenue lorsqu'un véhicule s'engage seul dans une rue vide. Autrement dit, il faut $m_{0-1} = 0$ et $u = \gamma^0 \delta^0 \oplus \gamma^1 \delta^* = e \oplus \gamma \delta^*$; m_{1-0} doit être égal à la capacité du segment ;
- la borne supérieure absolue des temps de parcours des véhicules est obtenue lorsqu'une infinité de véhicules se présentent à l'entrée d'une rue pleine dès l'instant $t = 0$. Autrement dit, il faut $m_{1-0} = 0$ et $u = e$; m_{0-1} doit être égal à la capacité du segment.

Enfin, les formules suivantes permettent d'obtenir les bornes de temps de parcours des véhicules, avec une rue vide et quelque soit la séquence d'entrée :

- la borne supérieure des temps de parcours des véhicules arrivés à partir de $t = 0$ est donnée par $\mathcal{D}_{x_0 \varnothing (\gamma^{-m_{0-1} x_1})}(0)$;
- la borne inférieure des temps de parcours des véhicules à partir de $t = 0$ est donnée par $\mathcal{D}_{(\gamma^{-m_{0-1} x_1}) \varnothing x_0}(0)$.

2.2.2.B État du trafic

Par définition dans un **GET**, chaque place a exactement une transition amont et une transition aval, il est donc aisé de calculer les marquages instantanés à partir des séries formelles qui décrivent les séquences de tir des transitions. Pour faire face au problème des valeurs infinies des compteurs, qui a été soulevé dans la sous-section 2.1.1 page 47, nous avons besoin du produit de Hadamard [Hadamard, 1893].

Définition 2.9 (Produit de Hadamard dans un dioïde de séries formelles [Gaubert, 1995]) Soit le dioïde $\mathbb{B}[[\gamma, \delta]]$ des séries formelles en deux variables (γ, δ) , à coefficients booléens et exposants dans \mathbb{Z} , muni de l'addition et de la multiplication des séries au sens classique [Baccelli et al., 1992, p. 232].

Soit l'opérateur \odot défini comme suit :

$$a \odot b = \bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{C}_a(t) \otimes \mathcal{C}_b(t)} \delta^t ,$$

pour toutes séries $a, b \in \gamma^* \mathbb{B}[[\gamma, \delta]]$ et

$$[A \odot B]_{ij} = \bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{C}_{A_{ij}(t)} \otimes \mathcal{C}_{B_{ij}(t)}} \delta^t ,$$

pour toutes matrices $A, B \in \gamma^* \mathbb{B}[[\gamma, \delta]]^{m \times n}$ de mêmes dimensions.

Nous adoptons également la notation suivante.

Notation 2.1 Soit a une série formelle de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$. Nous définissons alors l'application suivante :

$$Q_a : \begin{cases} \mathcal{M}_{in}^{ax}[[\gamma, \delta]] & \rightarrow \gamma^* \mathbb{B}[[\gamma, \delta]] , \\ x & \mapsto x \overset{\bullet}{\oslash} a = \frac{x}{a} \bullet = \bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{C}_x(t) \oslash \mathcal{C}_a(t)} \delta^t . \end{cases}$$

La série résultant de l'opération $\overset{\bullet}{\oslash}$ appartient au dioïde $\gamma^* \mathbb{B}[[\gamma, \delta]]$, autrement dit ses monômes arrangés par exposants de δ croissants permettent de décrire une suite composée des exposants de γ correspondants qui n'est pas nécessairement monotone [Baccelli et al., 1992, p. 235]. Grâce à cet opération, nous pouvons aisément retrouver les marquages instantanés d'une place à partir des séries décrivant les tirs des transitions amont et aval de cette place, comme le montre la proposition suivante.

Proposition 2.1 (Marquages instantanés) Dans tout *GET* vivant, la série de $\gamma^* \mathbb{B}[[\gamma, \delta]]$ décrivant les marquages $\mathcal{M}_t(\mathcal{P})$ d'une place \mathcal{P} encadrée par les transitions x_{amont} et x_{aval} aux instants t est obtenue à partir des séries x_{amont} et x_{aval} de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ par :

$$\bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{M}_t(\mathcal{P})} \delta^t = \gamma^{\mathcal{M}_0(\mathcal{P})} \delta^* \odot \frac{x_{amont} \bullet}{x_{aval}} .$$

Preuve: Par définition du marquage instantané, nous avons pour tout t : $\mathcal{M}_t(\mathcal{P}) = \mathcal{M}_0(\mathcal{P}) + \mathcal{C}_{x_{amont}}(t) - \mathcal{C}_{x_{aval}}(t)$ dans l'algèbre classique. Or, il est montré dans [Cohen et al., 1991] que $a \oslash b \equiv a - b$ dans $\overline{\mathbb{Z}}_{\min}$ pour des valeurs finies de a et b . Ainsi, nous pouvons décrire le marquage instantané $\mathcal{M}_t(\mathcal{P})$ de la place \mathcal{P} à l'instant t de manière équivalente dans $\overline{\mathbb{Z}}_{\min}$ ou l'algèbre classique :

$$\mathcal{M}_t(\mathcal{P}) = \mathcal{M}_0(\mathcal{P}) + (\mathcal{C}_{x_{amont}}(t) - \mathcal{C}_{x_{aval}}(t)) \equiv \mathcal{M}_0(\mathcal{P}) \otimes (\mathcal{C}_{x_{amont}}(t) \oslash \mathcal{C}_{x_{aval}}(t)) .$$

De plus, par définition, les séries de $\gamma^* \mathbb{B}[[\gamma, \delta]]$ sont composées de monômes à exposants dans \mathbb{Z} , autrement dit ces exposants ont des valeurs finies. Nous en déduisons l'écriture de la série de $\gamma^* \mathbb{B}[[\gamma, \delta]]$ décrivant les marquages instantanés de \mathcal{P} pour tout t :

$$\begin{aligned} \bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{M}_t(\mathcal{P})} \delta^t &= \bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{M}_0(\mathcal{P}) \otimes (\mathcal{C}_{x_{amont}}(t) \oslash \mathcal{C}_{x_{aval}}(t))} \delta^t \quad , \\ &= \bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{M}_0(\mathcal{P})} \delta^t \odot \bigoplus_{t \in \mathbb{Z}} \gamma^{\mathcal{C}_{x_{amont}}(t) \oslash \mathcal{C}_{x_{aval}}(t)} \delta^t \quad , \\ &= \gamma^{\mathcal{M}_0(\mathcal{P})} \delta^* \odot \frac{x_{amont}}{x_{aval}} \bullet \quad . \end{aligned}$$

■

Grâce à la proposition 2.1, nous sommes en mesure de calculer les marquages instantanés d'une place \mathcal{P} à partir des séries de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ décrivant les tirs des transitions encadrant \mathcal{P} . Ceci nous permet d'estimer l'état du trafic à tout instant. En effet, nous pouvons interpréter le marquage instantané de la place \mathcal{P}_2 du GET de la figure 2.5 comme le nombre instantané de véhicules présents dans le segment ainsi modélisé.

2.2.2.C Diagramme fondamental

Chaque point du diagramme fondamental est obtenu en simulant le modèle d'une rue avec une valeur fixe du débit d'entrée. Autrement dit, en prenant une série périodique pour décrire la séquence de tirs de la transition u . Dans ce cas :

- le taux d'occupation moyen $\bar{\tau}$ est donnée par l'équation suivante :

$$\bar{\tau} = 100 \times \frac{\sum_i \frac{\rho_i}{\rho_{\max}} \times (t_{i+1} - t_i)}{T} \quad ,$$

avec ρ_{\max} la densité maximum du segment, ρ_i la densité instantanée dans l'intervalle de temps $t_{i+1} - t_i$ et T la durée de simulation. ρ_i est obtenu par :

$$\rho_i = \frac{\mathcal{M}_{t_i}(\mathcal{P}_2)}{l} \quad \text{à } t = t_i,$$

avec l la longueur du segment.

- le débit moyen \bar{q} est donné par le compteur de la transition de sortie évalué à la fin de la simulation $\mathcal{C}_{x_i}(T)$, divisé par la durée de simulation :

$$\bar{q} = \frac{\mathcal{C}_{x_i}(T)}{T} \quad .$$

2.2.3 Validation

Le modèle présenté dans la sous-section précédente est validé à travers la simulation d'un exemple de rue. Cette simulation doit pouvoir exprimer les deux états connus du trafic, à savoir fluide et congestionné. Considérons la rue de la figure 2.6. Nous souhaitons analyser le comportement des véhicules dans les deux segments délimités par $\{c_0, c_1\}$ et $\{c_1, c_2\}$.

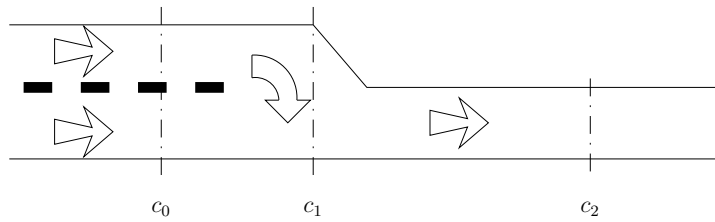


FIGURE 2.6 – Une rue découpée en sections.

Les valeurs des paramètres concernant chacun des segments sont regroupées dans le tableau 2.3.

Paramètre	Valeur	
	Segment $\{c_0, c_1\}$	Segment $\{c_1, c_2\}$
Longueur (m)	400	700
Temps minimum de traversée (s)	28.8 (50km.h^{-1})	64.0 (40km.h^{-1})
Densité maximum (veh.m^{-1})	0.15	0.15
Débit maximum (veh.s^{-1})	0.5	0.5
Nombre de voies	2	1
Véhicules présents initialement (veh)	58	41

TABLEAU 2.3 – Valeur des paramètres d'une rue découpée en sections.

Le modèle GET de cette rue est représenté par la figure 2.7.

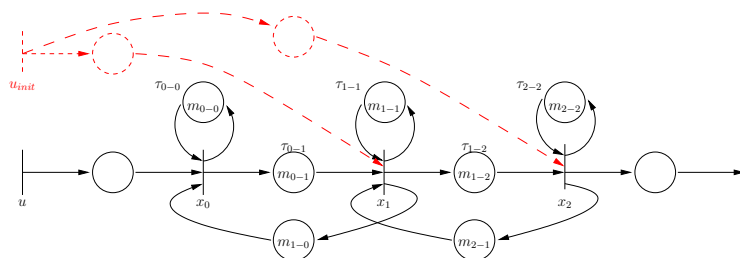


FIGURE 2.7 – Modèle modulaire d'une rue.

Les paramètres de ce modèle sont donnés par le tableau 2.4.

Paramètre	Valeur	
	Segment $\{c_0, c_1\}$	Segment $\{c_1, c_2\}$
m_{i-i}	$m_{0-0} = 2$ $m_{1-1} = m_{2-2} = 1$	
Capacité maximum	$ 400 \times 0.15 \times 2 = 120$	$ 700 \times 0.15 \times 1 = 105$
$m_{i-(i+1)}$	58	41
$m_{(i+1)-i}$	$120 - 58 = 62$	$105 - 41 = 64$
$\tau_{i-(i+1)}$	$ 28.8 = 29$	$ 64.0 = 64$
τ_{i-i}	$1/0.5 = 2$	

TABLEAU 2.4 – Valeur des paramètres du modèle d'une rue découpée en sections.

Le comportement du GET de la figure 2.7, configuré avec les valeurs du tableau 2.4, peut être décrit par les équations suivantes dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$:

$$\begin{aligned}
 X &= \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \gamma^2 \delta^2 & \gamma^{62} & \varepsilon \\ \gamma^{58} \delta^{29} & \gamma \delta^2 & \gamma^{64} \\ \varepsilon & \gamma^{41} \delta^{64} & \gamma \delta^2 \end{pmatrix} X \oplus \begin{pmatrix} \mathbf{e} & \varepsilon \\ \varepsilon & \mathbf{e} \\ \varepsilon & \mathbf{e} \end{pmatrix} \begin{pmatrix} u \\ \\ init \end{pmatrix} , \\
 y &= (\varepsilon \ \varepsilon \ \mathbf{e}) X .
 \end{aligned}$$

Nous pouvons calculer les bornes de temps de séjour des véhicules dans la rue (*i.e.* la portion $\{c_0, c_2\}$), autrement dit, les bornes de temps de séjour des jetons dans le chemin $x_0 \rightarrow x_2$. Les bornes de séjour relatives aux valeurs de paramètres définies dans le tableau 2.4 et à une séquence d'entrée $u = [\gamma\delta]^*$ (*i.e.* arrivée d'un véhicule chaque seconde) ont les valeurs suivantes :

- le temps de séjour minimum relatif est de 198 secondes ($(\gamma^{-99}x_2u)\varnothing(x_0u) = \delta^{198}[\gamma\delta^2]^*$);
- le temps de séjour maximum relatif est de 322 secondes ($(x_0u)\varnothing(\gamma^{-99}x_2u) = \delta^{-322}[\gamma\delta^2]^*$).

Il est également possible de calculer les bornes de temps de séjour absolues :

- le temps de séjour minimum absolu est de 93 secondes ($((x_2 \otimes (\mathbf{e} \oplus \gamma\delta^*))\varnothing(x_0 \otimes (\mathbf{e} \oplus \gamma\delta^*))) = \delta^{93}[\gamma\delta^2]^*$);
- le temps de séjour maximum absolu est de 450 secondes ($((x_0 \otimes \mathbf{e})\varnothing(\gamma^{-225}x_2 \otimes \mathbf{e})) = \delta^{-450}[\gamma\delta^2]^*$).

Enfin, les bornes de temps de séjour des jetons avec marquage initial nul sont, quelque soit la séquence d'entrée, les suivantes :

- le temps de séjour minimum absolu hors marquage initial est de 93 secondes ($(x_2\varnothing x_0 = \delta^{93}[\gamma\delta^2]^*$);
- le temps de séjour maximum absolu hors marquage initial est de 304 secondes ($(x_0\varnothing x_2 = \gamma^{-304}[\gamma\delta^2]^*$).

D'après les expressions du taux d'occupation et du débit instantanés proposées dans la sous-section 2.2.2.C page 63, nous obtenons les diagrammes fondamentaux de la figure 2.8.

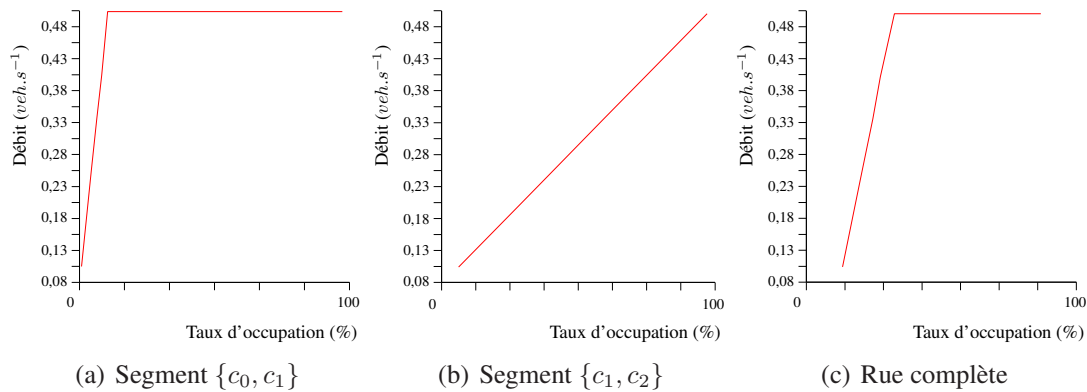


FIGURE 2.8 – Diagrammes fondamentaux du système.

À travers la simulation, nous pouvons immédiatement constater que la forme des diagrammes fondamentaux exprime les situations du trafic habituellement rencontrées dans la littérature. Ces figures montrent que le débit augmente jusqu'à une certaine valeur du taux d'occupation (valeur critique), ce qui exprime la situation du régime fluide. Étant donné que le système est ouvert et non contrôlé, rien n'empêche les véhicules d'être évacués en un temps fini dépendant de la vitesse du système en régime permanent. Le régime congestionné est exprimé par la partie horizontale de la courbe, montrant que la capacité de la rue est atteinte. Bien que la dernière partie de la courbe ne soit pas décroissante, nous retrouvons bien les deux parties du diagramme fondamental, à savoir celles décrivant les situations fluide et congestionnée.

Le fait de ne pas obtenir une courbe de régime congestionné décroissante est dû en grande partie à la discrétisation du système qui ne permet pas de visualiser les congestions absorbées à travers les taux d'occupations moyens. En effet, la théorie macroscopique du trafic suppose des variables du trafic continues dans l'espace et dans le temps. Cette différence d'approche laisse certaines mesures liées à la continuité des variables à l'intérieur des segments de route irréalisables dans le cadre de notre modèle. Plus précisément, cette difficulté de mesure s'explique, d'une part, par la fluidité du trafic en aval et, d'autre part, par l'onde de choc [Cohen, 1993] qui propage la congestion en amont en-dehors des segments modélisés. Cependant, le phénomène de la congestion peut être mieux apprécié à travers la variable du temps maximal de séjour. Ainsi, la figure 2.9 présente les variations du temps maximal de séjour en fonction du taux d'occupation.

Nous pouvons constater à travers la figure 2.9 que le temps maximal de parcours augmente proportionnellement au taux d'occupation. Ceci correspond parfaitement au phénomène de la congestion. Celui-ci s'exprime par un ralentissement dû à l'augmentation du nombre de véhicules circulant dans le segment. Ainsi, à travers les différentes simulations présentées ci-dessus, il est possible de confirmer que le modèle GET permet de représenter les différents états du trafic.

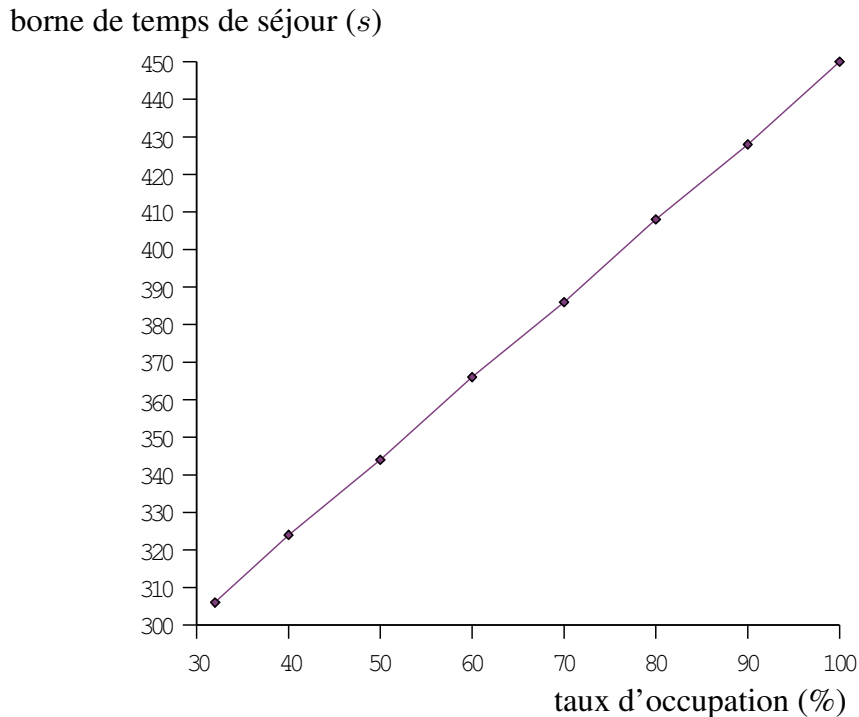


FIGURE 2.9 – Borne supérieure de temps de séjour en fonction du taux de remplissage.

2.3 Régulation du trafic

Puisque le modèle a été validé, nous pouvons l'appliquer à la régulation du trafic dans une autoroute. Le but est ici de garantir un trafic fluide dans chaque segment. Nous proposons de réduire le temps de parcours des véhicules en limitant l'accès à l'autoroute.

Les méthodes classiques de la commande des **GETs** par l'algèbre des diodes sont expliquées dans [Cottenceau, 1999, Cottenceau *et al.*, 2001, Lhommeau, 2003, Maia *et al.*, 2003]. Nous rappelons brièvement quelques résultats de [Lahaye *et al.*, 2004, Corrêia, 2004, Corrêia *et al.*, 2006c] sur la commande sous contraintes. Ces résultats permettent de réduire les bornes de temps de séjour des jetons (hors marquage initial) à l'aide d'un correcteur placé en amont des entrées, appelé *précompensateur*. Ce correcteur, calculé mathématiquement, peut être réalisé sous la forme d'un **GET**.

Dans un **GET**, le temps de séjour d'un jeton dans une place peut être très supérieur au délai de cette place. Par exemple, dans le **GET** de la figure 2.2 page 51, la synchronisation implique que le jeton initialement présent aura un temps de séjour supérieur à la temporisation de la place. En ce sens, le modèle de **GET** n'intègre pas les contraintes temporelles inhérentes aux systèmes dits à *temps critique* ou à *temps contraint*. Par exemple, il peut être souhaitable que les véhicules ne restent pas trop longtemps dans une section de la route pour des raisons de pollution ou de sécurité. Nous proposons de résoudre ce problème par une loi de commande.

Grâce à cette loi de commande, nous allons limiter le temps de séjour des jetons dans certains chemins d'un **GET**. Considérons un chemin $x_j \rightarrow x_i$ (de la transition x_j vers la transition

x_i), contenant initialement α_{ij} jetons. Le respect d'une contrainte temporelle, notée τ_{ij} , sur ce chemin s'exprime dans l'algèbre classique :

$$\underbrace{x_i(k + \alpha_{ij})}_{\substack{\text{date de sortie du jeton} \\ \text{introduit par le } k\text{e tir de } x_j}} - \underbrace{x_j(k)}_{\substack{\text{date du } k\text{e tir de } x_j}} \leq \tau_{ij} \quad ,$$

ou encore :

$$(-\tau_{ij}) + x_i(k + \alpha_{ij}) \leq x_j(k) \quad .$$

Soit la contrainte suivante sur les séries de $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ associées aux transitions :

$$\gamma^{-\alpha_{ij}} \delta^{-\tau_{ij}} x_i \preceq x_j \quad . \quad (2.12)$$

Selon l'inégalité (2.12), les contraintes temporelles τ_{ij} imposées à certains chemins d'un GET peuvent se traduire par une inégalité sur le vecteur d'état du système corrigé X_c :

$$\Delta X_c \preceq X_c, \quad \forall V, \quad (2.13)$$

où $\Delta \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times n}$ est tel que :

$$[\Delta]_{ji} = \begin{cases} \gamma^{-\alpha_{ij}} \delta^{-\tau_{ij}} & \text{s'il existe une contrainte sur le chemin } x_j \rightarrow x_i, \\ \varepsilon & \text{sinon.} \end{cases}$$

Soit un GET ayant pour matrice de transfert $H = CA^*B$ et soit un modèle de référence de transfert G_{ref} . Le précompensateur optimal, c'est-à-dire qui permet de respecter les contraintes de temps tout en garantissant que le système corrigé est au moins aussi rapide que G_{ref} , est donné par la proposition suivante.

Proposition 2.2 (Plus grand précompensateur [Corréia et al., 2006c]) *Le plus grand précompensateur P tel que le système corrigé respecte la ou les contrainte(s) temporelle(s) spécifiée(s) par Δ et tel que son transfert soit inférieur à G_{ref} , est la plus grande solution de :*

$$P \preceq \frac{G_{ref}}{H} \wedge \frac{A^*BP}{\Delta A^*B} \wedge \text{Pr}_+(P) \quad , \quad (2.14)$$

où $\Delta \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times n}$ est tel que :

$$[\Delta]_{ji} = \begin{cases} \gamma^{-\kappa_{ij}} \delta^{-\tau_{ij}} & \text{s'il existe une contrainte } \tau_{ij} \text{ sur le chemin } x_j \rightarrow x_i \text{ ayant} \\ & \text{pour marquage initial } \kappa_{ij}, \\ \varepsilon & \text{sinon.} \end{cases}$$

Preuve: Développons l'inégalité (2.13) :

$$\begin{aligned} & \Delta X_c \preceq X_c , \\ \Leftrightarrow & \Delta A^*BPV \preceq A^*BPV \quad \forall V, \\ \Leftrightarrow & \Delta A^*BP \preceq A^*BP . \end{aligned}$$

Or d'après la théorie de la résiduation, $ax \preceq b \Leftrightarrow x \preceq a \oslash b$. Ainsi l'inégalité précédente est équivalente à

$$P \preceq \frac{A^*BP}{\Delta A^*B} . \quad (2.15)$$

Le précompensateur doit être causal pour être réalisable : $P = \text{Pr}_+(P)$. Or il est aisé de vérifier que $\text{Pr}_+(P) \preceq P$, donc la relation suivante peut être considérée comme une contrainte sur P :

$$P \preceq \text{Pr}_+(P) . \quad (2.16)$$

En résumé, en partant de (2.15) et (2.16) tout en gardant à l'esprit que $HP \preceq G_{ref} \Leftrightarrow P \preceq H \oslash G_{ref}$, le résultat (2.14) apparaît naturellement. ■

Nous observons que le précompensateur P est défini par une inégalité au *point fixe* (i.e. de la forme $P \preceq f(P)$). De par la nature ordonnée des dioïdes, les résultats concernant les points fixes énoncés pour les treillis s'appliquent. En particulier, un théorème bien connu [Tarski, 1955, Lassez *et al.*, 1982] permet d'affirmer l'existence d'une plus grande solution à $f(x) = x$ lorsque f est isotone. Le résultat suivant spécifie une méthode classique pour son calcul dans un dioïde complet [Davey et Priestley, 1990, Kumar et Garg, 1995, Zad *et al.*, 1999]. Soit h une application isotone d'un dioïde complet \mathcal{D} dans lui-même, soit $w \in \mathcal{D}$ et soit l'application isotone f définie par $\forall x \in \mathcal{D}, f(x) = h(x) \wedge w$. Si la suite

$$\begin{aligned} y_0 &= w , \\ y_{k+1} &= h(y_k) \wedge y_k , \end{aligned} \quad (2.17)$$

converge en un nombre fini k_f d'itérations, alors y_{k_f} est le plus grand point fixe de f .

Nous pouvons directement appliquer ce résultat pour calculer le correcteur P défini par l'équation (2.14). En effet, si la suite définie par

$$\begin{aligned} y_0 &= \frac{G_{ref}}{H} , \\ y_{k+1} &= \frac{A^*By_k}{\Delta A^*B} \wedge \text{Pr}_+(y_k) \wedge y_k , \end{aligned}$$

converge en un nombre fini k_f d'itérations, alors y_{k_f} est le plus grand précompensateur solution de (2.14). Certaines conditions sont suffisantes pour garantir la convergence de cette suite, elles sont discutées dans [Houssin, 2006].

Reprenons l'exemple de la sous-section 2.2.3 page 63. Ce système est présenté par la figure 2.6 ; son modèle est illustré par la figure 2.7 page 64.

Nous avons vu que le temps de séjour maximum absolu des jetons hors marquage initial dans la portion $\{c_0, c_2\}$, correspondant au chemin $x_0 \rightarrow x_2$ du modèle GET, est de 304 secondes. Nous souhaitons réduire cette borne à 93 secondes, afin de garantir un trafic fluide à tout instant.

Puisque l'objectif de cette démarche est de limiter la borne du temps de séjour des jetons dans un chemin hors marquage initial, nous pouvons partir de l'hypothèse d'une rue vide. Cette hypothèse rend la transition u_{init} et ses places aval inutiles, aussi nous les retirons du modèle qui servira pour le calcul du correcteur. La figure 2.10 donne le modèle GET ainsi obtenu.

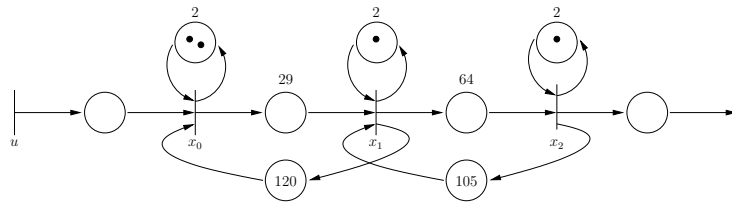


FIGURE 2.10 – Modèle GET d'une rue initialement vide.

Le modèle algébrique diffère quelque peu de celui utilisé dans la sous-section 2.2.3 ; il est donné par les équations suivantes :

$$X = \begin{pmatrix} \gamma^2 \delta^2 & \gamma^{120} & \varepsilon \\ \delta^{59} & \gamma \delta^2 & \gamma^{105} \\ \varepsilon & \delta^{64} & \gamma \delta^2 \end{pmatrix} X \oplus \begin{pmatrix} \mathbf{e} \\ \varepsilon \\ \varepsilon \end{pmatrix} u \quad ,$$

$$y = (\varepsilon \quad \varepsilon \quad \mathbf{e}) X \quad .$$

Nous choisissons une *correction neutre*, autrement dit notre modèle de référence G_{ref} a un transfert identique au système nominal H . Ainsi,

$$H = (\delta^{93} [\gamma \delta^2]^*) \quad ,$$

$$G_{ref} = H \quad .$$

Compte tenu de la contrainte temporelle, la matrice Δ est définie par

$$\Delta = \begin{pmatrix} \varepsilon & \varepsilon & \delta^{-93} \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \quad .$$

L'algorithme converge en une seule itération :

$$y_0 = \frac{G_{ref}}{H} = [\gamma \delta^2]^* \quad ,$$

$$y_1 = \frac{A^* B}{\Delta A^* B y_0} \wedge \text{Pr}_+(y_0) \wedge y_0 = [\gamma \delta^2]^* = y_0 \quad .$$

D'où

$$P = [\gamma\delta^2]^* .$$

Une réalisation de ce précompensateur est donnée par la figure 2.11.

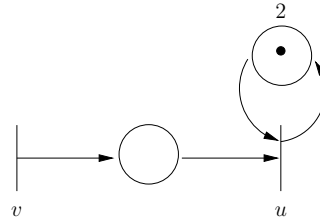


FIGURE 2.11 – Réalisation d'un précompensateur GET pour le modèle de la figure 2.10.

La borne supérieure absolue de temps de séjour des jetons dans le chemin $x_0 \rightarrow x_2$ du système corrigé est de 93 secondes ($x_0 \oslash x_2 = \delta^{-93}[\gamma\delta^2]^*$), ce qui remplit l'objectif fixé par la contrainte. En effet, le précompensateur P garantit que les véhicules entrant dans la rue peuvent la parcourir en un temps égal au temps de parcours à vitesse maximum (93 secondes). En conséquence de quoi, le trafic est toujours fluide.

Pour parvenir à ce résultat, le correcteur filtre l'accès des véhicules à l'autoroute. En l'occurrence, seul un véhicule toutes les deux secondes peut accéder à l'autoroute. Le précompensateur fait correspondre le débit d'arrivée de nouveaux véhicules avec le débit maximum du système dans son ensemble, lequel est limité par le débit maximum du segment ayant la capacité d'écoulement la plus faible, à savoir le segment $c_1 \rightarrow c_2$. Celui-ci ne possède en effet qu'une voie, contre deux voies pour le segment $c_0 \rightarrow c_1$. Ce dernier ne peut donc pas être exploité au maximum de son débit, tandis que le segment $c_1 \rightarrow c_2$ peut l'être.

Conclusion

Après quelques rappels sur les RdPs et plus précisément les GETs, nous avons mentionné certains résultats de la littérature expliquant comment décrire le comportement des GETs au moyen d'équations linéaires dans l'algèbre des dioïdes. Nous avons alors introduit un nouveau modèle mesoscopique du trafic en assimilant le trafic aux systèmes à événements discrets. Ce modèle associe un GET à chaque segment de route, lesquels sont délimités par des changements de propriété de la route ou du trafic (vitesse moyenne, nombre de voies, ...).

Nous avons validé ce modèle à travers la simulation. En effet, nous avons constaté qu'il décrit parfaitement les cas de trafic fluide, mais également le phénomène de congestion. De plus, il permet d'évaluer de nombreux paramètres tels que les bornes de temps de parcours, le nombre instantané de véhicules dans chaque segment, ou encore le débit et la densité.

En appliquant certains résultats bien connus de la commande des GETs par l'algèbre des dioïdes combinée à la théorie de la résiduation, notre modèle permet de garantir un trafic fluide

en toute circonstance. Pour cela, un précompensateur est calculé. Celui-ci filtre les entrées de véhicules de manière à homogénéiser le trafic. Plus précisément, le débit maximum de ce correcteur est égal au plus petit débit maximum parmi les segments.

Ce modèle de rue peut être utilisé dans la régulation des carrefours par les feux. Cependant, il ne permet pas à lui seul de représenter le comportement des véhicules aux intersections. En effet, celles-ci représentent par nature un conflit, puisque plusieurs mobiles tentent d'accéder simultanément à une seule et unique ressource qu'est la zone de conflit. Or, par définition, les **GETs** ne permettent pas de modéliser des structures de choix. Si nous souhaitons conserver un modèle **RdP** avec une écriture dans les dioïdes, nous allons donc devoir enrichir cet outil algébrique.

Aussi, il nous incombe désormais de développer une nouvelle méthode de commande des **RdP** à l'aide de l'algèbre des dioïdes, qui soit en concordance avec les objectifs de l'ingénierie du trafic. Ce but ne saurait être atteint sans la résolution d'un problème inhérent à la définition même des **GETs**.

Chapitre



Régulation de carrefour : un problème de partage de ressources

3.1	Gestion des conflits dans les dioïdes	75
3.1.1	Problème de ressources partagées et état de l'art	76
3.1.2	Modèle analytique	79
3.1.3	Écriture du modèle dans l'algèbre des dioïdes	82
3.1.4	Exemple : modélisation d'une gare de fret	82
3.2	Modélisation et commande d'un carrefour simplifié	91
3.2.1	Modèle d'un carrefour simplifié	91
3.2.2	Commande par fenêtres de temps	96
3.2.3	Commande optimale	98
3.3	Étude comparative	103

Introduction

DEPUIS une dizaine d'années, plusieurs chercheurs proposent des avancées dans la modélisation et la commande du trafic par l'algèbre des dioïdes. Alors que les premiers travaux, notamment [van Egmond et Olsder, 1998], proposent de synchroniser les feux de manière à réaliser une onde verte, des travaux plus récents tentent de retranscrire le plus fidèlement possible le comportement des véhicules, afin d'obtenir une analyse fine [Mancinelli *et al.*, 2001, Lotito *et al.*, 2005]. Ces travaux se heurtent cependant à un défi majeur, induit par le fait que les GETs ne peuvent retranscrire les conflits inhérents aux carrefours. De ce fait, il faut élargir la portée de l'algèbre des dioïdes à une classe plus générale de RdP [Farhi *et al.*, 2005, Farhi *et al.*, 2007].

Dans un avenir proche, chaque véhicule sera sans doute équipé de capteurs modernes tels le GPS ou les caméras vidéo. Nous proposons dans ce contexte un modèle analytique accompagné d'une loi de commande. L'ensemble, implémenté dans chaque véhicule, peut servir d'interface entre un système de positionnement et un système de guidage et de régulation de vitesse automatique.

La difficulté technique centrale reste la gestion des conflits dans l'algèbre des dioïdes. L'exploration de cette piste nous conduit à définir un nouveau modèle algébrique générique, capable de décrire le comportement d'une large majorité des problèmes de ressources partagées. Ce modèle permet de construire complètement l'ensemble des solutions admissibles, sans inclure de règle d'arbitrage *a priori*. D'un point de vue théorique, il est en effet intéressant de modéliser le problème en-dehors de toute solution particulière.

Après avoir défini le problème de ressources partagées et ses différentes modélisations, proposées dans la littérature, nous introduisons un nouveau modèle générique pour ce type de problème. Nous appliquons ensuite ce modèle à l'analyse puis la commande d'un carrefour urbain simplifié. Enfin, nous validons ce modèle et sa commande en comparant les résultats obtenus en simulation avec ceux des travaux les plus récents dans le domaine.

3.1 Gestion des conflits dans les dioïdes

L'intersection contient, par nature, un conflit qui doit être retranscrit comme tel dans son modèle RdP. Avant de proposer un modèle de trafic dans l'algèbre des dioïdes, nous devons donc commencer par résoudre le problème de la modélisation du conflit dans cette algèbre. Nous rappelons en effet que les GETs, dont le comportement peut être intégralement décrit par des équations linéaires dans l'algèbre des dioïdes, ne permettent pas de modéliser les conflits.

Puisque notre application relève d'un type particulier de conflit, nous commençons cette section en le définissant. Nous énumérons ensuite plusieurs modèles de la littérature permettant d'éviter ou de résoudre partiellement ce type de conflit. Enfin, nous proposons un nouveau modèle générique permettant de définir rigoureusement l'ensemble des solutions admissibles par un système incluant un tel conflit. L'avantage de ce modèle est d'élargir l'ensemble des systèmes se trouvant dans le champ d'application des résultats obtenus.

3.1.1 Problème de ressources partagées et état de l'art

Les **SEDs** sont des processus caractérisés par des séquences d'événements [Baccelli *et al.*, 1992, p. *ix*]. Le *problème de ressources partagées* est couramment rencontré dans les **SEDs**. Il survient à chaque fois que deux utilisateurs souhaitent accéder à la même ressource au même moment. Un tel problème est habituellement traité par la recherche opérationnelle. Dans ce domaine, les modèles proposés sont définis à travers une approche empirique. De ce fait, chaque modèle ne convient que pour une classe particulière de problèmes, qui dépend de la dynamique du système [Graham *et al.*, 1979]. Cependant, dans un souci de généralité, l'approche de modélisation se doit d'inclure le comportement complet des utilisateurs vis-à-vis des ressources.

Cette situation a motivé de nombreux chercheurs pour développer des méthodes et techniques afin de contourner la difficulté rencontrée dans les **SEDs**. Dans [Lampert, 1978, Naimi *et al.*, 1996], des algorithmes d'exclusion mutuelle ont été développés pour traiter les cas liés à l'informatique. Cependant, ils ne conviennent pas à une modélisation analytique.

3.1.1.A Littérature des **RdPs**

Plusieurs outils graphiques sont proposés pour représenter les problèmes de ressources partagées, tels que les **RdPs**. En particulier, dans [Moody *et al.*, 1995, Moody et Antsaklis, 2000], les auteurs s'affranchissent du problème en ajoutant un correcteur au système. Cependant, une telle approche ne permet pas de percevoir clairement l'influence des divers facteurs sur le comportement du système dans son ensemble. Dans [Desrochers et Al-Jaar, 1995] et [Giua *et al.*, 1992] deux modèles basés sur les **RdPs** sont proposés. Ils décrivent respectivement les phénomènes d'exclusion mutuelle parallèle et généralisée. La contribution principale de ces travaux consiste en la simulation du système. Cependant, même si les approches basées sur la simulation sont encore largement utilisées pour évaluer et comparer différents scénarios, le besoin de développer une formulation mathématique pertinente capable de produire des solutions de grande qualité à ce type de problème complexe se fait sentir.

Les **GETs** forment la classe de **RdPs** temporisés et déterministes qui remplissent ces deux objectifs en offrant un mécanisme de spécification mathématique pratique et une bonne interface pour l'analyse et la simulation des systèmes étudiés [Murata, 1989, p. 554] et [David et Alla, 1992, pp. 27 et 125]. En effet, cette classe admet une modélisation algébrique linéaire dans les dioïdes. Malheureusement, par définition, toutes les places d'un **GET** ont exactement une transition amont et une transition aval. Ceci permet de modéliser les phénomènes de synchronisation et de parallélisme, mais pas celui de la concurrence.

3.1.1.B Littérature des dioïdes

De nombreux efforts ont été fournis pour appréhender cette contrainte de modélisation. En particulier, dans [Trouillet et Benasser, 2002, Trouillet et Gentina, 2003], les auteurs traitent le problème en transformant le **RdP** en **GM**. Ceci permet l'écriture du comportement dans l'algèbre des dioïdes. Cependant ils imposent que l'attribution des ressources partagées aux

utilisateurs suivre une séquence rigide, donnée *a priori*. C'est pourquoi une telle approche ne peut nous permettre de prendre en considération les différentes possibilités offertes par le système. Dans [van den Boom et De Schutter, 2006], les auteurs proposent un modèle à permutations de comportements, lesquels sont décrits par des équations linéaires dans $\overline{\mathbb{Z}}_{\max}$. Ils considèrent que les SEDs peuvent commuter d'un mode de fonctionnement à un autre suivant une séquence soit donnée, soit aléatoire, soit calculée suivant un critère d'optimisation. Cependant, lorsque nous essayons de modéliser des systèmes avec beaucoup de ressources partagées, le nombre de modes peut considérablement augmenter, de façon exponentielle. Ainsi, ce modèle nécessite une analyse qui est à la fois difficile à accomplir sur le plan pratique et très coûteuse en temps de calcul. Un autre travail remarquable dans ce domaine est celui présenté dans [Al Saba *et al.*, 2006], où une combinaison de GETs et d'« automates de type tas » [Gaubert et Mairesse, 1999] est commandée par un algorithme basé sur la « règle de première date due » [Jackson, 1955]. Cette règle organise les utilisations de ressources par ordre croissant des dates de fin d'utilisation souhaitées. Les deux types de composants (GETs et automates) doivent être répartis de manière séquentielle et ne peuvent être imbriqués l'un dans l'autre. Cette limitation empêche, comme le mentionnent les auteurs, d'appréhender des systèmes plus génériques.

3.1.1.C Système à Ressources Partagées Invariantes

Afin de situer rigoureusement notre travail par rapport à la littérature, nous définissons formellement la famille de problèmes que nous traitons dans cette section.

Définition 3.1 (Système à Ressources Partagées Invariantes) *Un Système à Ressources Partagées Invariantes (SRPI) est un SED dans lequel nous pouvons identifier un ensemble composé d'un nombre constant de ressources, ainsi qu'un ensemble fini d'utilisateurs de ces ressources. Ces utilisateurs requièrent et restituent les ressources, de la manière suivante : soit $v_i(t)$ et $x_i(t)$ le nombre de fois qu'un utilisateur i a, respectivement, requis ou restitué une ressource jusqu'à la date t , alors la quantité $v_i(t) - x_i(t) \geq 0$, $\forall t$ est le nombre de ressources détenues par l'utilisateur i à l'instant t .*

Les SEDs incluant un ou plusieurs SRPIs se retrouvent dans de nombreux domaines d'applications tels que les chaînes de production et de livraison (domaine industriel), les systèmes temps-réel (domaine informatique) et bien sûr les systèmes ferroviaires et routiers (domaine du transport). Lorsqu'un tel système ne contient pas par nature une règle stricte d'attribution des ressources aux utilisateurs, nous sommes face à un *Problème de Ressources Partagées Invariantes (PRPI)*, lequel focalise notre intérêt. En effet, nous tenons à traiter ce problème au sens large en définissant strictement l'espace de solutions. D'un point de vue théorique, il convient de modéliser le problème en-dehors de toute solution particulière. Puisque le problème traité inclut des conflits, nous ne pouvons pas le modéliser à l'aide d'un GET. Ainsi, le modèle graphique générique d'un SRPI présenté par la figure 3.1 appartient à une classe plus générale de RdPs.

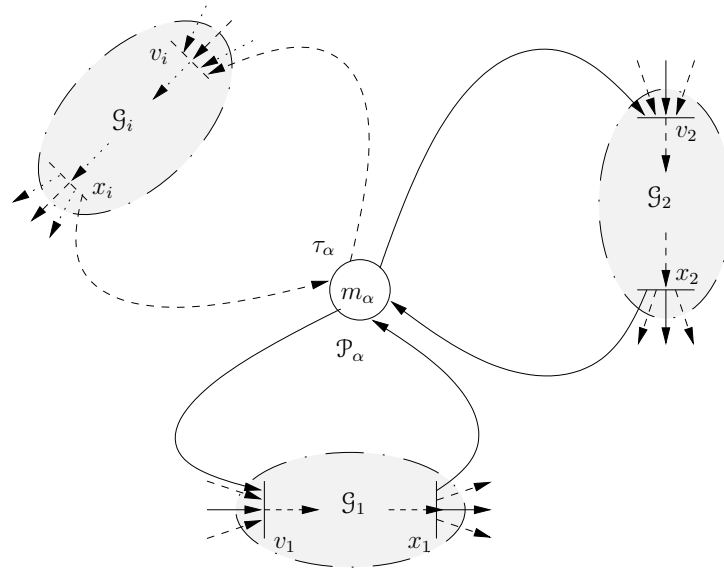


FIGURE 3.1 – Modèle RdP d'un PRPI.

Ce modèle graphique représente \mathcal{N} utilisateurs partageant r ressources de même nature. Nous assignons à chaque utilisateur $i \in [1, \mathcal{N}]$ un sous-RdP \mathcal{G}_i qui modélise son comportement pendant qu'il utilise une ou plusieurs ressources (*section critique* de i). Un tel sous-RdP ne doit avoir qu'une transition d'entrée et une transition de sortie en relation avec la place \mathcal{P}_α contenant les ressources disponibles ; ces transitions sont notées v_i et x_i respectivement.

Dans cette représentation générale, nous pouvons observer l'évolution des compteurs de chaque transition. Le tableau 3.1 détaille la sémantique du modèle.

v_i	transition d'entrée : $v_i(t)$ est le nombre de fois que l'utilisateur i a requis une ressource jusqu'à la date t
x_i	transition de sortie : $x_i(t)$ est le nombre de fois que l'utilisateur i a restitué une ressource jusqu'à la date t
τ_α	durée (constante) de l'indisponibilité de chaque ressource entre deux utilisations (<i>e.g.</i> réglage d'une machine, changement d'outil, ...)
m_α	nombre de ressources disponibles initialement ($m_\alpha \leq r$)
m_i	nombre de ressources initialement détenues par l'utilisateur i

TABLEAU 3.1 – Sémantique associée à la figure 3.1.

Hormis le travail présenté dans [van den Boom et De Schutter, 2006], qui conduit à un nombre de modes exponentiel, nous rappelons qu'il n'y a pas de technique systématique aussi générale disponible dans la littérature pour modéliser de tels problèmes dans l'algèbre des dioïdes. Le but des sous-sections suivantes est précisément de combler ce manque théorique [Correia et al., 2007b].

3.1.2 Modèle analytique

Le modèle RdP de la figure 3.1 est scindé en deux parties, tel que montré en figure 3.2. Nous distinguons la partie concernant les utilisateurs, qui contient les sous-RdP notés \mathcal{G}_i , de celle relevant des ressources partagées, dénotée \mathcal{G}_α . Nous rappelons que nous limitons notre contribution au cas discret dans lequel l'invariance de ressources est respectée, à savoir, il existe, à tout instant $t \geq 0$, une fonction f et un entier naturel τ tel que $x_i(t + \tau) = f(v_i(t))$.

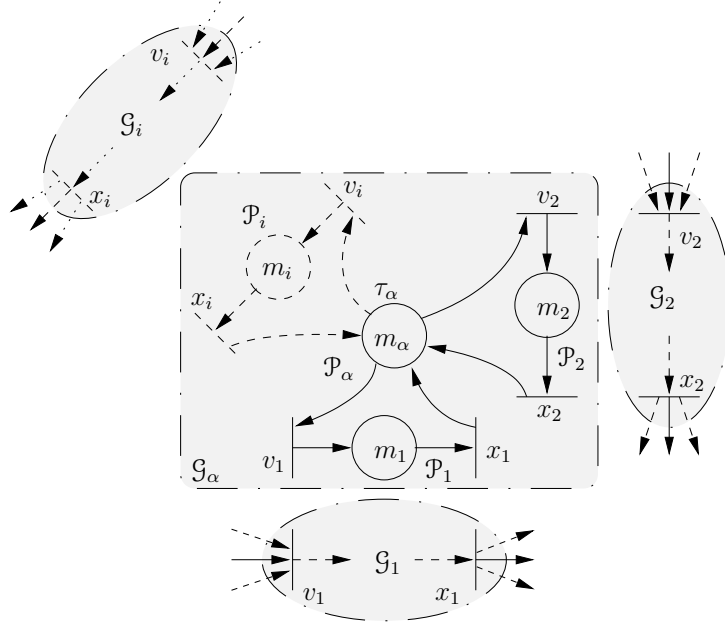


FIGURE 3.2 – Modèle scindé d'un PRPI.

Dans ce modèle, \mathcal{G}_α est construit autour de la place \mathcal{P}_α , dont le marquage $\mathcal{M}_t(\mathcal{P}_\alpha)$ représente le nombre de ressources non utilisées (disponibles ou non) à la date t . À ce stade, il est important de noter que \mathcal{G}_α ne suit pas un fonctionnement au plus tôt. Ainsi, dans le but d'étudier le comportement des transitions d'entrée et de sortie de chaque sous-RdP \mathcal{G}_i dans les conditions de partage de ressources, celles-ci sont dupliquées. Sans perte de généralité, chaque sous-RdP \mathcal{G}_i est remplacé par une place fictive \mathcal{P}_i dans \mathcal{G}_α , de telle manière que son marquage initial $\mathcal{M}_0(\mathcal{P}_i)$ corresponde au nombre de ressources détenues initialement par l'utilisateur i ($\mathcal{M}_0(\mathcal{P}_\alpha) + \sum_{i=1}^N \mathcal{M}_0(\mathcal{P}_i) = r$). Ceci contribue à concentrer l'analyse sur les ressources partagées, quelque soit le comportement des utilisateurs. En résumé, nous considérons \mathcal{G}_α comme le modèle élémentaire d'un SRPI. L'analyse structurelle de \mathcal{G}_α nous fournit le lemme suivant.

Lemme 3.1 (Contrainte) Soient N utilisateurs partageant r ressources de même nature. Si, à tout instant t , chaque ressource est utilisée par au plus un utilisateur, alors les compteurs de leurs transitions d'entrée $v_i(\cdot)$ et de sortie $x_i(\cdot)$ satisfont l'inégalité suivante :

$$\sum_{i=1}^N (v_i(t) - x_i(t)) \leq \mathcal{M}_0(\mathcal{P}_\alpha) \quad \forall t. \quad (3.1)$$

Preuve: \mathcal{G}_α constitue un invariant de place, il vient donc, pour tout t :

$$\sum_{i=1}^{\mathcal{N}} \mathcal{M}_t(\mathcal{P}_i) + \mathcal{M}_t(\mathcal{P}_\alpha) = r \quad .$$

D'après [Cohen *et al.*, 1991], nous avons $\mathcal{M}_t(\mathcal{P}_i) = v_i(t) - x_i(t) + \mathcal{M}_0(\mathcal{P}_i)$. Ainsi,

$$\begin{aligned} \sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t) + \mathcal{M}_0(\mathcal{P}_i)) + \mathcal{M}_t(\mathcal{P}_\alpha) &= r \quad , \\ \Leftrightarrow \sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t)) + \mathcal{M}_t(\mathcal{P}_\alpha) &= r - \sum_{i=1}^{\mathcal{N}} \mathcal{M}_0(\mathcal{P}_i) \quad . \end{aligned}$$

Puisque $r - \sum_{i=1}^{\mathcal{N}} \mathcal{M}_0(\mathcal{P}_i) = \mathcal{M}_0(\mathcal{P}_\alpha)$, il vient aussitôt $\sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t)) + \mathcal{M}_t(\mathcal{P}_\alpha) = \mathcal{M}_0(\mathcal{P}_\alpha)$. Puisque $\mathcal{M}_t(\mathcal{P}_\alpha)$ est positif pour tout t , le lemme est démontré. ■

Il est important de noter que ce lemme a un fort pouvoir de contrainte sur le système étudié. En effet, il fournit une relation qui implique seulement des compteurs de transitions de \mathcal{G}_α et le nombre de ressources initialement disponibles. Cependant, aux niveaux théorique et pratique, il est intéressant de vérifier que la contrainte (3.1) est en accord avec le comportement typique d'un SRPI. Pour ce faire, il suffit de montrer que :

- (a) la contrainte rejette toute réquisition de ressource s'il n'y en a plus de disponible,
- (b) s'il y a au moins une ressource disponible, la contrainte honore une telle requête,
- (c) la contrainte ne doit pas empêcher les utilisateurs de restituer une ressource.

Remarquons que la condition (c) est évidemment satisfaite puisque les compteurs des transitions $x_i(\cdot)$ sont multipliés par -1 dans la contrainte (3.1). Ce qui signifie que chaque tir d'une telle transition relaxe la contrainte. Ainsi, il reste à prouver (a) et (b), ce qui est le but du lemme suivant.

Lemme 3.2 *Soit un SRPI. Alors, sous la contrainte (3.1), une transition v_i peut être tirée si au moins une ressource est disponible.*

Preuve: Supposons qu'une transition v_i soit franchie alors qu'aucune ressource n'est disponible. Lorsqu'un tel cas se produit à une date t , nous avons :

$$\sum_{i=1}^{\mathcal{N}} \mathcal{M}_t(\mathcal{P}_i) = \sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t) + \mathcal{M}_0(\mathcal{P}_i)) > r \quad .$$

Or $r = \sum_{i=1}^{\mathcal{N}} \mathcal{M}_t(\mathcal{P}_i) + \mathcal{M}_t(\mathcal{P}_\alpha)$. En particulier, pour $t = 0$, nous avons aussi $r = \sum_{i=1}^{\mathcal{N}} \mathcal{M}_0(\mathcal{P}_i) + \mathcal{M}_0(\mathcal{P}_\alpha)$. Il s'ensuit que

$$\sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t) + \mathcal{M}_0(\mathcal{P}_i)) > \mathcal{M}_0(\mathcal{P}_\alpha) + \sum_{i=1}^{\mathcal{N}} \mathcal{M}_0(\mathcal{P}_i) \quad .$$

Ainsi, nous obtenons

$$\sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t)) > \mathcal{M}_0(\mathcal{P}_\alpha) \quad ,$$

ce qui contredit la contrainte (3.1). ■

Le lemme suivant montre comment combiner la contrainte (3.1) et la relation de transfert entre x_i et v_i dans une inégalité.

Lemme 3.3 *Soient \mathcal{N} utilisateurs partageant r ressources. Si, à tout instant t , chaque ressource est utilisée par au plus un utilisateur, alors les compteurs de leurs transitions d'entrée $v_i(\cdot)$ et de sortie $x_i(\cdot)$ satisfont l'inégalité suivante :*

$$\sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t - \tau_\alpha)) \leq \mathcal{M}_0(\mathcal{P}_\alpha) \quad \forall t.$$

Preuve: Soulignons tout de suite qu'il existe une relation de transfert entre les transitions v_i et x_i pour chaque utilisateur i , qui dépend du comportement du graphe \mathcal{G}_α . Cette dernière est donnée par

$$\sum_{i=1}^{\mathcal{N}} v_i(t) \leq \sum_{i=1}^{\mathcal{N}} x_i(t - \tau_\alpha) + \mathcal{M}_0(\mathcal{P}_\alpha) \quad .$$

Rappelons également que les fonctions compteur sont non-décroissantes : notamment, pour tous i et t , $x_i(t) \geq x_i(t - \tau_\alpha)$. Ainsi, la relation précédente implique

$$\sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t - \tau_\alpha)) \leq \mathcal{M}_0(\mathcal{P}_\alpha) \implies \sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t)) \leq \mathcal{M}_0(\mathcal{P}_\alpha) \quad ,$$

ce qui prouve le lemme. ■

Le théorème suivant résulte de la combinaison des lemmes précédents.

Théorème 3.1 *Le comportement d'un SRPI avec \mathcal{N} utilisateurs est décrite par les inégalités d'état suivantes, pour tout utilisateur i :*

$$x_i(t) \leq v_i(t - \tau_i(t)) + \mathcal{M}_0(\mathcal{P}_i) \quad , \tag{3.2}$$

où $\tau_i(t)$ est la durée variable pendant laquelle l'utilisateur i détient une ressource (cette durée dépend du comportement interne de l'utilisateur, décrit par \mathcal{G}_i) et où chaque $v_i(\cdot)$ doit être choisi de telle manière que, pour tout t :

$$\sum_{i=1}^{\mathcal{N}} (v_i(t) - x_i(t - \tau_\alpha)) \leq \mathcal{M}_0(\mathcal{P}_\alpha) \quad . \tag{3.3}$$

Preuve: Il est prouvé, notamment dans [Baccelli *et al.*, 1992], que les inégalités (3.2) décrivent l'ensemble des tirs possibles des transitions. La contrainte (3.3) est directement dérivée du lemme 3.3. Enfin le lemme 3.2 démontre que l'utilisation de (3.2) et (3.3) est compatible avec le comportement d'un SRPI. ■

3.1.3 Écriture du modèle dans l'algèbre des dioïdes

Puisque la description du comportement d'un **SRPI** donnée par le théorème 3.1 est très simple et donne un large aperçu de la modélisation du **PRPI**, il est naturel de se demander si la même description peut être écrite dans un dioïde de séries. Puisqu'il est désormais possible, d'après la proposition 2.1 page 62, de calculer le marquage instantané d'une place à partir des séries de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$, nous sommes en position d'écrire la contrainte (3.2) dans un dioïde de séries, comme le montre le lemme suivant.

Lemme 3.4 (Contrainte dans un dioïde de séries) Soient \mathcal{N} utilisateurs partageant r ressources. Si, à tout instant t , chaque ressource est utilisée par au plus un utilisateur, alors les séries de leurs transitions d'entrée v_i et de sortie x_i satisfont l'inégalité suivante :

$$\bigoplus_{i=1}^{\mathcal{N}} \frac{v_i}{\delta^{\tau_\alpha} x_i} \bullet \preceq \gamma^{\mathcal{M}_0(\mathcal{P}_\alpha)} \delta^* .$$

Preuve: La preuve découle immédiatement de (2.1) et (2.5) page 48 en appliquant l'opérateur \bullet à l'équation (3.2), ce qui permet d'écrire la contrainte pour tout t sous forme de séries de $\gamma^* \mathbb{B}[[\gamma, \delta]]$. ■

En résumé, puisque les équations d'état $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ qui décrivent le comportement d'un **GET** vivant sont bien connues (voir la sous-section 2.1.2 page 49), alors le lemme 3.4 conduit au théorème suivant.

Théorème 3.2 Le comportement d'un **GET** vivant et incluant un **SRPI** avec \mathcal{N} utilisateurs peut être décrit, dans le cadre d'un fonctionnement au plus tôt, par des équations d'état de la forme suivante :

$$\begin{cases} X = AX \oplus BU \oplus SV \quad , \\ Y = CX \quad , \\ \text{avec } \bigoplus_{i=1}^{\mathcal{N}} \frac{\Delta_i^q V}{\delta^{\tau_\alpha} \Delta_i^n X} \bullet \preceq \gamma^{\mathcal{M}_0(\mathcal{P}_\alpha)} \delta^* , \end{cases}$$

où V est le vecteur des transitions d'entrée dans les sections critiques des utilisateurs, X est le vecteur des transitions de sortie et les vecteurs $\Delta_i^{qT} \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]^q$ et $\Delta_i^{nT} \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]^n$ sont définis pour tout i par

$$[\Delta_i]_j = \begin{cases} \mathbf{e} & \text{si } j = i, \\ \varepsilon & \text{sinon.} \end{cases}$$

3.1.4 Exemple : modélisation d'une gare de fret

Nous montrons dans cette sous-section une façon d'utiliser notre modèle générique de **PRPI** dans le domaine des transports ferroviaires.

3.1.4.A Présentation du problème et modélisation

Le problème suivant traite d'une gare de fret dotée de deux voies. Trois types de trains sont susceptibles de passer par cette gare : les trains de type 1 ont un container à charger ; ceux de type 2 doivent en livrer un ; enfin, les trains de type 3 n'ont pas besoin de s'arrêter dans la gare. Les durées associées à ces différentes situations sont données par le tableau 3.2.

Le comportement d'un tel système peut être décrit par un GET combiné avec le modèle Rdp d'un SRPI (en gras), comme le montre la figure 3.3. La sémantique utilisée dans cette figure est donnée par le tableau 3.2.

D'après le théorème 3.2 page précédente, le modèle algébrique du Rdp de la figure 3.3 est donné par les équations suivantes dans $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$:

$$\left\{ \begin{array}{l} X \\ \\ \\ Y \\ \gamma^2 \delta^* \end{array} \right. = \left(\begin{array}{cccccccccccc} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \mathbf{e} & \mathbf{e} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^3 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \delta^2 & \delta^4 & \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{array} \right) X \\
 \oplus \left(\begin{array}{cccc} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \mathbf{e} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{e} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \mathbf{e} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \mathbf{e} \end{array} \right) U \oplus \left(\begin{array}{ccc} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \mathbf{e} & \varepsilon & \varepsilon \\ \mathbf{e} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{e} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \mathbf{e} \end{array} \right) V , \\
 Y = \left(\begin{array}{cccccccccccc} \mathbf{e} & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \mathbf{e} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \mathbf{e} & \varepsilon \\ \varepsilon & \varepsilon & \mathbf{e} & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{array} \right) X , \\
 \gamma^2 \delta^* \preceq \bigcirc_{i=1}^3 \frac{\Delta_i^q V}{\delta^3 \Delta_i^n X} \bullet , \tag{3.4}$$

Place	Marquage	Temporisation (temps de l'opération), en u.t.s
\mathcal{P}_1	la grue de chargement est inutilisée ou indisponible	temps de positionnement de la grue de chargement (1u.t.)
\mathcal{P}_2	les opérateurs apportent un container à la grue de chargement	2u.t.
\mathcal{P}_3	un train de type 1 se place en face de la grue de chargement	4u.t.
\mathcal{P}_4	un train de type 1 quitte la gare	2u.t.
\mathcal{P}_5	la grue charge un container sur un train	2u.t.
\mathcal{P}_6	un train de type 2 se place en face de la grue de déchargement	4u.t.
\mathcal{P}_7	la grue décharge un container	2u.t.
\mathcal{P}_8	un train de type 2 quitte la gare	3u.t.
\mathcal{P}_9	un train de type 3 traverse la gare	3u.t.
\mathcal{P}_α	nombre de voies disponibles	temps de préparation des voies (3u.t.)

TABLEAU 3.2 – Sémantique utilisée pour le modèle Rdp d'une gare de fret.

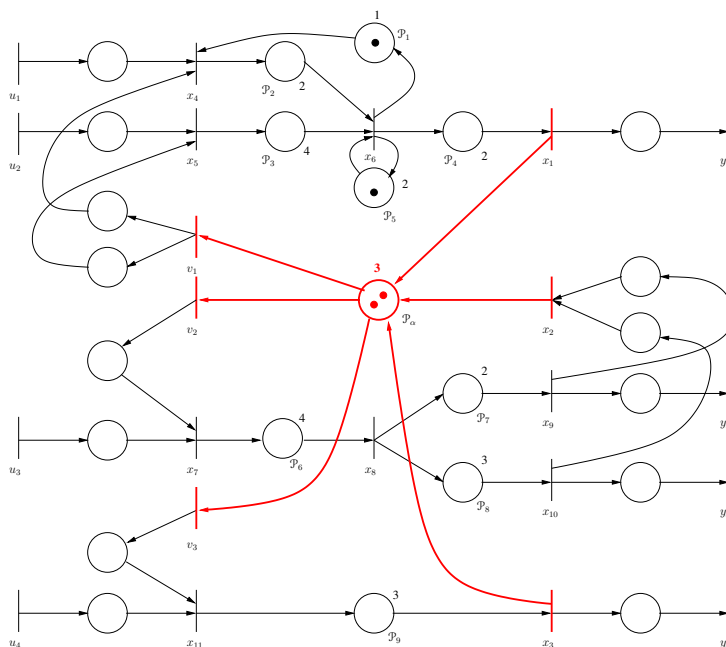


FIGURE 3.3 – Modèle Rdp d'une gare de fret.

où $\Delta_i^{qT} \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^q$ et $\Delta_i^{nT} \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^n$ sont définis pour tout i par

$$[\Delta_i]_j = \begin{cases} \mathbf{e} & \text{si } j = i, \\ \varepsilon & \text{sinon.} \end{cases}$$

L'objectif est de planifier l'arrivée des trains de manière précise, en respectant la disponibilité des voies. La date limite de livraison du container délivré par le train de type 2, ainsi que les dates de départ des trains les plus tardives qui soient admissibles, sont données dans le tableau 3.3.

Train de type 1	Container livré par le train de type 2	Train de type 2	Train de type 3
20	22	19	11
25	–	–	16
–	–	–	21

TABLEAU 3.3 – Dates dues pour la planification des trains dans une gare de fret.

Grâce à ce tableau, nous écrivons immédiatement les trajectoires de tir de l'objectif Z :

$$Z = \begin{pmatrix} \delta^{20} \oplus \gamma\delta^{25} \oplus \gamma^2\delta^* \\ \delta^{22} \oplus \gamma\delta^* \\ \delta^{19} \oplus \gamma\delta^* \\ \delta^{11} \oplus \gamma\delta^{16} \oplus \gamma^2\delta^{21} \oplus \gamma^3\delta^* \end{pmatrix} .$$

Ainsi, dans la suite, nous devons calculer U (arrivées des trains aux abords de la gare) et V (autorisation d'accès aux voies) qui donnent le plus grand Y (départ de la gare) de telle sorte que $Y \preceq Z$, tout en respectant la contrainte (3.4).

3.1.4.B Commande

Nous proposons un algorithme de commande constitué de deux phases : la phase 1 donne les meilleures dates (*i.e.* les plus tardives) sans prendre en considération la partie concernant les ressources partagées, au moyen d'une commande en boucle ouverte [Cottenceau, 1999]. La phase 2 identifie, à partir du système d'équation proposé, les trains qui ne peuvent accéder simultanément à la gare lorsque aucune voie n'est disponible. Afin de choisir le train dont la date de départ doit être avancé le cas échéant, nous utilisons la règle de première date due [Jackson, 1955]. Ainsi, au moins un train est définitivement planifié à chaque itération de la phase 2. Puisque ce train est retiré de la liste des trains considérés pour l'itération suivante, l'algorithme est de complexité $\mathcal{O}(n^2)$, où n est le nombre total de trains à planifier. En effet, dans le pire des cas, à savoir lorsque toutes les dates dues sont égales, nous avons $\sum_{i=1}^{i=n-1} i$ opérations.

Algorithme 3.1**Entrées**

Les équations d'état et d'évolution d'un **SED** incluant un **SRPI**, respectivement $X = f(X, U, V)$ et $Y = g(X)$. Des trajectoires d'entrée U correspondantes.

Sorties

Trajectoires de commandes optimales et de sorties V et Y , respectivement.

Phase 1

1. Initialiser V , U et X :

1.1. Fusionner V et U dans U' :

$$X = AX \oplus B'U' = AX \oplus (B \mid S) \begin{pmatrix} U \\ V \end{pmatrix} ;$$

1.2. Calculer les trajectoires d'entrées idéales (éventuellement irréalisables) $\hat{U}' = CA^*B' \otimes Z$;

1.3. Scinder \hat{U}' en \hat{U} et \hat{V} pour calculer $\hat{X} = A^*(B\hat{U} \oplus S\hat{V})$;

1.4. Initialiser U à \hat{U} , X à \hat{X} et V à \hat{V} . Soit w_i le nombre de trains de type i à planifier : initialiser t à $\max_{i \in [1, \mathcal{N}]} (\mathcal{D}_{[X]_i}(w_i - 1))$.

Phase 2

2. Tant que la contrainte (3.4) est vérifiée à t , t prend pour valeur $t-1$. Si $t = 0$, l'algorithme se termine.

3. Tant que la contrainte (3.4) n'est pas vérifiée, faire :

3.1. Calculer le nombre g de trains à avancer à cette itération :

$$g = \frac{\bigotimes_{i=1}^{\mathcal{N}} \frac{\mathcal{C}_{[V]_i}^{\triangleright}(t)}{\mathcal{C}_{[X]_i}^{\triangleright}(t - \tau_\alpha)}}{r} ;$$

3.2. Construire la liste L des trains qui étaient jusque là prévus pour quitter la gare à la date t : $L = \{l \in \mathcal{N} \mid x_l(t) > x_l(t-1)\}$;

3.3. Construire la liste $L^* \subset L$ des trains à avancer à cette itération :

$$L^* = \left\{ l \in L \mid \mathcal{D}_l = \bigwedge_{i \in L} \frac{\mathcal{D}_{[\hat{X}]_i}(\mathcal{C}_{[X]_i}(t))}{\bigoplus_{j \in L - \{i\}} \mathcal{D}_{[V]_j}(\mathcal{C}_{[V]_j}(t + \tau_\alpha))} \right\} .$$

S'il y a plus de g trains dans cette liste, en choisir g au hasard. S'il y a moins de g trains dans cette liste, choisir des trains de la même manière dans l'ensemble $L - L^*$. Une fois que L^* a g éléments, trier cette liste par valeurs décroissantes de \mathcal{D}_l ;

4. Pour tout $l \in L^*$, faire :

4.1. Considérer le plus grand X tel que $X \preceq Q$ avec Q défini par

$$[Q]_i = \begin{cases} [X]_i \wedge \left(\delta^{\oplus_{j \neq i}} \frac{\mathcal{D}_{[V]_j} (e_{[V]_j}^{(t+\tau\alpha)})}{\tau_\alpha} \oplus \gamma^{1e_{[X]_i}(t)} \delta^* \right) & \text{si } i \in L^*, \\ [X]_i & \text{sinon;} \end{cases}$$

4.2. Calculer les plus grands U et V qui induisent un tel X :

4.2.1. Fusionner à nouveau U et V dans U' , afin de pouvoir calculer plus facilement ces vecteurs. En effet,

$$\begin{aligned} & X \preceq Q, \\ \Leftrightarrow & A^* B' U' \preceq Q, \forall U', \\ \Rightarrow & U' \preceq \frac{Q}{A^* B'}; \end{aligned}$$

4.2.2. Calculer $U' = A^* B' \oslash Q$, puis extraire V et U et calculer le X induit : $X = A^*(BU \oplus SV)$;

4.3. Revenir à l'étape 2.

Cet algorithme permet d'ordonner les trains en suivant la règle de Jackson. Autrement dit, les trains sont planifiés de telle sorte que l'ensemble du processus commence le plus tard possible. La validation de la contrainte (3.4) aux étapes 2 et 3 permet de s'assurer que les conditions (a) à (c) sont bien respectées à chaque itération.

3.1.4.C Application numérique

L'étape 1 initialise les vecteurs d'état avec les valeurs suivantes :

$$\hat{X} = \begin{pmatrix} \delta^{20} \oplus \gamma\delta^{25} \oplus \gamma^2\delta^* \\ \delta^{19} \oplus \gamma\delta^* \\ \delta^{11} \oplus \gamma\delta^{16} \oplus \gamma^2\delta^{21} \oplus \gamma^3\delta^* \\ \delta^{16} \oplus \gamma\delta^{21} \oplus \gamma^2\delta^* \\ \delta^{14} \oplus \gamma\delta^{19} \oplus \gamma^2\delta^* \\ \delta^{18} \oplus \gamma\delta^{23} \oplus \gamma^2\delta^* \\ \delta^{12} \oplus \gamma\delta^* \\ \delta^{16} \oplus \gamma\delta^* \\ \delta^{18} \oplus \gamma\delta^* \\ \delta^{19} \oplus \gamma\delta^* \\ \delta^7 \oplus \gamma\delta^{12} \oplus \gamma^2\delta^{17} \oplus \gamma^3\delta^* \\ \delta^9 \oplus \gamma\delta^{14} \oplus \gamma^2\delta^{19} \oplus \gamma^3\delta^* \end{pmatrix},$$

$$\hat{V} = \begin{pmatrix} \delta^{14} \oplus \gamma\delta^{19} \oplus \gamma^2\delta^* \\ \delta^{12} \oplus \gamma\delta^* \\ \delta^7 \oplus \gamma\delta^{12} \oplus \gamma^2\delta^{17} \oplus \gamma^3\delta^* \end{pmatrix},$$

$$\hat{U} = \begin{pmatrix} \delta^{16} \oplus \gamma\delta^{21} \oplus \gamma^2\delta^* \\ \delta^{14} \oplus \gamma\delta^{19} \oplus \gamma^2\delta^* \\ \delta^{12} \oplus \gamma\delta^* \\ \delta^7 \oplus \gamma\delta^{12} \oplus \gamma^2\delta^{17} \oplus \gamma^3\delta^* \end{pmatrix},$$

$$\hat{Y} = \begin{pmatrix} \delta^{20} \oplus \gamma\delta^{25} \oplus \gamma^2\delta^* \\ \delta^{18} \oplus \gamma\delta^* \\ \delta^{19} \oplus \gamma\delta^* \\ \delta^{11} \oplus \gamma\delta^{16} \oplus \gamma^2\delta^{21} \oplus \gamma^3\delta^* \end{pmatrix}.$$

Après six itérations, nous obtenons les valeurs suivantes pour les vecteurs d'état :

$$\begin{aligned}
 X &= \begin{pmatrix} \delta^{16} \oplus \gamma\delta^{25} \oplus \gamma^2\delta^* \\ \delta^{14} \oplus \gamma\delta^* \\ \delta^4 \oplus \gamma\delta^7 \oplus \gamma^2\delta^{21} \oplus \gamma^3\delta^* \\ \delta^{12} \oplus \gamma\delta^{21} \oplus \gamma^2\delta^* \\ \delta^{10} \oplus \gamma\delta^{19} \oplus \gamma^2\delta^* \\ \delta^{14} \oplus \gamma\delta^{23} \oplus \gamma^2\delta^* \\ \delta^7 \oplus \gamma\delta^* \\ \delta^{11} \oplus \gamma\delta^* \\ \delta^{13} \oplus \gamma\delta^* \\ \delta^{14} \oplus \gamma\delta^* \\ \mathbf{e} \oplus \gamma\delta^3 \oplus \gamma^2\delta^{17} \oplus \gamma^3\delta^* \\ \delta^2 \oplus \gamma\delta^5 \oplus \gamma^2\delta^{19} \oplus \gamma^3\delta^* \end{pmatrix}, \\
 V &= \begin{pmatrix} \delta^{10} \oplus \gamma\delta^{19} \oplus \gamma^2\delta^* \\ \delta^7 \oplus \gamma\delta^* \\ \mathbf{e} \oplus \gamma\delta^3 \oplus \gamma^2\delta^{17} \oplus \gamma^3\delta^* \end{pmatrix}, \\
 U &= \begin{pmatrix} \delta^{12} \oplus \gamma\delta^{21} \oplus \gamma^2\delta^* \\ \delta^{10} \oplus \gamma\delta^{19} \oplus \gamma^2\delta^* \\ \delta^7 \oplus \gamma\delta^* \\ \mathbf{e} \oplus \gamma\delta^3 \oplus \gamma^2\delta^{17} \oplus \gamma^3\delta^* \end{pmatrix}, \\
 Y &= \begin{pmatrix} \delta^{16} \oplus \gamma\delta^{25} \oplus \gamma^2\delta^* \\ \delta^{13} \oplus \gamma\delta^* \\ \delta^{14} \oplus \gamma\delta^* \\ \delta^4 \oplus \gamma\delta^7 \oplus \gamma^2\delta^{21} \oplus \gamma^3\delta^* \end{pmatrix}.
 \end{aligned}$$

D'après V et X nous pouvons extraire la planification des trains qui est présentée par le tableau 3.4.

Train de type 1		Train de type 2		Train de type 3	
Arrivée	Départ	Arrivée	Départ	Arrivée	Départ
10	16	7	14	0	4
19	25	–	–	3	7
–	–	–	–	17	21

TABEAU 3.4 – Tableau de planification final d'une gare de fret.

Il est aisé de construire un *diagramme de Gantt* [Gantt, 1919] d'après les valeurs définitives de X et V , voir figure 3.4.

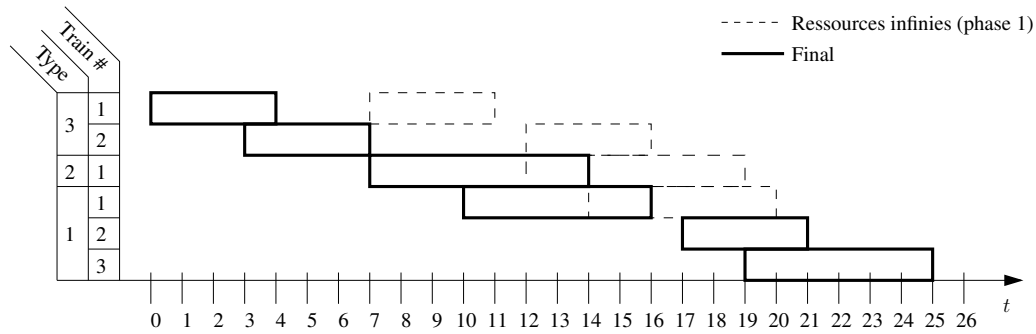


FIGURE 3.4 – Diagramme de Gantt d’une gare de fret.

Nous portons à l’attention du lecteur le fait que le théorème 3.2 page 82 est exploité directement dans l’algorithme aux étapes 2 et 3. En conséquence de quoi, les trois conditions de la sous-section 3.1.2 page 79 sont respectées :

- nous pouvons vérifier dans la figure 3.4 que la contrainte (3.4) empêche un train de s’engager sur une voie si elle n’est pas disponible (condition (a)). En effet, si nous avons un nombre infini de voies (en pointillés), trois trains se trouveraient dans la gare sur les périodes $t \in [14, 16]$ et $t \in [17, 20]$, ce qui est impossible. La solution proposée par l’algorithme (en gras) ne contient pas cette erreur ;
- dans la figure 3.5, nous pouvons voir que les voies sont occupées en permanence, ce qui signifie que la contrainte (3.4) permet que chaque voie soit utilisée dès qu’elle est disponible (condition (b)). Plus précisément, les périodes de préparation de la voie et les périodes d’occupation par un train se succèdent sans temps mort (pas de partie non grisée) ;
- enfin, nous constatons par l’intermédiaire de la figure 3.6 que lorsqu’un train quitte la gare, il libère une voie (condition (c)). Nous observons sur cette figure que, pour chacune des deux voies, tous les départs de trains correspondent bien à un changement de l’état « occupée » à l’état « indisponible ».

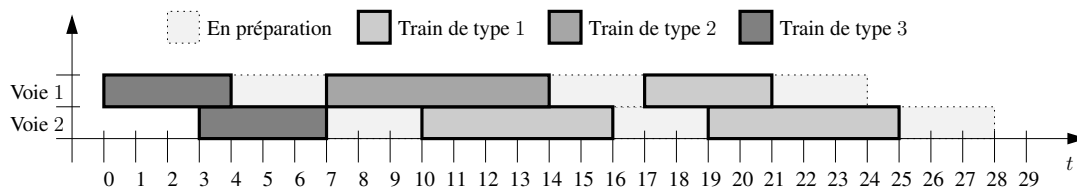


FIGURE 3.5 – Occupation des voies d’une gare de fret.

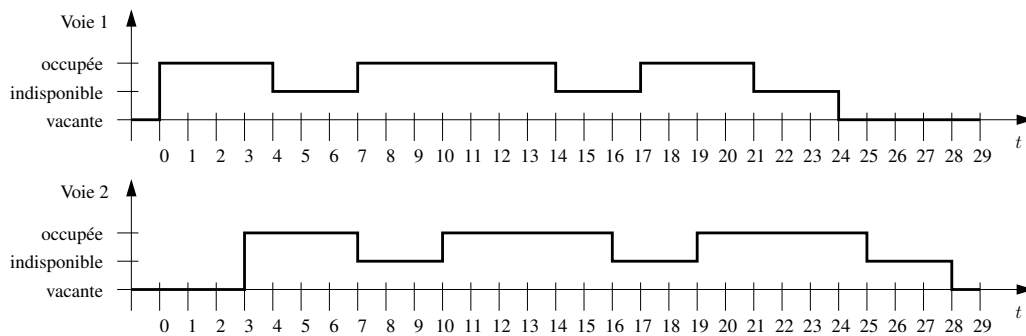


FIGURE 3.6 – État d'utilisation des voies d'une gare de fret.

3.2 Modélisation et commande d'un carrefour simplifié

Nous appliquons ici le modèle présenté précédemment au cas qui nous concerne, à savoir une intersection de rues. Nous faisons ici plusieurs suppositions sur l'intersection. Nous partons du principe que les segments de rue en contact avec la zone de conflit ne possèdent qu'une voie et que la vitesse est constante dans cette zone. En particulier, les variations de vitesse dues aux arrêts à hauteur des feux ou aux fluctuations de trafic ne sont pas prises en compte. Un modèle plus spécifique est ainsi développé, suivi d'une commande à horizon fixe. Il s'agit dans ce cas de planifier le passage des véhicules à l'intérieur d'une fenêtre de temps qui commence par la présence d'un véhicule dans le segment et se termine par le temps minimum nécessaire à un véhicule observé pour parvenir au bout de l'intersection. Enfin, une autre commande est proposée. Celle-ci permet d'élargir la fenêtre de temps de manière à considérer tous les véhicules présents dans le système étudié.

3.2.1 Modèle d'un carrefour simplifié

Afin de modéliser une intersection complète par un RdP, chaque rue est traitée séparément et discrétisée en segments dont les variables caractéristiques dépendent seulement du temps et non pas de la position des véhicules dans le segment (voir section 2.2 page 58). De ce fait, les deux rues ont un segment en commun, qui constitue la zone de conflit en elle-même (voir figure 1.2 page 26). Nous prenons en considération ses dimensions (les longueurs partagées peuvent être différentes pour chaque rue) et le débit maximum. La densité est naturellement limitée dans l'intersection puisque nous n'autorisons, par hypothèse, qu'une voiture à la fois dans la surface partagée.

3.2.1.A Modélisation par RdP

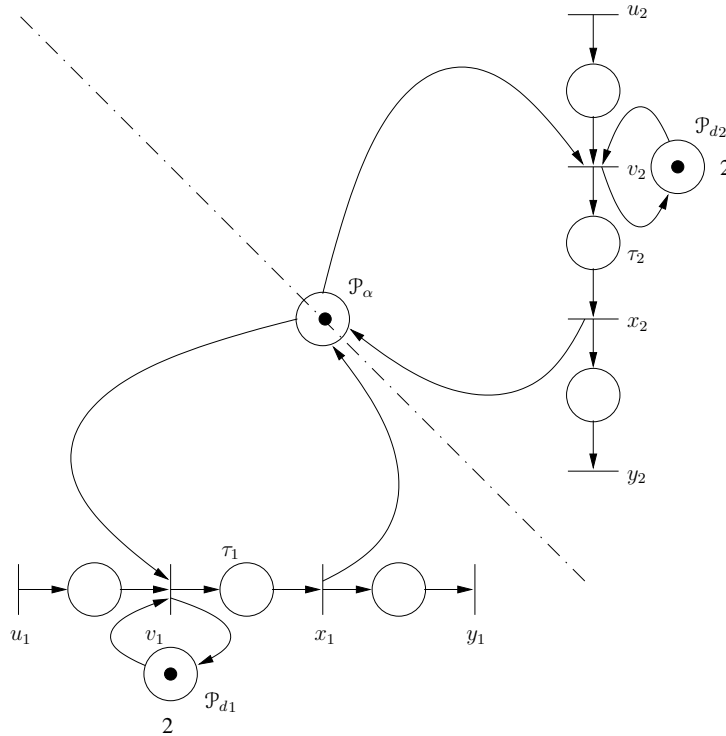


FIGURE 3.7 – Modèle RdP d'une intersection élémentaire.

Le modèle de la figure 3.7 représente le comportement moyen des véhicules circulant au sein d'une intersection élémentaire. Plus précisément, les jetons, qui représentent les voitures, évoluent à travers les chemins délimités par les transitions u_i et y_i , qui représentent chacun une rue i . Les places \mathcal{P}_{d_i} modélisent le débit maximum (un véhicule toutes les deux unités de temps correspond à un débit maximum de $0.5veh.s^{-1}$). La place \mathcal{P}_α représente la surface partagée, sur laquelle ne peut circuler qu'un seul véhicule à la fois. Un jeton est présent dans cette place à une date t ssi l'intersection est vide à cette date. Dans cette représentation, nous pouvons observer les compteurs des transitions. Le tableau 3.5 détaille la sémantique du modèle.

3.2.1.B Modélisation dans le dioïde $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$

Puisque le modèle RdP du système sans intersection est un GET, le modèle RdP de la figure 3.7 est scindé en deux sous-RdPs comme explicité par la figure 3.8. Le premier sous-RdP regroupe les GETs représentant le comportement des véhicules dans les rues hors intersection. Le second, noté \mathcal{G}_α , représente le comportement des véhicules dans la zone de conflit à proprement parler.

Nom	Signification
u_i	transition d'entrée ; $u_i(t)$ est le nombre de voitures entrées dans la rue i jusqu'à la date t
y_i	transition de sortie ; $y_i(t)$ est le nombre de voitures sorties de la rue i jusqu'à la date t
v_i	$v_i(t)$ est le nombre de fois qu'une voiture venant de la rue i s'est engagée dans l'intersection jusqu'à la date t
x_i	$x_i(t)$ est le nombre de fois qu'une voiture venant de la rue i a traversé l'intersection jusqu'à la date t
τ_i	temps de parcours de la surface partagée. Il dépend de la longueur que la rue i partage avec les autres rues et de la vitesse moyenne dans cette rue

TABLEAU 3.5 – Sémantique du modèle de la figure 3.7.

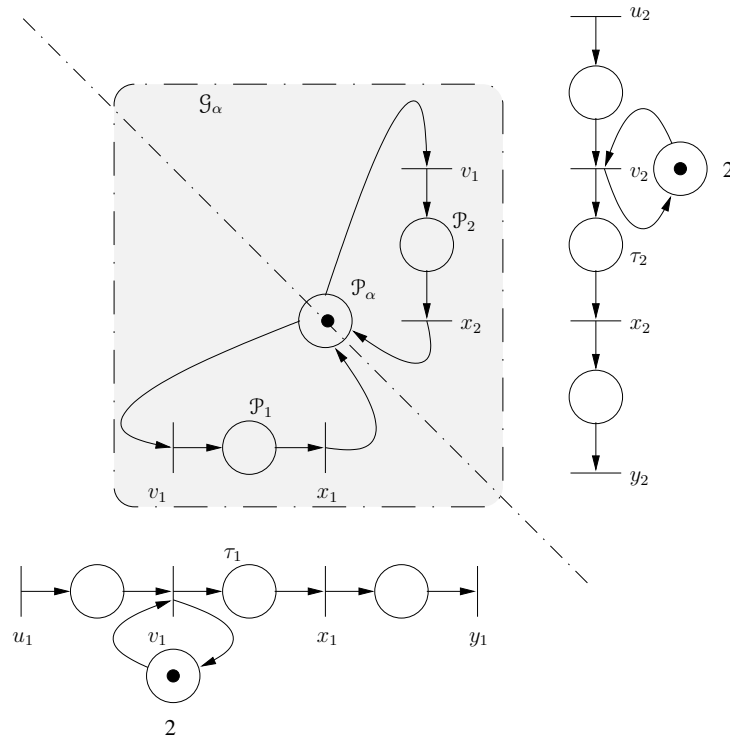


FIGURE 3.8 – Modèle scindé d'une intersection élémentaire.

Afin d'étudier le comportement des voitures évoluant à travers l'intersection, les transitions v_i et x_i sont dupliquées. Ceci permet de concentrer l'analyse sur l'intersection en éliminant l'influence externe des rues. Ainsi, nous considérons que \mathcal{G}_α est le modèle d'une intersection élémentaire.

Le théorème suivant nous fournit un modèle $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ d'une intersection élémentaire entre deux rues [Corréia *et al.*, 2006b].

Théorème 3.3 *Le comportement des véhicules au sein de l'intersection élémentaire de deux*

rues est décrit par les équations d'état suivantes :

$$\begin{cases} X = \begin{pmatrix} \delta^{\tau_1} & \varepsilon \\ \varepsilon & \delta^{\tau_2} \end{pmatrix} V, \\ V \preceq \begin{pmatrix} \gamma\delta^2 & \varepsilon \\ \varepsilon & \gamma\delta^2 \end{pmatrix} V \oplus \begin{pmatrix} \mathbf{e} & \varepsilon \\ \varepsilon & \mathbf{e} \end{pmatrix} U \\ Y = \begin{pmatrix} \mathbf{e} & \varepsilon \\ \varepsilon & \mathbf{e} \end{pmatrix} X, \\ \text{avec } \bigodot_{i=1}^2 \frac{\Delta_i V}{\Delta_i X} \bullet \preceq \gamma\delta^*, \end{cases}$$

où le vecteur $\Delta_i^T \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^n$ est défini pour tout i par

$$[\Delta_i]_j = \begin{cases} \mathbf{e} & \text{si } j = i, \\ \varepsilon & \text{sinon.} \end{cases}$$

$\Delta_i V$ est le compteur de la transition d'entrée (régulée) répertoriant le nombre de véhicules venant de la rue i et qui se sont engagés dans l'intersection et $\Delta_i X$ fournit le nombre de ceux qui en sont sortis.

3.2.1.C Exemple

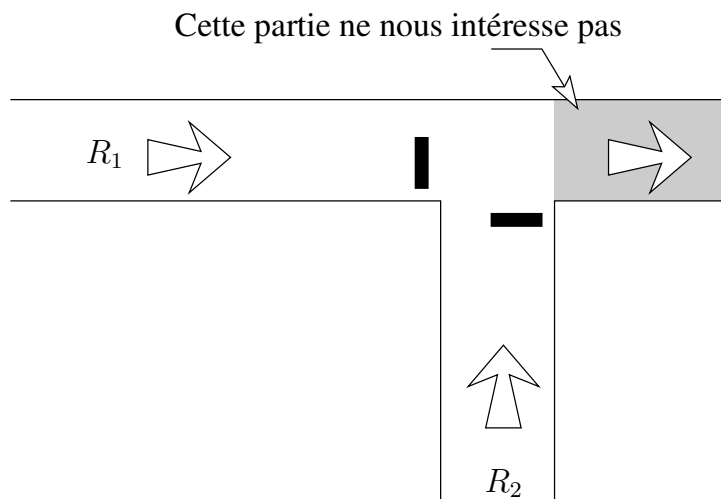


FIGURE 3.9 – Rencontre de deux rues à une intersection élémentaire.

Considérons le carrefour proposé par la figure 3.9. Le tableau 3.6 donne les informations sur la configuration de ce carrefour.

Paramètre	Valeur	
	R_1	R_2
Vitesse moyenne ($km.h^{-1}$)	50	
Longueur (m)	300	400
Nombre de véhicules initialement présents (veh)	12	20
Vitesse moyenne dans l'intersection ($km.h^{-1}$)	40	25
Longueur de la surface partagée (m)	22	25

TABLEAU 3.6 – Informations sur la configuration du carrefour de la figure 3.9.

Afin de modéliser ce système complet, nous combinons le modèle d'intersection élémentaire présenté dans cette section avec le modèle de rue proposé dans la section 2.2 page 58. Le modèle ainsi composé est présenté par la figure 3.10, dans lequel les valeurs des paramètres sont lisibles.

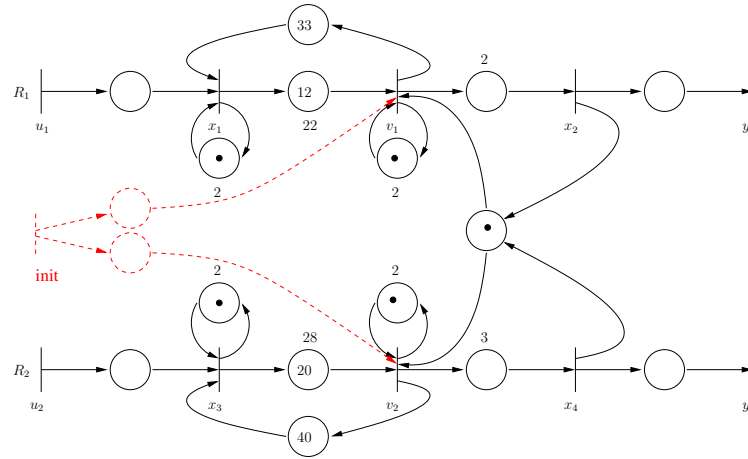


FIGURE 3.10 – Modèle RdP de deux rues se rencontrant à une intersection élémentaire (1u.t.= 1s).

Nous sommes à présent en position d'écrire le modèle analytique du carrefour considéré dans le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. Notons que la pseudo-entrée *init* est introduite comme dernier élément du vecteur U .

$$\begin{aligned}
 X &= \begin{pmatrix} \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} X \oplus \begin{pmatrix} \mathbf{e} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{e} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} U \\
 &\oplus \begin{pmatrix} \varepsilon & \varepsilon \\ \delta^2 & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \delta^3 \end{pmatrix} V, \\
 Y &= \begin{pmatrix} \varepsilon & \mathbf{e} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \mathbf{e} \end{pmatrix} X,
 \end{aligned}$$

sous les contraintes suivantes :

$$\begin{aligned} V &\preceq \begin{pmatrix} \gamma\delta^2 & \varepsilon \\ \varepsilon & \gamma\delta^2 \end{pmatrix} V \oplus \begin{pmatrix} \gamma^{12}\delta^{22} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \gamma^{20}\delta^{28} & \varepsilon \end{pmatrix} X \\ &\oplus \begin{pmatrix} \varepsilon & \varepsilon & \mathbf{e} \\ \varepsilon & \varepsilon & \mathbf{e} \end{pmatrix} U \quad , \\ \gamma\delta^* &\preceq \frac{v_1}{x_2} \bullet \odot \frac{v_2}{x_4} \bullet \quad . \end{aligned}$$

Le modèle proposé ne fait que définir l'espace des comportements admissibles par le système. Nous devons désormais nous atteler à réduire cet espace en ne gardant que les meilleurs comportements, ceux qui permettent aux véhicules de passer le moins de temps possible dans le système. Pour ce faire, nous devons trouver une commande de type poursuite de trajectoire, qui doit être appliquée au vecteur V .

3.2.2 Commande par fenêtres de temps

Puisque nous considérons des rues à une seule voie, nous proposons une commande qui exploite au maximum cette propriété. Depuis chaque rue, un véhicule ne peut s'engager dans la zone de conflit avant son prédécesseur. Ainsi, nous connaissons exactement le temps minimum nécessaire à chaque véhicule pour quitter l'intersection dès l'instant où il s'est engagé dans la rue. Autrement dit, nous pouvons calculer toutes les dates idéales (*i.e.* minimales) de sortie de chaque véhicule venant de chaque rue. Ce problème est similaire au problème de planification d'une machine seule.

Dans la notation $\alpha|\beta|\gamma$ utilisée habituellement dans la littérature pour décrire les problèmes de planification (voir [Graham *et al.*, 1979, Brucker et Knust, 1999]), notre problème peut être écrit comme suit : $1|p_i = 1; chains(l_{ij} = l); r_i|L_{\max}$. La surface partagée fait office de « machine » pour le problème de planification ainsi défini ; elle est évidemment unique, d'où le début de la notation, « 1 ». Dans la partie β , $p_i = 1$ signifie que la ressource est toujours utilisée pendant 1 u.t., ceci est une hypothèse réaliste à notre sens. En effet, les voitures ont en général besoin de deux secondes pour traverser la zone de conflit. Ainsi, il suffit de considérer qu'une u.t. vaut 2 secondes. $chains(l_{ij} = l)$ exprime le fait que nous avons affaire à une chaîne de véhicules pour chaque rue. r_i est ajouté pour spécifier que les véhicules ont besoin d'un certain temps pour arriver à hauteur du feu et donc ne sont pas nécessairement « disponibles » dès le début de la simulation. Enfin, L_{\max} clôt la description puisque nous souhaitons minimiser le retard maximum de chaque véhicule par rapport à sa date de sortie idéale. S'il n'y a pas de contrainte temporelle soulevée, par exemple, par la présence d'un véhicule prioritaire dans la file d'attente, le problème devient C_{\max} , dans lequel l'objectif est de vider les zones de stockage le plus rapidement possible. Ainsi, nous pouvons trier les véhicules par dates de sortie croissantes, afin de minimiser le retard global maximal. Pour réaliser cet objectif, nous avons proposé dans [Corréia *et al.*, 2006a] d'utiliser l'algorithme de [Lawler, 1977] (voir en annexe).

3.2.2.A Description de la commande

Nous optimisons la commande des feux à l'intérieur des fenêtres de temps, lesquelles commencent avec l'arrivée d'un véhicule et se terminent par le temps minimum nécessaire à un véhicule observé pour parvenir au bout de l'intersection. Soient A, B, C, E, F, L, Q les matrices d'état définies de manière appropriée. Pour chaque fenêtre de temps, notre algorithme se compose des étapes suivantes :

1. calculer l'état courant X avec la valeur de la commande V obtenue lors de l'optimisation de la fenêtre de temps précédente, ou $V = \varepsilon$ pour la première fenêtre (tous les feux sont rouges) :

$$X = A^*(BU \oplus EV) \quad ;$$

2. calculer V sans les contraintes représentant l'intersection mais avec un retour d'état permettant de s'assurer qu'au maximum une seule voiture peut entrer dans l'intersection pour chaque rue (tous les feux sont verts) :

$$\hat{V} = F^*(LX \oplus QU) \quad ;$$

3. recalculer X avec ce V idéal (dates au plus tôt auxquelles chaque rue peut quitter le carrefour) :

$$\hat{X} = A^*(BU \oplus E\hat{V}) \quad ;$$

4. trier les véhicules à l'aide de l'algorithme de [Lawler, 1977] pour minimiser le retard maximum par rapport aux dates de sorties idéales (ce qui donne les dates de tirs des transitions v_i jusqu'à ce que tous les véhicules considérés aient quitté le carrefour). Ceci permet de prendre en compte la contrainte et retourne les vecteurs X et V qui traduisent un ordonnancement dans lequel le temps d'attente maximum est minimisé pour les véhicules circulant dans la fenêtre de temps considérée ;
5. recommencer au point 1 avec les véhicules arrivés entre temps (*i.e.* saut vers la prochaine fenêtre de temps).

3.2.2.B Exemple

Reprenons l'exemple de la figure 3.9 page 94, avec des taux d'arrivées de véhicules de $0.25veh.s^{-1}$ et $0.2veh.s^{-1}$ pour les rues R_1 et R_2 , respectivement. Le tableau 3.7 présente les résultats de simulation obtenus pour la première fenêtre de temps, où le véhicule 0 est le premier véhicule de la file d'attente de chaque rue (en considérant les véhicules initialement présents).

La première fenêtre de temps utilisée pour l'optimisation commence avec le véhicule 0 de chaque rue, puisqu'aucun véhicule n'a quitté le carrefour; elle se termine à la date $\min(22, 28) = 22s$.

La commande proposée ici est optimale localement, puisqu'elle ne permet pas d'appréhender l'ensemble complet des véhicules lors de l'optimisation. En continuant notre investigation dans la littérature des algorithmes de planification, nous pouvons réaliser un meilleur algorithme de commande. Ceci est l'objectif de la sous-section suivante.

Véhicule	date d'arrivée		date de sortie idéale		date de sortie optimale		retard	
	R_1	R_2	R_1	R_2	R_1	R_2	R_1	R_2
0	$-\infty$	$-\infty$	0	0	45	0	45	0
1	$-\infty$	$-\infty$	2	3	47	3	45	0
2	$-\infty$	$-\infty$	4	6	49	6	45	0
3	$-\infty$	$-\infty$	6	9	51	9	45	0
4	$-\infty$	$-\infty$	8	12	56	12	48	0
5	$-\infty$	$-\infty$	10	15	61	15	51	0
6	$-\infty$	$-\infty$	12	18	63	18	51	0
7	$-\infty$	$-\infty$	14	21	68	21	54	0
8	$-\infty$	$-\infty$	16	24	73	24	57	0
9	$-\infty$	$-\infty$	18	27	75	27	57	0
10	$-\infty$	$-\infty$	20	30	80	30	60	0
11	$-\infty$	$-\infty$	22	33	85	33	63	0
12	0	$-\infty$	24	36	87	36	63	0
13	—	$-\infty$	—	39	—	39	—	0
14	—	$-\infty$	—	42	—	42	—	0
15	—	$-\infty$	—	45	—	53	—	8
16	—	$-\infty$	—	48	—	58	—	10
17	—	$-\infty$	—	51	—	65	—	14
18	—	$-\infty$	—	54	—	70	—	16
19	—	$-\infty$	—	57	—	77	—	20
20	—	0	—	60	—	82	—	22
21	—	20	—	63	—	89	—	26

TABLEAU 3.7 – Résultats de simulation concernant la première fenêtre de temps.

3.2.3 Commande optimale

Dans la sous-section précédente, nous avons vu que le problème de planification des feux peut être ramené à un problème de planification d'une machine seule. Ici encore, nous allons exploiter le fait que chaque rue n'a qu'une seule voie, en amenant le problème dans le cadre des algorithmes de planification les plus récents de la littérature.

3.2.3.A Description de la commande

Dans cette partie nous considérons le même problème, en proposant une nouvelle commande. En effet, ce problème est un cas particulier d'un des nombreux problèmes résolus de manière optimale par l'algorithme proposé dans [Leung *et al.*, 2001]. Puisque cet algorithme donne la solution optimale en un temps polynomial, nous pouvons recalculer la planification optimale à chaque arrivée d'un véhicule dans le système. Ce problème est suffisamment petit

pour considérer une application en temps réel. L'intérêt principal par rapport à la commande de la sous-section précédente est que nous ne considérons pas seulement les véhicules entrés dans une certaine fenêtre de temps, mais bien tous les véhicules présents dans le système. Ceci permet évidemment de calculer une commande plus précise, puisque nous avons une vue globale de l'ensemble des véhicules à trier.

Nous avons adapté l'algorithme pour interagir avec notre modèle dioïde [Corréia *et al.*, 2007a]. Voici les principales étapes :

1. calculer les dates de tirs idéales (ne pouvant pas être atteintes) pour les transitions v_i , en ignorant les contraintes :

$$\hat{V} = D^*(EX \oplus FU) \quad ,$$

avec D, E, F des matrices d'état définies de manière appropriée ;

2. calculer les « dates de disponibilité modifiées » au sens spécifié dans [Leung *et al.*, 2001]. Plus précisément, chaque « date de disponibilité modifiée » $d_{r_i}(k)$ de la rue i concernant le k^{e} véhicule correspondent aux dates décrites par $[\hat{V}]_i$ espacées d'au moins le TIV minimum T_i plus le temps de traversée de l'intersection τ_i :

$$d_{r_i}(k) = \max(\mathcal{D}_{[\hat{V}]_i}(k), d_{r_i}(k-1) + T_i + \tau_i) \quad ,$$

avec $d_{r_i}(0) = \mathcal{D}_{[\hat{V}]_i}(0)$;

3. calculer les « dates dues modifiées » au sens spécifié dans [Leung *et al.*, 2001]. Les « dates dues modifiées » sont des dates arbitraires utilisées pour classer les véhicules en fonction de leurs temps d'attentes. Soit un nombre N arbitrairement grand, chaque « date due modifiée » $d_{d_i}(k)$ correspondant au k^{e} véhicule de la rue i est calculé comme suit :

$$d_{d_i}(k) = N - (\mathcal{M}_0(\mathcal{P}_{q_i}) + \mathcal{C}_{[U]_i}(\mathcal{T}) - k) \times (T_i + \tau_i) \quad ,$$

avec \mathcal{P}_{q_i} la place représentant la queue dans la rue i et \mathcal{T} l'horizon (temporel) de la simulation ;

4. mettre à zéro l'indice de date courante t . Tant que tous les véhicules n'ont pas quitté le système, faire :
 - 4.1. construire une liste des rues « éligibles » c'est-à-dire des rues qui n'ont pas été choisies à $t-1$ (car le débit maximum est d'un véhicule toutes les deux secondes) et ont au moins un véhicule dont la « date de disponibilité modifiée » est inférieure ou égale à t ;
 - 4.2. choisir, dans cette liste, la rue s qui a le véhicule dont la « date due modifiée » est la plus proche ;
 - 4.3. compléter le vecteur Y et toutes les séries x_i et v_i où $i \neq s$ avec les monômes correspondants à la date t ;
 - 4.4. compléter les séries v_s en considérant les contraintes, puis compléter les séries x_s ;
 - 4.5. incrémenter t .

Cet algorithme a été testé avec le logiciel Scilab (voir code source en annexe).

3.2.3.B Exemple

Reprenons une fois encore l'exemple de la figure 3.9 page 94. Puisque le temps moyen nécessaire aux véhicules pour traverser la zone de conflit est de 2 secondes, nous prenons cette durée comme **u.t.** Ainsi, les temporisations de toutes les places du modèle de la figure 3.10 page 95 doivent être divisées par 2 (arrondies par valeur inférieure). Bien entendu, cette **u.t.** est également appliquée aux fonctions compteurs décrivant les tirs des transitions. À toutes fins utiles, nous donnons le **RdP** avec les nouvelles temporisations dans la figure 3.11.

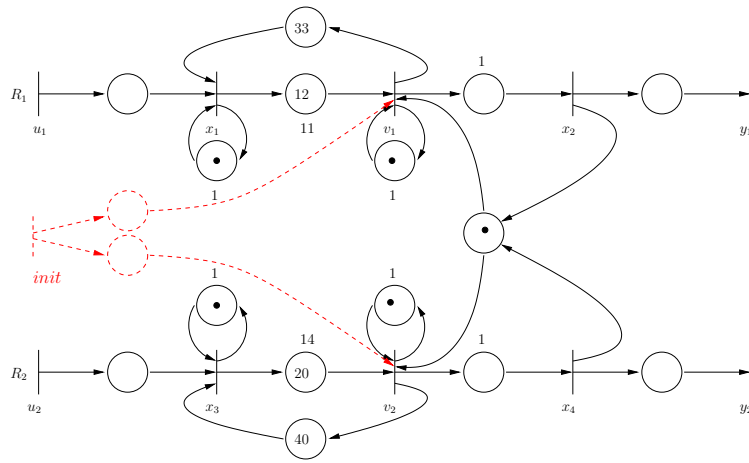


FIGURE 3.11 – Modèle **RdP** de deux rues se rencontrant à une intersection élémentaire (1**u.t.** = 2s).

Nous sommes à présent en position d'écrire le modèle analytique du carrefour considéré dans le dioïde $\mathcal{M}_{in}^{ax}[\gamma, \delta]$. Comme pour le modèle précédent, la pseudo-entrée *init* est introduite comme dernier élément du vecteur U .

$$\begin{aligned}
 X &= \begin{pmatrix} \gamma\delta & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \gamma\delta & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} X \oplus \begin{pmatrix} \mathbf{e} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{e} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} U \\
 &\oplus \begin{pmatrix} \varepsilon & \varepsilon \\ \delta & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \delta \end{pmatrix} V, \\
 Y &= \begin{pmatrix} \varepsilon & \mathbf{e} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \mathbf{e} \end{pmatrix} X,
 \end{aligned}$$

sous les contraintes suivantes :

$$\begin{aligned}
 V &\preceq \begin{pmatrix} \gamma\delta & \varepsilon \\ \varepsilon & \gamma\delta \end{pmatrix} V \oplus \begin{pmatrix} \gamma^{12}\delta^{11} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \gamma^{20}\delta^{14} & \varepsilon \end{pmatrix} X \\
 &\oplus \begin{pmatrix} \varepsilon & \varepsilon & \mathbf{e} \\ \varepsilon & \varepsilon & \mathbf{e} \end{pmatrix} U \quad , \\
 \gamma\delta^* &\preceq \frac{v_1}{x_2} \bullet \odot \frac{v_2}{x_4} \bullet \quad .
 \end{aligned}$$

Pour réaliser la simulation, nous avons considéré des séquences de tirs des transitions d'entrées aléatoires pour chacune des rues. Les résultats sont donnés par la figure 3.12. Dans le cas traité, nous observons que les temps d'attente maximaux des véhicules circulant dans les rues R_1 et R_2 sont respectivement de 22 et 54 secondes ; le processus complet d'évacuation se termine en 160 secondes.

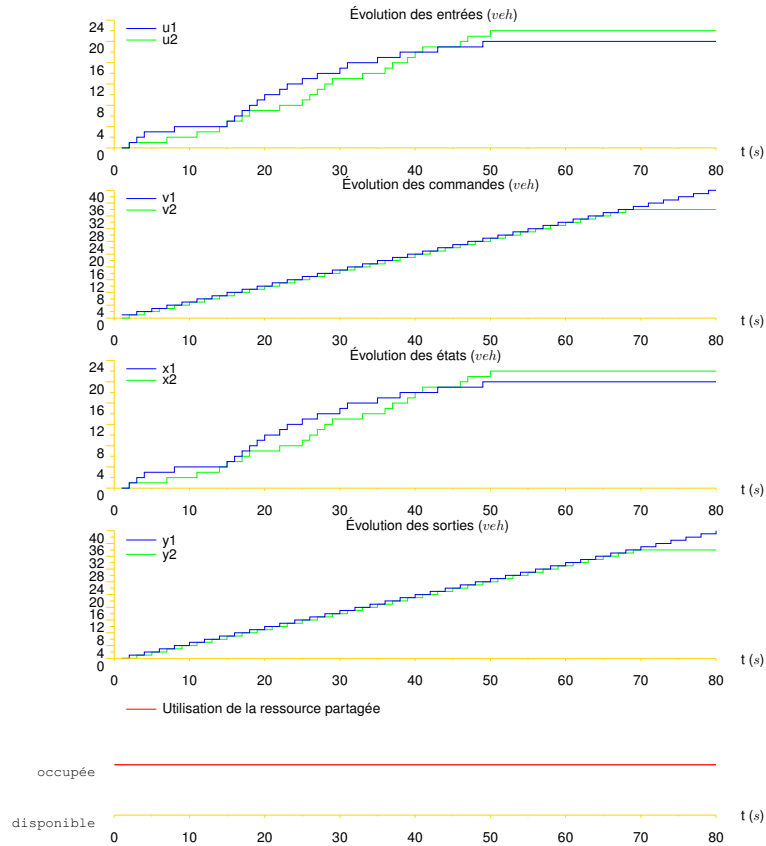


FIGURE 3.12 – Simulation d'un carrefour élémentaire (1u.t.= 2s).

Comme nous l'avons vu dans la sous-section 2.2.2.C page 63, un point du diagramme fondamental est obtenu en simulant le système avec des débits d'entrée constants et égaux pour chaque rue sur une période suffisamment longue. En augmentant progressivement ce débit à travers vingt simulations successives, nous pouvons construire par extrapolation les diagrammes fondamentaux de la figure 3.13.

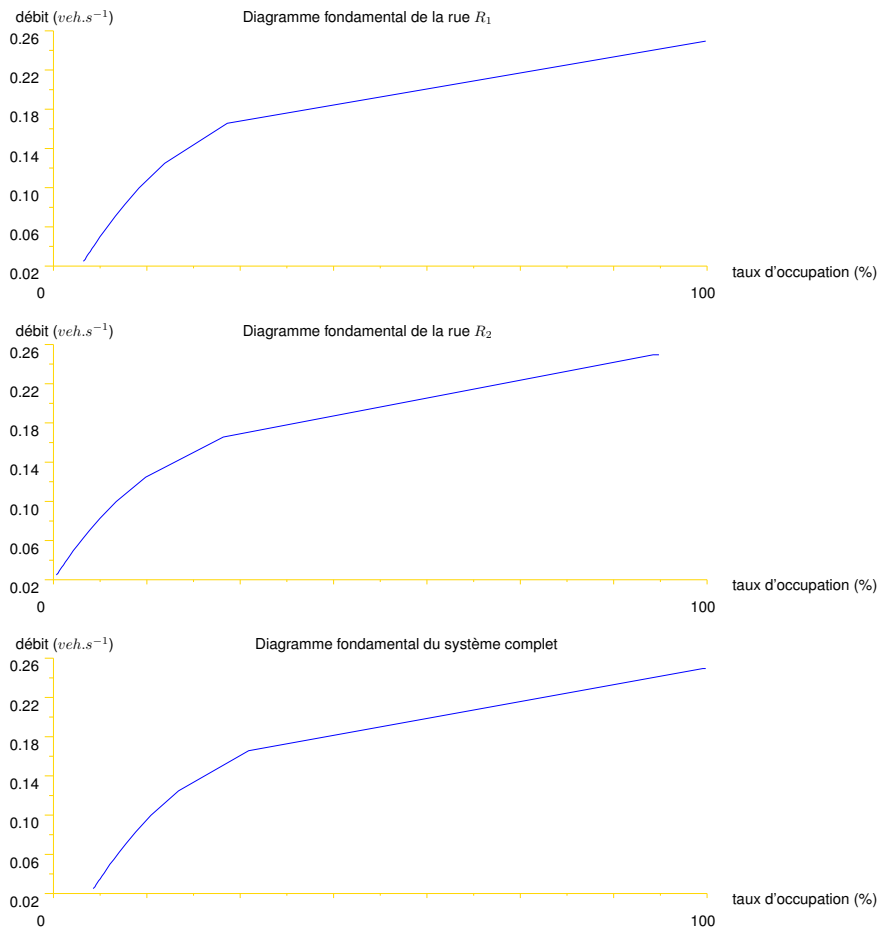


FIGURE 3.13 – Diagramme fondamental d'un carrefour élémentaire.

Le lecteur attentif a sans doute remarqué que la troisième partie (décroissante) du diagramme fondamental, communément observée dans la littérature, n'apparaît pas dans notre cas. Ceci est dû au fait que nous considérons la valeur moyenne du débit pour l'ensemble du processus. C'est pourquoi le phénomène de saturation, qui a lieu quelque part au milieu de la simulation, est absorbé par le régime permanent qui règne partout ailleurs. Néanmoins, ce phénomène apparaît clairement lorsque nous traçons l'évolution de la vitesse en fonction du taux d'occupation, voir figure 3.14.

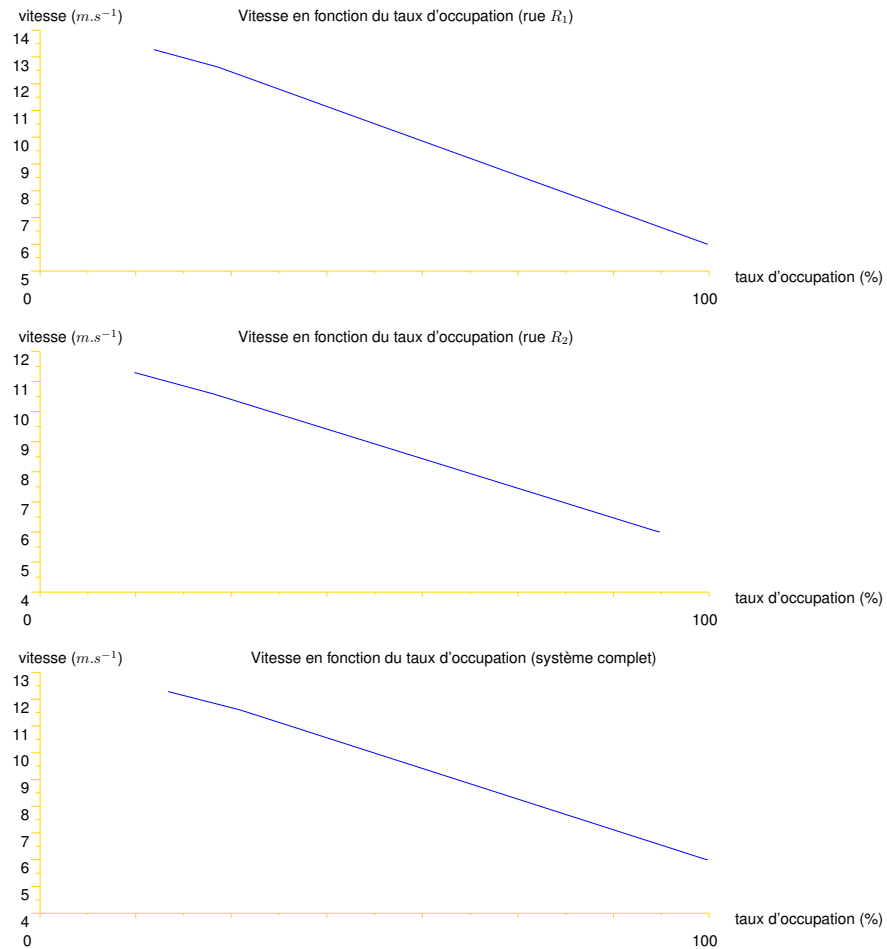


FIGURE 3.14 – Vitesse en fonction du taux d’occupation dans un carrefour élémentaire.

Cette fois, le modèle d’une intersection élémentaire, combiné à celui des rues, est exploité au maximum de ces capacités. Ce modèle peut tout à fait convenir pour modéliser une intersection simplifiée, dans laquelle les variations de vitesse sont négligées. Cet ensemble modulaire peut être commandé par n’importe quelle stratégie de régulation. En particulier, lorsque nous utilisons un algorithme de d’ordonnancement optimal, les performances obtenues en terme de temps de parcours et de longueurs de files d’attente sont tout à fait satisfaisants.

3.3 Étude comparative

L’exemple de système urbain en boucle fermée présenté ici est tiré de [Farhi *et al.*, 2005]. Le système considéré est composé de deux rues circulaires à sens unique et d’un carrefour, tel que schématisé dans la figure 3.15. Lorsque les mobiles quittent le carrefour, un véhicule sur deux tourne, *i.e.* il change de rue.

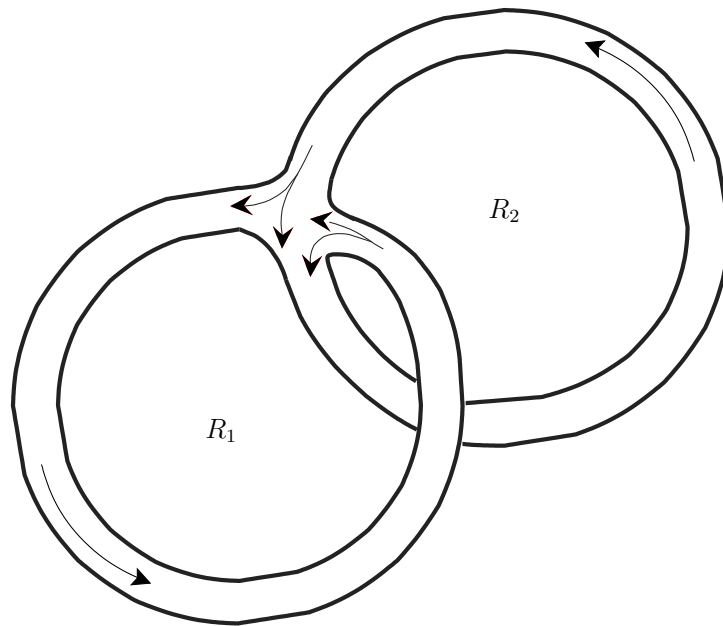


FIGURE 3.15 – Système urbain en boucle fermée composé de deux rues et d’un carrefour.

Dans [Farhi *et al.*, 2005], les auteurs utilisent un modèle **GET** de rue présenté figure 3.16. Dans ce modèle, chaque rue est découpée respectivement en ν et ν' sections capables de contenir un seul véhicule. À chaque section q est associé un couple de places a_q et \bar{a}_q . Si $a_q = 1$, alors $\bar{a}_q = 0$ et la section est occupée par un véhicule. Si $a_q = 0$, alors $\bar{a}_q = 1$ et la section est vide. La transition q correspond à l’entrée de la section q .

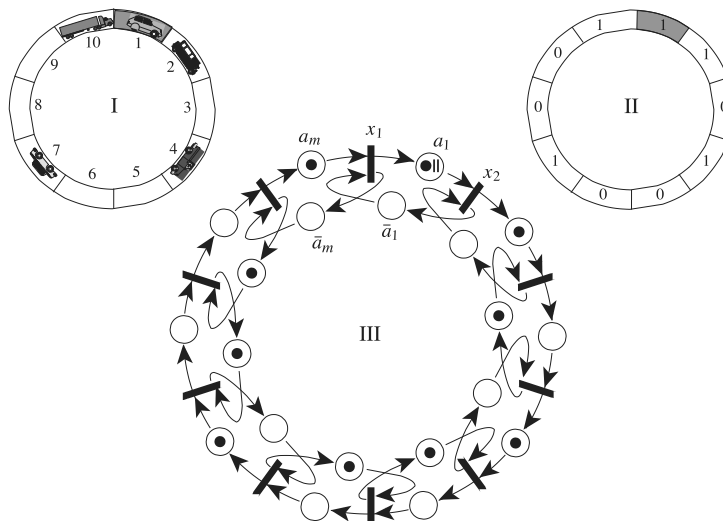


FIGURE 3.16 – Modèle **GET** d’une rue circulaire, présenté dans [Farhi *et al.*, 2005].

Le carrefour est considéré comme une section avec deux entrées et deux sorties, voir figure 3.17. Ce modèle est décrit par une équation de la forme suivante, en utilisant à la fois l’algèbre

classique et l'algèbre $(\min, +)$:

$$X = (XH(\delta)) \otimes D \quad .$$

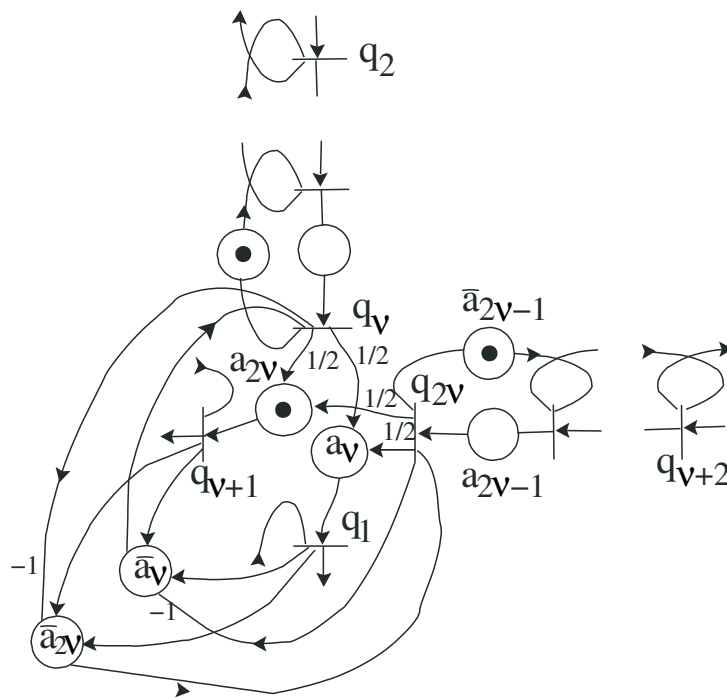


FIGURE 3.17 – Modèle RDP d'un carrefour, présenté dans [Farhi *et al.*, 2005].

Les auteurs fournissent de surcroît une analyse du comportement asymptotique et exploitent au maximum l'aspect cyclique d'un tel système. Bien que nous ne visons pas les mêmes objectifs théoriques et pratiques, nous pouvons valider notre modèle en comparant les analyses de performance telles que le diagramme fondamental. Dans l'exemple numérique de [Farhi *et al.*, 2005] que nous avons choisi pour cette étude comparative, les rues modélisées ont la même capacité, à savoir 25 véhicules. Les auteurs considèrent que les véhicules traversent la zone de conflit en une seconde. Ils obtiennent le diagramme fondamental de la figure 3.18 pour l'ensemble du système, dans le cadre d'un cycle de feux fixe composé de deux phases de même durée.

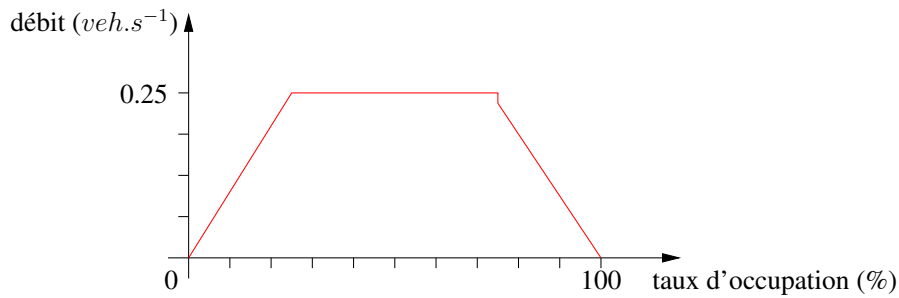


FIGURE 3.18 – Diagramme fondamental d'un système urbain en boucle fermée, présenté dans [Farhi *et al.*, 2005].

Afin de réaliser une comparaison consistante avec les résultats de [Farhi *et al.*, 2005], nous supposons, comme proposé par les auteurs, que les véhicules circulent à la vitesse maximum lorsque cela est possible. Cette vitesse est calculée à partir du débit maximum et de la longueur de chaque rue. Cette longueur est elle-même obtenue à partir de la capacité de la rue et de la densité maximum. Le conflit induit par le carrefour est arbitré par notre algorithme de commande optimale (voir sous-section 3.2.3 page 98). Nous proposons également un modèle différent de celui utilisé dans [Farhi *et al.*, 2005]. Dans la figure 3.19, nous présentons un scénario dans lequel 20 véhicules circulent initialement dans la rue R_1 et 15 dans la rue R_2 .

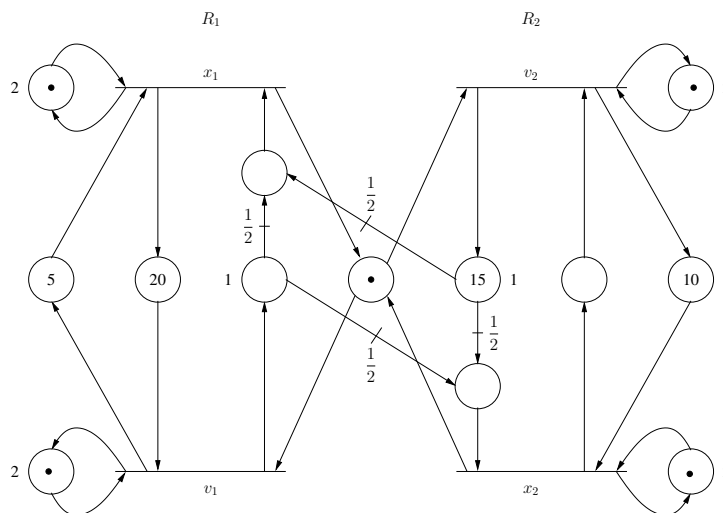


FIGURE 3.19 – Modèle RdP d'un système urbain en boucle fermée.

Le modèle algébrique du RdP de la figure 3.19 est donné dans $\overline{\mathbb{Z}}_{\min}$ par :

$$\begin{aligned} x_1(t) &= 1x_1(t-2) \oplus \left[\frac{1}{2}v_1(t-1) \right] \otimes \left[\frac{1}{2}v_2(t-1) \right] \oplus 5v_1(t) \quad , \\ x_2(t) &= 1x_2(t-2) \oplus \left[\frac{1}{2}v_2(t-1) \right] \otimes \left[\frac{1}{2}v_1(t-1) \right] \oplus 10v_2(t) \quad , \\ v_1(t) &\succeq 1v_1(t-2) \oplus 20x_1(t) \quad , \\ v_2(t) &\succeq 1v_2(t-2) \oplus 15x_2(t) \quad , \end{aligned}$$

sous la contrainte suivante :

$$\gamma\delta^* \preceq \frac{v_1}{x_1} \bullet \odot \frac{v_2}{x_2} \bullet .$$

La simulation de ce scénario donne les résultats présentés par la figure 3.20. Les plus grands temps d'attente calculés pour les rues R_1 et R_2 sont respectivement de 6 et 8 secondes. Nous avons stoppé la simulation après deux cents itérations.

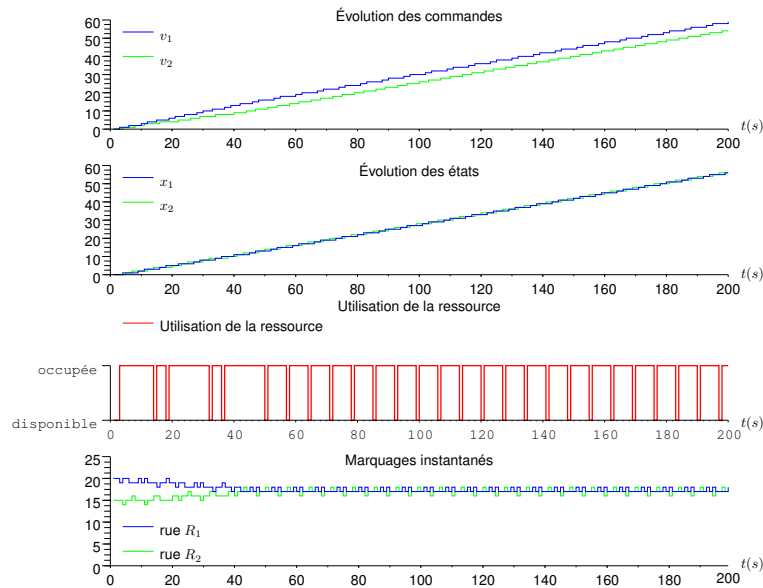


FIGURE 3.20 – Simulation d'un système urbain en boucle fermée.

Pour être en mesure de tracer le diagramme fondamental et la vitesse en fonction du taux d'occupation comme dans l'exemple de la sous-section 3.2.3.B page 100, nous augmentons progressivement le nombre de véhicules présents initialement dans chacune des rues à travers vingt simulations. Ceci nous fournit vingt points de chaque courbe. Le diagramme fondamental obtenu est présenté par la figure 3.21, la vitesse en fonction du taux d'occupation est tracée dans la figure 3.22. Chaque simulation est interrompue au bout de mille itérations.

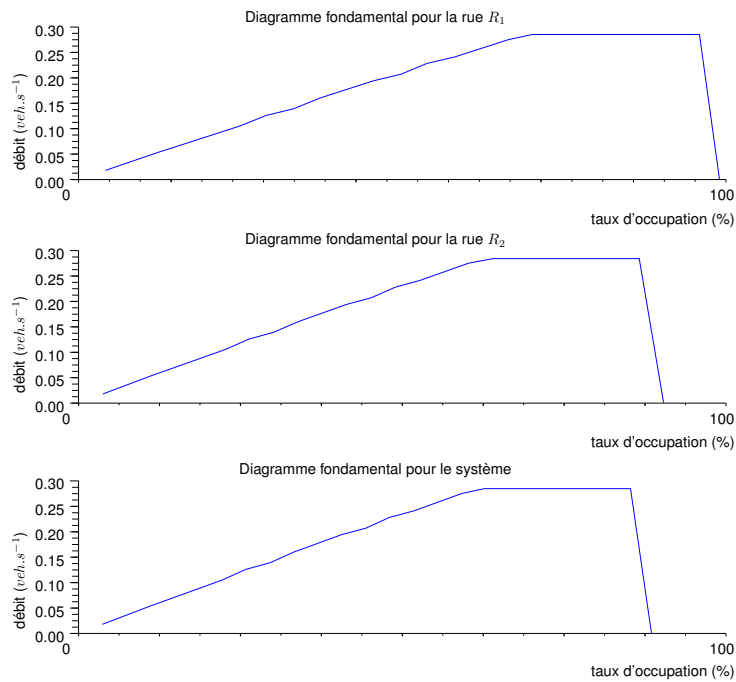


FIGURE 3.21 – Diagramme fondamental d'un système urbain en boucle fermée.

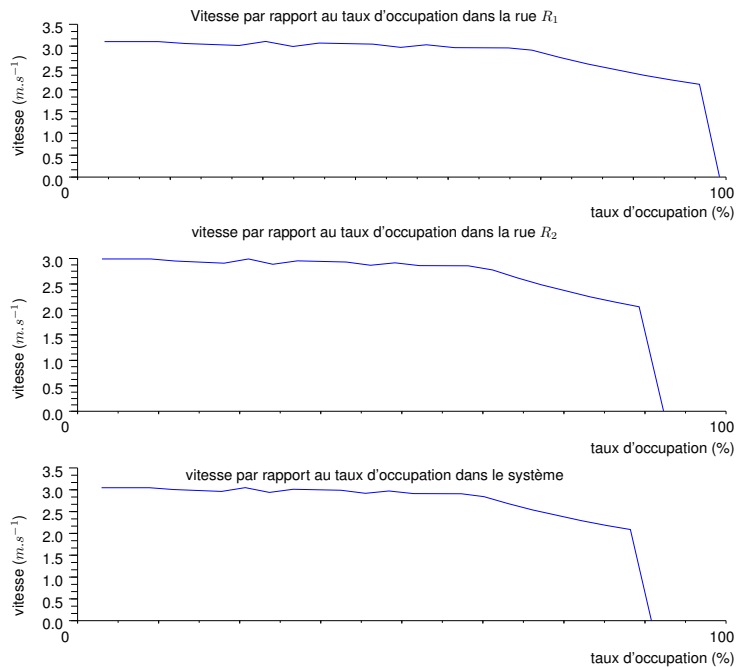


FIGURE 3.22 – Vitesse en fonction du taux d'occupation dans un système urbain en boucle fermée.

Grâce à la simulation, nous pouvons établir que les résultats obtenus sont proches de ceux

présentés dans [Farhi *et al.*, 2005]. En particulier, la partie décroissante du diagramme fondamental est visible. En effet, le dernier point est calculé avec des rues saturées. Ainsi, puisque le système est fermé, aucun véhicule ne peut bouger.

Conclusion

Comme nous le constatons dans la littérature récente, l'utilisation de l'algèbre des dioïdes pour modéliser et commander les systèmes de trafic urbain se heurte à une difficulté majeure. En effet, il est établi que cet algèbre permet de décrire sous forme d'équations linéaires le comportement de GETs. Or, cette classe de RdP ne permet pas de représenter les conflits structurels composant la représentation du problème de ressources partagées. C'est pourquoi nous avons introduit un nouveau modèle générique capable de fournir complètement l'ensemble des solutions admissibles, en-dehors de toute solution particulière.

À l'instar de nombreux articles de la littérature, nous avons proposé un modèle algébrique d'une intersection en considérant plusieurs hypothèses simplificatrices. Le modèle proposé nous a permis d'élaborer une nouvelle approche du problème de la régulation d'un carrefour. En effet, il est assimilé ici à un espace partagé pour lequel nous devons définir un ordonnancement des véhicules le traversant. Deux algorithmes de commande sont ainsi fournis : une commande par fenêtre de temps et une commande optimale prenant en considération tout véhicule présent dans le système. Cette dernière calcule, pour l'ensemble des mobiles présents, l'ordre d'attribution de la zone de conflit.

Basé sur l'algèbre des dioïdes, le comportement du système est décrit par des équations linéaires sous contraintes. Ce modèle, combiné avec le modèle d'une rue présenté au chapitre précédent, nous fournit des évaluations de performance intéressantes, telles que le nombre instantané de véhicules, les bornes de temps d'attente ou encore le diagramme fondamental.

Nous avons cependant constaté que ce modèle ne permet pas d'appréhender les systèmes de trafic urbain actuels de manière réaliste. En effet, les variations de vitesse dues aux arrêts et à plus forte raison les périodes de rouge intégral, pendant lesquelles tous les feux sont rouges, ne sont pas prises en considération. Ainsi, nous devons continuer notre étude pour parvenir à un modèle réaliste de modélisation et de commande des feux de circulation. Ceci est l'objectif du chapitre suivant.

Chapitre

Carrefour contemporain

4.1	Présentation du problème et modélisation	113
4.1.1	Problème de ressources partagées avec temps de configuration	113
4.1.2	Modélisation analytique	115
4.1.3	Écriture du modèle dans l’algèbre des dioïdes	119
4.1.4	Application au trafic urbain	120
4.2	Commande	123
4.2.1	Commande en juste-à-temps	124
4.2.2	Commande optimale par programmation dynamique	126
4.3	Validation	130
4.3.1	Simulation d’un carrefour	130
4.3.2	Simulation de deux carrefours en cascade	134

Introduction

Dans un proche avenir, nous pouvons imaginer que nos voitures seront majoritairement équipées de systèmes de géopositionnement de pointe, tels que le **GPS** ou les caméras, couplés avec des calculateurs de vitesse et de trajectoire. Cependant, ce n'est pas le cas aujourd'hui. Le trafic est actuellement mesuré principalement par les boucles magnétiques et régulé par les feux de circulation. Ainsi, lorsqu'un mobile suit un parcours dans nos villes, sa vitesse varie souvent et fortement.

En effet, les carrefours à feux induisent par nature des comportements de mobiles que tout modèle réaliste doit pouvoir reproduire. Notamment, nous relevons trois points importants : lors d'une période de feu orange, certains véhicules ne peuvent s'arrêter en toute sécurité ; entre deux phases, nous observons systématiquement une période de rouge intégral (tous les feux sont rouges) ; enfin, le véhicule en tête d'une file qui vient de recevoir le feu vert doit accélérer pendant un temps non négligeable avant d'atteindre sa vitesse moyenne.

Afin de prendre en considération ces trois points dans la conception de notre modèle de carrefour contemporain, nous allons, dans un premier temps, enrichir les outils théoriques que nous avons introduits au chapitre précédent. Nous posons ainsi formellement le problème général dont relève le carrefour contemporain. Ceci nous amène à proposer des lois de régulation appropriées.

Dans ce chapitre, nous menons la réflexion permettant de parvenir à un nouveau modèle algébrique d'une intersection avec rouge intégral. Nous présentons ensuite deux commandes optimales pour deux modes de fonctionnement de notre modèle. Enfin, nous validons ce travail à travers deux exemples concrets d'application au trafic urbain.

4.1 Présentation du problème et modélisation

4.1.1 Problème de ressources partagées avec temps de configuration

Afin de marquer les différences avec les problèmes abordés dans le chapitre précédent, nous définissons formellement la famille de problèmes que nous traitons dans cette section.

Définition 4.1 *Un Système à Ressources Partagées Invariantes avec Temps de Configuration (SRPITC) est un SRPI pour lequel les ressources nécessitent, lors de tout changement d'utilisateur, un temps de configuration préalable. Plus précisément, pour toute paire (i, j) définissant un changement de l'utilisateur i vers l'utilisateur j , le délai correspondant est constant et connu à l'avance.*

Les **SEDs** incluant un ou plusieurs **SRPITCs** se retrouvent dans de nombreux domaines d'applications tels que les chaînes de production et de livraison (domaine industriel), les systèmes temps-réel (domaine informatique) et bien sûr les systèmes ferroviaires et routiers (domaine du transport). Lorsqu'un tel système ne contient pas par nature une règle stricte d'attribution des ressources aux utilisateurs, nous nous retrouvons face à un *Problème de Ressources Partagées Invariantes avec Temps de Configuration (PRPITC)*. En effet, nous tenons à traiter ce

problème au sens large en définissant strictement l'espace de solutions. D'un point de vue théorique, il convient de modéliser le problème en-dehors de toute solution particulière. Puisque le problème traité inclut des conflits, nous ne pouvons pas le modéliser à l'aide d'un GET. Ainsi, le modèle graphique générique d'un SRPITC présenté par la figure 4.1 appartient à une classe plus générale de RdPs.

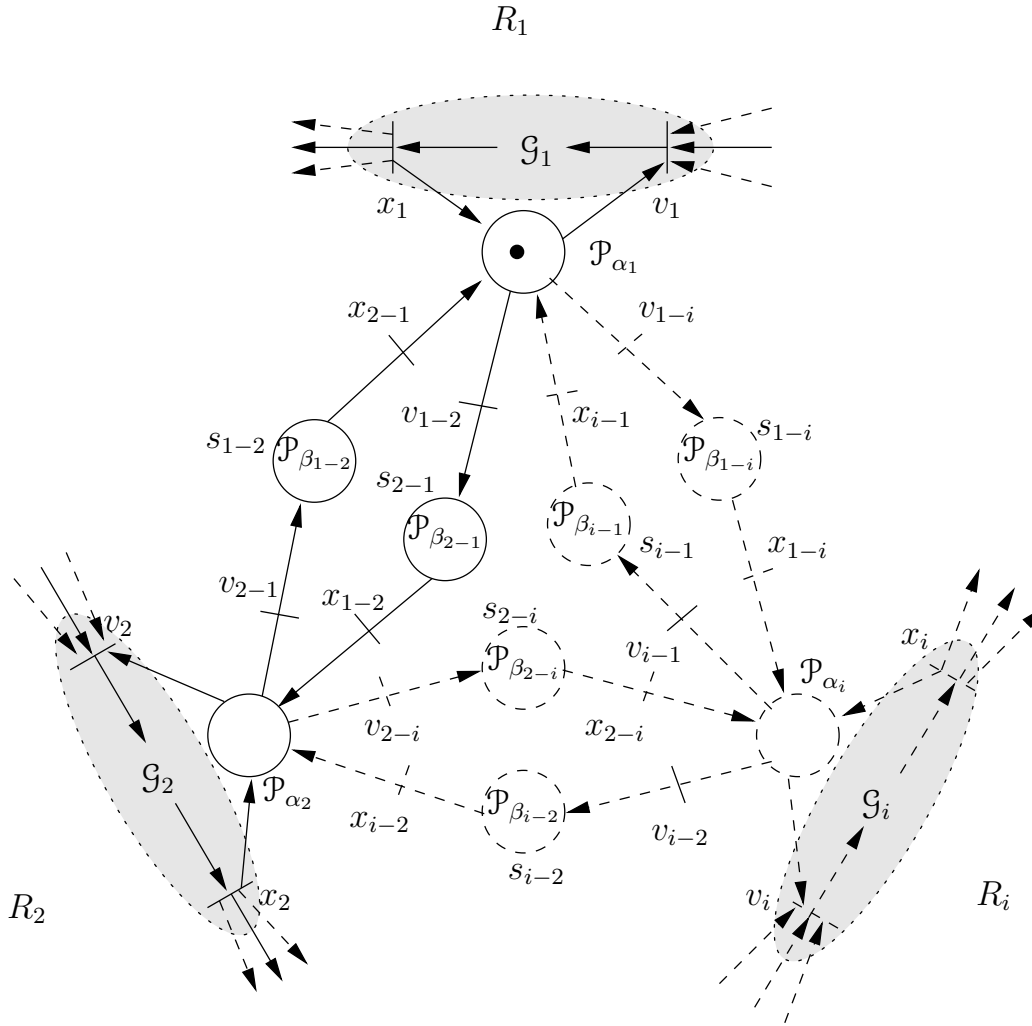


FIGURE 4.1 – Modèle RdP d'un PRPITC.

Ce modèle graphique représente \mathcal{N} utilisateurs partageant r ressources. Nous assignons à chaque utilisateur $i \in [1, \mathcal{N}]$ un sous-RdP \mathcal{G}_i qui modélise son comportement pendant qu'il utilise une ou plusieurs ressources (section critique de i). Un tel sous-RdP ne doit avoir qu'une transition d'entrée et une transition de sortie en relation avec la place \mathcal{P}_{α_i} qui lui correspond ; ces transitions sont dénotées v_i et x_i respectivement. Dans cette représentation générale, nous pouvons observer l'évolution des compteurs de chaque transition. Le tableau 4.1 détaille la sémantique du modèle.

	Signification
s_{i-j}	temps (fixe) nécessaire à la reconfiguration pour un utilisateur j d'une ressource réglée pour un utilisateur i
v_i	$v_i(t)$ est le nombre de requêtes de l'utilisateur i honorées jusqu'à la date t
x_i	$x_i(t)$ est le nombre de ressources restituées par l'utilisateur i jusqu'à la date t
v_{i-j}	$v_{i-j}(k)$ est la date de début de la k^{e} opération de reconfiguration pour un utilisateur j d'une ressource réglée pour un utilisateur i
x_{i-j}	$x_{i-j}(k)$ est la date de fin de la k^{e} opération de reconfiguration pour un utilisateur j d'une ressource réglée pour un utilisateur i

TABLEAU 4.1 – Sémantique des figures 4.1 et 4.2.

Dans le but de modéliser un système complet dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$, nous proposons d'abord un modèle générique pour un PRPITC. Comme dans le PRPI classique, chaque utilisateur i garde la ressource pendant un temps variable $\tau_i(t)$ qui lui est propre. Cependant, une nouvelle difficulté émerge. En effet, lorsque la ressource change d'utilisateur, un temps de configuration (fixe) est nécessaire pour préparer la nouvelle attribution. Ce temps fixe dépend de l'utilisateur courant i et de l'utilisateur j pour lequel la ressource doit être configurée. Il est habituellement supérieur aux temps $\tau_i(\cdot)$ et $\tau_j(\cdot)$, ce qui complexifie le problème d'attribution de la ressource.

Hormis le travail présenté dans [van den Boom et De Schutter, 2006], qui conduit à un nombre de modes exponentiel, nous rappelons qu'il n'y a pas de technique systématique aussi générale disponible dans la littérature pour modéliser de tels problèmes dans l'algèbre des dioïdes. Le but de cette section est précisément de combler ce manque théorique [Corréia et al., 2007c].

4.1.2 Modélisation analytique

Le modèle RdP de la figure 4.1 est scindé en deux parties, telles que montrées dans la figure 4.2. Nous distinguons la partie concernant les utilisateurs, qui contient les sous-RdPs notés \mathcal{G}_i , de celle relevant des ressources partagées, dénotée \mathcal{G}_α . Nous rappelons que nous limitons notre contribution au cas discret dans lequel l'invariance de ressources est respectée, à savoir, il existe, à tout instant $t \geq 0$, un entier naturel τ tel que $x_i(t + \tau) = f(v_i(t))$.

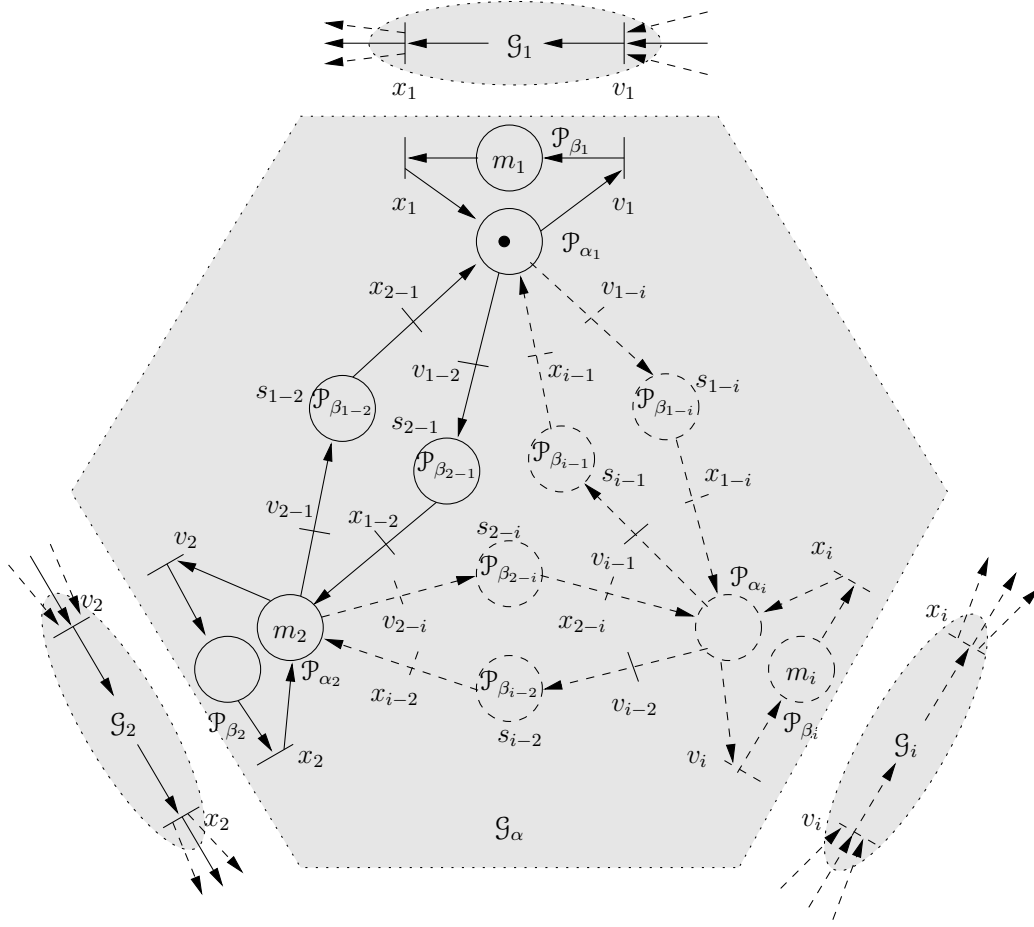


FIGURE 4.2 – Modèle scindé d'un PRPITC.

Le sous-RdP \mathcal{G}_α possède un invariant de place. À tout instant, la somme des marquages de ses places est égale au nombre de ressources dont dispose le système. Dans le but d'étudier le comportement des transitions d'entrée et de sortie de chaque sous-RdP \mathcal{G}_i dans les conditions de partage de ressources, celles-ci sont dupliquées. Sans perte de généralité, chaque sous-RdP \mathcal{G}_i est remplacé par une place fictive \mathcal{P}_{β_i} dans \mathcal{G}_α , de telle manière que son marquage initial $\mathcal{M}_0(\mathcal{P}_{\beta_i})$ corresponde au nombre de ressources détenues initialement par l'utilisateur i . Ceci contribue à concentrer l'analyse sur les ressources partagées, quelque soit le comportement des utilisateurs. Ainsi, nous considérons \mathcal{G}_α comme le modèle élémentaire d'un SRPITC. Partant du principe qu'aucune ressource n'est en cours de configuration à l'instant $t = 0$ (i.e. $\mathcal{M}_0(\mathcal{P}_{\beta_{i-j}}) = 0, \forall i, j \in [0, N]$), nous pouvons écrire :

$$\sum_{i=1}^N (\mathcal{M}_0(\mathcal{P}_{\alpha_i}) + \mathcal{M}_0(\mathcal{P}_{\beta_i})) = r \quad .$$

À tout instant t , si un jeton est présent dans la place \mathcal{P}_{α_i} , cela signifie qu'une ressource est attribuée à l'utilisateur i . Un jeton dans la place \mathcal{P}_{β_i} traduit le fait que l'utilisateur i est en train

de se servir d'une ressource. Pour chaque jeton dans une place $\mathcal{P}_{\beta_{ij}}$, une ressource attribuée précédemment à l'utilisateur i est reconfigurée pour l'utilisateur j . Chacune de ces opérations a un temps d'exécution qui lui est propre : il est noté au-dessus de la place correspondante (voir tableau 4.1). Chaque tir de transition u_i correspond à une requête de l'utilisateur i . Chaque fois qu'une ressource est attribuée à i , un tir de v_i a lieu. Une fois que l'utilisateur i restitue la ressource, x_i est tirée. Enfin, les tirs de y_i énumèrent les tâches que l'utilisateur i a terminées. La sémantique du modèle de la figure 4.2 est résumée dans le tableau 4.1. L'analyse structurale de \mathcal{G}_α nous fournit le lemme suivant.

Lemme 4.1 (Contraintes) *Soient \mathcal{N} utilisateurs partageant r ressources de même nature avec temps de configuration. Si, à tout instant t , chaque ressource est utilisée par au plus un utilisateur, alors les compteurs $(\min, +)$ associés à ses transitions d'entrée et de sortie, notés respectivement $v_i(\cdot)$ et $x_i(\cdot)$, ainsi que les compteurs des transitions reflétant les débuts et fin de configuration des ressources depuis ou vers un utilisateur j , notés respectivement $x_{j-i}(\cdot)$ et $v_{i-j}(\cdot)$, satisfont les \mathcal{N} inégalités suivantes :*

$$v_i(t) - x_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} (v_{i-j}(t) - x_{j-i}(t)) \leq \mathcal{M}_0(\mathcal{P}_{\alpha_i}) \quad \forall t. \quad (4.1)$$

Preuve: Prenons un utilisateur i . Par définition, les transitions v_i et v_{i-j} , $\forall j$, sont validées ssi il y a au moins un jeton dans la place \mathcal{P}_{α_i} correspondante. Soit $\mathcal{M}_t(\mathcal{P}_{\alpha_i})$ le marquage instantané de cette place. Nous savons que

$$\mathcal{M}_t(\mathcal{P}_{\alpha_i}) = \mathcal{M}_0(\mathcal{P}_{\alpha_i}) + x_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} x_{j-i}(t) - v_i(t) - \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} v_{i-j}(t) \quad \forall t.$$

Or par définition, $\mathcal{M}_t(\mathcal{P}_{\alpha_i}) \geq 0 \forall t$. D'où :

$$v_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} v_{i-j}(t) \leq \mathcal{M}_0(\mathcal{P}_{\alpha_i}) + x_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} x_{j-i}(t) \quad ,$$

ce qui est équivalent aux contraintes (4.1) définie pour l'utilisateur i . ■

Il est important de noter que les contraintes (4.1) ont un fort pouvoir de contrôle du bon fonctionnement du système étudié. En effet, il fournit une relation qui implique seulement des compteurs de transitions de \mathcal{G}_α et le nombre de ressources initialement disponibles. Cependant, des points de vue théorique et pratique, il est intéressant de vérifier que ces contraintes sont en accord avec le comportement typique d'un SRPITC. Pour ce faire, il suffit de montrer que, pour chaque utilisateur i :

- (a) les contraintes rejettent toute réquisition ou reconfiguration de ressource s'il n'y en a plus de disponible parmi celles réglées pour l'utilisateur i ,

- (b) s'il y a au moins une ressource disponible et correctement configurée, les contraintes permettent l'attribution ou la reconfiguration d'une ressource,
- (c) les contraintes ne doivent pas empêcher l'utilisateur i de restituer une ressource.

Remarquons que la condition (c) est évidemment satisfaite puisque les compteurs des transitions $x_i(\cdot)$ et $x_{j-i}(\cdot)$ sont multipliés par -1 dans la contrainte (4.1) correspondant à l'utilisateur i . Ce qui signifie que chaque tir d'une telle transition relaxe cette contrainte. Ainsi, il reste à prouver (a) et (b), ce qui est le but du lemme suivant.

Lemme 4.2 *Soit un SRPITC. Alors, sous les contraintes (4.1), une transition v_i peut être tirée si au moins une ressource est disponible.*

Preuve: Lorsque aucune ressource n'est à la fois disponible et configurée pour l'utilisateur i , alors la place \mathcal{P}_{α_i} ne contient aucun jeton. Dans ce cas,

$$\begin{aligned} 0 &= \mathcal{M}_t(\mathcal{P}_{\alpha_i}) \quad , \\ \iff 0 &= \mathcal{M}_0(\mathcal{P}_{\alpha_i}) - v_i(t) + x_i(t) - \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} (v_{i-j}(t) - x_{j-i}(t)) \quad , \\ \iff v_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} v_{i-j}(t) &= \mathcal{M}_0(\mathcal{P}_{\alpha_i}) + x_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} x_{j-i}(t) \quad . \end{aligned}$$

Autrement dit, tirer une transition v_i ou v_{i-j} , $\forall j \in [1, \mathcal{N}] - \{i\}$ signifie violer la contrainte correspondant à l'utilisateur i . ■

En combinant les résultats précédents, nous obtenons le théorème suivant.

Théorème 4.1 *Le comportement d'un SRPITC avec \mathcal{N} utilisateurs est décrit par les inégalités d'état suivantes, pour tout utilisateur i :*

$$x_i(t) \leq v_i(t - \tau_i(t)) + \mathcal{M}_0(\mathcal{P}_i) \quad , \quad (4.2)$$

où $\tau_i(t)$ est la durée variable pendant laquelle l'utilisateur i détient une ressource (cette durée dépend du comportement interne de l'utilisateur, décrit par \mathcal{G}_i). Pour chaque utilisateur i , les séquences de tirs $v_i(\cdot)$ et $v_{i-j}(\cdot)$, $\forall j \in [1, \mathcal{N}] - \{i\}$, doivent être choisies de telle manière que, pour tout t :

$$v_i(t) - x_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} (v_{i-j}(t) - x_{j-i}(t)) \leq \mathcal{M}_0(\mathcal{P}_{\alpha_i}) \quad . \quad (4.3)$$

Preuve: Il est prouvé, notamment dans [Baccelli *et al.*, 1992], que les inégalités (4.2) décrivent l'ensemble des tirs possibles des transitions. Les contraintes (4.3) sont directement dérivées du lemme 4.1. Enfin le lemme 4.2 démontre que l'utilisation de (4.2) et (4.3) est compatible avec le comportement d'un SRPITC. ■

4.1.3 Écriture du modèle dans l'algèbre des dioïdes

Puisque la description du comportement d'un SRPITC donnée par le théorème 4.1 est très simple et donne un large aperçu de la modélisation du PRPI, il est naturel de se demander si la même description peut être écrite dans l'algèbre $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$. Puisqu'il est désormais possible, d'après la proposition 2.1 page 62, de calculer le marquage instantané d'une place à partir des séries de $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$, nous sommes en position de récrire la contrainte (4.1) dans $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$, comme le montre le lemme suivant.

Lemme 4.3 (Contrainte dans $\overline{\mathbb{Z}}_{\min}$) Soient \mathcal{N} utilisateurs partageant r ressources avec temps de configuration. Si, à tout instant t , chaque ressource est utilisée par au plus un utilisateur, alors les séries $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ associés à ses transitions d'entrée et de sortie, notés respectivement v_i et x_i , ainsi que les séries des transitions reflétant les débuts et fin de configuration des ressources depuis ou vers un utilisateur j , notés respectivement x_{j-i} et v_{i-j} , satisfont les \mathcal{N} inégalités suivantes :

$$\frac{v_i}{x_i} \bullet \odot \left(\bigoplus_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} \frac{v_{i-j}}{x_{j-i}} \right) \bullet \preceq \gamma^{\mathcal{M}_0(\mathcal{P}_{\alpha_i})} \delta^*.$$

Preuve: La preuve découle immédiatement de (2.1) et (2.5) page 48 en appliquant l'opérateur \bullet à l'équation (4.1), ce qui permet d'écrire la contrainte pour tout t sous forme de séries de $\gamma^* \mathbb{B}[[\gamma, \delta]]$. ■

En résumé, puisque les équations d'état $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ qui décrivent le comportement d'un GET vivant sont bien connues (voir la sous-section 2.1.2 page 49), alors le lemme 4.3 conduit au théorème suivant.

Théorème 4.2 Dans le cadre d'un fonctionnement au plus tôt, le comportement d'un GET vivant incluant un SRPITC avec \mathcal{N} utilisateurs est donné par les équations d'état suivantes dans $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$:

$$\left\{ \begin{array}{l} X = AX \oplus BU \oplus SV \quad , \\ Y = CX \quad , \\ \text{avec, pour tout } i \in [1, \mathcal{N}] : \frac{\Gamma_i V}{\Delta_i X} \bullet \odot \left(\bigoplus_{\substack{j=1 \\ j \neq i}}^{\mathcal{N}} \frac{\Gamma_{i-j} V}{\Delta_{j-i} X} \right) \bullet \preceq \gamma^{\mathcal{M}_0(\mathcal{P}_{\alpha_i})} \delta^* , \end{array} \right.$$

où V est le vecteur des transitions d'entrée dans les sections critiques des utilisateurs ou dans les sections de configuration, X est le vecteur des transitions de sortie correspondant et les vecteurs $\Gamma^T \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]^q$ et $\Delta^T \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]^n$ sont définis pour tout $l \in [1, \mathcal{N}(1 + \mathcal{N})]$ par

$$[\Gamma]_m, [\Delta]_m = \begin{cases} \mathbf{e} & \text{si } m = l, \\ \varepsilon & \text{sinon.} \end{cases}$$

Nous avons proposé une modélisation du PRPITC dans l'algèbre des dioïdes. Nous allons maintenant voir comment utiliser cette modélisation pour représenter le comportement du trafic au sein d'une intersection avec rouge intégral.

4.1.4 Application au trafic urbain

Dans cette partie, nous considérons l'intersection présentée par la figure 1.2 page 26, c'est-à-dire la rencontre de deux rues à sens unique et constituées d'une seule voie. Cependant, nous imposons cette fois-ci que les deux phases composant la commande des feux, à savoir vert-rouge et rouge-vert, soient séparées par une période rouge-rouge, voir figure 4.3. Les temps de feu orange sont répartis entre la phase considéré et la période de rouge intégral qui la suit. La proportion utilisée pour cette répartition dépend de la capacité (ou de la propension) des conducteurs à s'arrêter ou non. Pour chaque courant de mobiles, les signaux lumineux suivent la séquence suivante : rouge, vert et orange. Une période de vert est suivie d'une période d'orange, pendant laquelle un véhicule doit s'arrêter s'il peut le faire en toute sécurité.

feu de la rue R_1	vert	rouge	rouge	rouge	vert	rouge	...
feu de la rue R_2	rouge		vert		rouge		rouge
→ temps →							

FIGURE 4.3 – Les deux phases de feux sont séparées par une période de rouge intégral.

Si deux véhicules traversent successivement l'intersection, nous distinguons deux scénarios :

- les deux véhicules proviennent de la même rue, ils sont alors espacés d'au moins 2 secondes (débit maximum de $0.5veh.s^{-1}$);
- les deux véhicules proviennent chacun d'une rue différente, ils sont alors espacés d'un temps s supérieur ou égal à 2 secondes. En effet, cette situation implique un changement de phase.

La durée s inclut le temps d'orange, ainsi que le temps d'accélération dû au départ du véhicule en tête de la file, après un changement de phase.

4.1.4.A Modélisation par RdP

Nous présentons un nouveau modèle RdP pour une zone de conflit partagée par deux files d'attente de véhicules dans la figure 4.4. Ce modèle est composé de trois parties. R_1 et R_2 décrivent respectivement le comportement des véhicules lorsqu'ils traversent l'intersection. La partie centrale cerne les comportements admissibles par le système en termes de commande de feux.

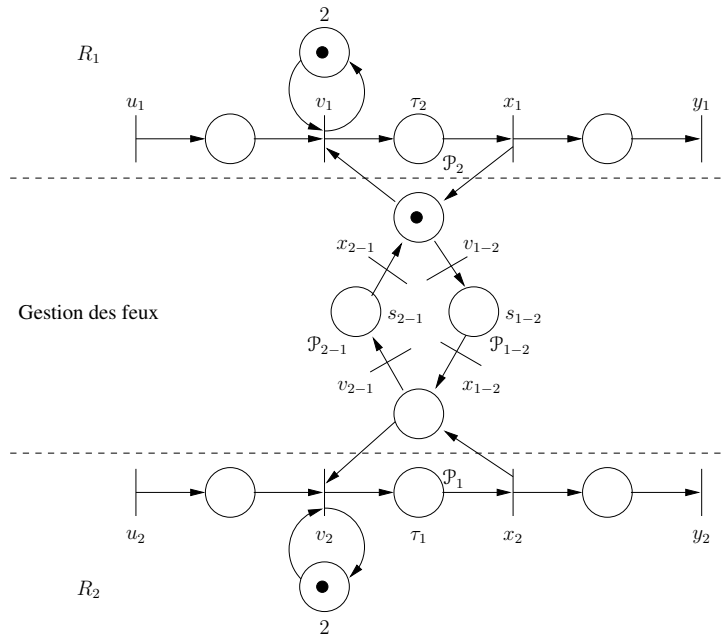


FIGURE 4.4 – Modèle RdP d’une intersection avec rouge intégral.

τ_1 et τ_2 représentent respectivement le temps nécessaire aux véhicules venant des rues R_1 et R_2 pour traverser la zone de conflit. Ces temps dépendent des longueurs partagées par ces rues et de la vitesse moyenne. Si un jeton est présent dans la place \mathcal{P}_1 , cela signifie qu’un véhicule provenant de R_1 traverse l’intersection. Il en est de même pour la place \mathcal{P}_2 concernant les véhicules provenant de R_2 . Un jeton dans la place \mathcal{P}_{1-2} indique qu’une phase de rouge intégral succède à une période de feu vert pour la rue R_1 et inversement pour la place \mathcal{P}_{2-1} . Les temporisations s_{1-2} et s_{2-1} de ces places tiennent compte de la durée du rouge intégral. Elles incluent également la durée du feu orange et le temps d’accélération du premier véhicule de la file qui s’apprête à recevoir le feu vert. La sémantique de la figure 4.4 est détaillée dans le tableau 4.2.

Soulignons enfin qu’en agissant sur les transitions v , nous pouvons choisir de prolonger une phase ou, au contraire, d’en changer. Ce modèle autorise toutes sortes de commande des feux. En particulier il n’impose pas de considérer la notion de cycle de feux.

4.1.4.B Modélisation dans $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

Puisque le modèle RdP du système sans intersection est un GET, le modèle RdP de la figure 4.4 est scindé en deux sous-RdPs comme explicité par la figure 4.5. Le premier sous-RdP regroupe les GETs représentant le comportement des véhicules dans les rues hors intersection, ainsi que la partie passive du modèle de gestion des feux. Le second représente la partie active de la gestion des feux, sur laquelle nous devons appliquer une loi de commande.

Signification	
τ_i	temps de parcours de la surface partagée. Il dépend de la longueur que la rue i partage avec les autres rues et de la vitesse moyenne dans cette rue
s_{i-j}	temps de rouge intégral succédant à une phase de vert pour la rue R_i et précédant une phase de vert pour la rue R_j
u_i	transition d'entrée ; $u_i(t)$ est le nombre de voitures entrées dans la rue i jusqu'à la date t
y_i	transition de sortie ; $y_i(t)$ est le nombre de voitures sorties de la rue i jusqu'à la date t
v_i	$v_i(t)$ est le nombre de fois qu'une voiture venant de la rue i s'est engagée dans l'intersection jusqu'à la date t
x_i	$x_i(t)$ est le nombre de fois qu'une voiture venant de la rue i a traversé l'intersection jusqu'à la date t
v_{i-j}	$v_{i-j}(k)$ est la date de début de la k^{e} phase de rouge intégral succédant à une phase de vert pour la rue R_i et précédant une phase de vert pour la rue R_j
x_{i-j}	$x_{i-j}(k)$ est la date de fin de la k^{e} phase de rouge intégral succédant à une phase de vert pour la rue R_i et précédant une phase de vert pour la rue R_j

TABLEAU 4.2 – Sémantique des figures 4.4 et 4.5

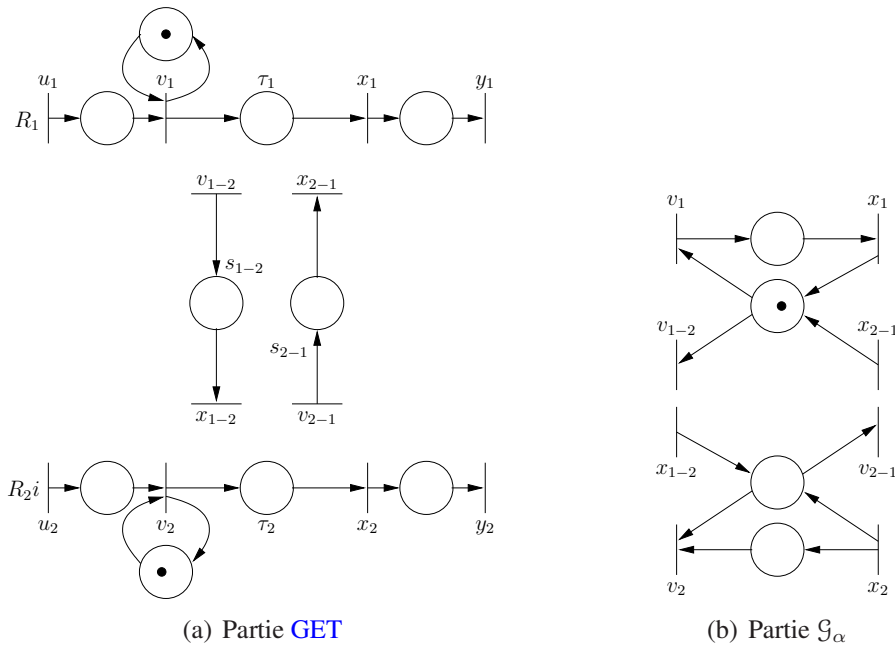


FIGURE 4.5 – Modèle scindé d'une intersection avec rouge intégral.

Nous avons désormais tous les éléments pour écrire un modèle $\mathcal{M}_{in}^{\alpha x}[\gamma, \delta]$ d'une intersection élémentaire concernant deux rues avec rouge intégral. Soient les vecteurs X et V définis comme

suit :

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_{1-2} \\ x_{2-1} \end{pmatrix},$$

$$V = \begin{pmatrix} v_1 \\ v_2 \\ v_{1-2} \\ v_{2-1} \end{pmatrix}.$$

Le comportement des véhicules au sein de l'intersection élémentaire de deux rues avec rouge intégral est décrit par les équations d'état suivantes :

$$\left\{ \begin{array}{l} X = \begin{pmatrix} \delta^{\tau_1} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \delta^{\tau_2} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \delta^{s_{1-2}} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \delta^{s_{2-1}} \end{pmatrix} V, \\ V \stackrel{!}{=} \begin{pmatrix} \gamma \delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \gamma \delta^2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} V \oplus \begin{pmatrix} \mathbf{e} & \varepsilon \\ \varepsilon & \mathbf{e} \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{pmatrix} U, \\ Y = \begin{pmatrix} \mathbf{e} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{e} & \varepsilon & \varepsilon \end{pmatrix} X, \end{array} \right. \quad (4.4)$$

avec, pour tous $i, j \in [1, 2]$ tels que $j \neq i$, $\frac{\Gamma_i V}{\Delta_i X} \odot \frac{\Gamma_{i-j} V}{\Delta_{j-i} X} \preceq \gamma^{\mathcal{M}_0(\mathcal{P}_{\alpha_i})} \delta^*$,

où V est le vecteur des transitions d'entrée des véhicules dans la zone de conflit ou de début de phase de rouge intégral, X est le vecteur des transitions de sortie correspondant et les vecteurs $\Gamma^T \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]^q$ et $\Delta^T \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]^n$ sont définis pour tout $l \in [1, 6]$ par

$$[\Gamma_l]_m, [\Delta_l]_m = \begin{cases} \mathbf{e} & \text{si } m = l, \\ \varepsilon & \text{sinon.} \end{cases}$$

4.2 Commande

Comme nous l'avons mentionné, nous considérons l'intersection comme une ressource partagée par les véhicules de deux rues. Ainsi, la stratégie de commande doit allouer l'utilisation

de la ressource aux différents véhicules en fonction de leurs dates d'arrivée dans le système. L'objectif est de libérer la ressource le plus tôt possible. Autrement dit, il s'agit de vider les rues et l'intersection au plus vite. Le modèle algébrique (4.4) nous informe sur les arrivées de véhicule et sur les situations de conflit. Il convient désormais de choisir une stratégie de résolution de ces conflits.

La difficulté du problème vient des temps de configuration qui interviennent à chaque changement d'utilisateur. En effet, dans la plupart des situations où il faut faire un choix, la question est de savoir s'il faut prolonger le bail de l'utilisateur courant, même si celui-ci ne requiert pas la ressource immédiatement, ou bien changer d'utilisateur, en sachant que cela implique une forte pénalité en terme de temps. Il a été démontré que le problème général est *NP*-complet [Potts et Kovalyov, 2000].

Pour contourner ce problème de complexité algorithmique, nous recadrons le problème de régulation d'une intersection par rapport au problème général, afin de dégager des sous-problèmes qui soient solubles en un temps polynomial. Ainsi, nous proposons deux commandes optimales vis-à-vis de deux modes de fonctionnement distincts.

4.2.1 Commande en juste-à-temps

Dans un premier temps, nous proposons de simplifier le problème en supposant que le système doit être occupé en permanence, soit parce qu'un utilisateur détient une ressource, soit parce que celle-ci est en cours de configuration. Ceci correspond à un fonctionnement au plus tôt. Dans la littérature de la modélisation des GETs à l'aide de l'algèbre des dioïdes, ce comportement est régulièrement adopté. Il reste à arbitrer les conflits lorsque deux transitions sont validées, lors d'une synchronisation, par le même unique jeton dans une place amont commune. Ce cas n'apparaît que lorsque l'utilisateur pour lequel la ressource est configurée restitue celle-ci et la requiert à nouveau à la même date t . Le choix se résume à décider si le bail de l'utilisateur doit être prolongé, ou bien si la ressource doit être configurée pour un autre utilisateur.

Dans ce cas, nous pouvons montrer que la meilleure solution consiste à attribuer la ressource à l'utilisateur pour laquelle elle est configurée, tant que celui-ci la réclame. Dès que l'utilisateur marque une pause, la ressource doit être immédiatement configurée pour le prochain utilisateur, c'est-à-dire, celui qui l'a demandée depuis la date la plus ancienne (voir [Jackson, 1955]).

Appliquée au trafic, cette règle d'attribution revient à maintenir la phase courante tant que des véhicules de la rue ayant le feu vert se présentent à l'entrée de l'intersection. Plus formellement, la commande optimale au plus tard est donnée par la proposition suivante.

Proposition 4.1 *Si une intersection avec rouge intégral fonctionne au plus tôt, la solution optimale consiste à privilégier, lorsqu'un choix s'impose, le passage véhicules par rapport au changement de phase.*

Preuve: Nous utilisons les notations du tableau 4.2 page 122 et le modèle de la figure 4.4 page 121. Pour améliorer la lisibilité de la démonstration, nous supposons que les véhicules traversent l'intersection en 1 seconde : $\tau_1 = \tau_2 = 1\text{u.t.}$

Soit $q_1(t)$ et $q_2(t)$ le nombre de véhicules en attente respectivement dans la rue R_1 et R_2 . Rappelons que le critère à minimiser est $J = q_1(t) + q_2(t)$ pour tout t . D'après la contrainte donnée par le lemme 4.1 page 117 et puisque le RdP suit un fonctionnement au plus tôt, nous pouvons établir l'équation suivante :

$$v_1(t) + v_{1-2}(t) = 1 + v_{2-1}(t - s_{2-1})v_1(t - 1) \quad .$$

Or, par définition, $q_1(t) = u_1(t) - v_1(t)$, autrement dit $v_1(t) = u_1(t) - q_1(t)$. Ainsi, si nous remplaçons v_1 dans l'équation précédente, nous obtenons :

$$u_1(t) - q_1(t) + v_{1-2}(t) - v_{2-1}(t - s_{2-1}) - u_1(t - 1) + q_1(t - 1) = 1 \quad .$$

Notons $\Delta q_1(t) = q_1(t) - q_1(t - 1)$ et $\Delta u_1(t) = u_1(t) - u_1(t - 1)$. Il est alors possible d'écrire :

$$\begin{aligned} 1 &= -\Delta q_1(t) + \Delta u_1(t) + v_{1-2}(t) - v_{2-1}(t - s_{2-1}) \quad , \\ \iff \Delta q_1(t) &= \Delta u_1(t) + v_{1-2}(t) - v_{2-1}(t - s_{2-1}) - 1 \quad . \end{aligned}$$

Observons que $q_1(t) = \sum_{i=0}^t \Delta q_1(t)$ et $u_1(t) = \sum_{i=0}^t \Delta u_1(t)$. De plus, la règle de causalité impose que $v_i(t) = 0$ pour tout $t < 0$. Nous avons alors, en sommant l'équation précédente par rapport à t :

$$q_1(t) = \sum_{i=0}^t \Delta q_1(t) = u_1(t) + \sum_{i=0}^t v_{1-2}(i) - \sum_{i=0}^{t-s_{2-1}} v_{2-1}(i) - t - 1 \quad .$$

En procédant de la même manière pour la rue 2, nous retrouvons facilement :

$$q_2(t) = u_2(t) + \sum_{i=0}^t v_{2-1}(i) - \sum_{i=0}^{t-s_{1-2}} v_{1-2}(i) \quad .$$

En conséquence de quoi, notre critère J s'écrit :

$$\begin{aligned} J &= q_1(t) + q_2(t) = u_1(t) + \sum_{i=0}^t v_{1-2}(i) - \sum_{i=0}^{t-s_{2-1}} v_{2-1}(i) - t - 1 \\ &+ u_2(t) + \sum_{i=0}^t v_{2-1}(i) - \sum_{i=0}^{t-s_{1-2}} v_{1-2}(i) \quad , \\ &= u_1(t) + u_2(t) + \sum_{i=t-s_{1-2}+1}^t v_{1-2}(i) + \sum_{i=t-s_{2-1}+1}^t v_{2-1}(i) - t - 1 \quad . \end{aligned}$$

Ainsi, nous constatons que chaque tir de v_{1-2} et v_{2-1} est pénalisant vis-à-vis du critère J , alors que les tirs de v_1 et v_2 n'agissent pas sur J , ce qu'il fallait démontrer. ■

Notons que nous sommes dans le cas où le changement d'attribution est imposé puisque, à cause du fonctionnement au plus tôt, aucun temps mort n'est toléré. Notons également que malgré ce fait, les changements doivent être évités au maximum. Ainsi, cette proposition met en lumière une propriété intéressante de la régulation optimale dans une intersection en général. D'un point de vue pratique, elle nous enseigne qu'il vaut mieux prolonger la durée du feu vert pour les véhicules suiveurs vis-à-vis du critère à optimiser.

4.2.2 Commande optimale par programmation dynamique

Comme pour le PRPI abordé dans le chapitre précédent, la littérature de la recherche opérationnelle fournit, pour le PRPITC, un algorithme de commande optimale en temps polynomial lorsqu'il n'y a qu'une ressource [Ghosh et Gupta, 1997, Potts et Kovalyov, 2000, Brucker, 2006]. Ceci convient parfaitement pour résoudre le problème de la planification des feux dans une intersection avec rouge intégral.

4.2.2.A Spécification du problème

Nous considérons l'intersection de la figure 1.2 page 26 avec rouge intégral. Avant de présenter la stratégie de commande, nous introduisons les notations présentées dans le tableau 4.3.

Variable	Définition
g	indice de la rue
n_g	nombre instantané de véhicules dans la rue g
(q_g, g)	q_g^{e} véhicule de la rue g , $1 \leq q_g \leq n_g$
$r_{(q_g, g)}$	date d'arrivée du véhicule (q_g, g)
$p_{(q_g, g)}$	date d'arrivée au plus tôt du véhicule (q_g, g) à hauteur du feu
$s_{g', g}$	somme des temps de rouge intégral et d'accélération du premier véhicule lors d'un changement de phase de la rue g' à la rue g
$G(q_1, q_2, g)$	temps total minimum de traversée en fonction de la stratégie (mise à jour à chaque itération de l'algorithme) pour tout véhicule $(1, 1), \dots, (q_1, 1), (1, 2), \dots, (q_2, 2)$. Selon cette stratégie, le dernier véhicule quittant le système provient de la rue g

TABLEAU 4.3 – Notations utilisées pour décrire la commande optimale d'un PRPITC.

Nous supposons que les véhicules provenant de chaque rue quittent l'intersection dans l'ordre croissant de leurs dates d'arrivées. Autrement dit, la règle « premier entré, premier sorti » est de vigueur. Ceci concorde bien avec la réalité de notre système de trafic, dont les rues ne possèdent qu'une voie, ce qui empêche les véhicules de se doubler. Ainsi, nous avons, pour chaque rue $g \in [1, 2]$:

$$(1, g) \longrightarrow (2, g) \longrightarrow \dots \longrightarrow (n_g, g) \quad ,$$

où $a \longrightarrow b$ signifie que le véhicule a précède le véhicule b .

Puisque l'objectif est d'évacuer les véhicules du système aussi vite que possible, nous devons réduire les temps de parcours de chaque véhicule dans chaque rue. Nous avons défini $G(q_1, q_2, g)$ comme étant le temps total minimum de présence des véhicules dans le système. Cette variable peut être interprétée comme la période de temps calculée par la différence entre la date de départ (*i.e.* sortie du système) du dernier véhicule et la date d'arrivée (*i.e.* entrée dans le système) du premier véhicule. Autrement dit, la stratégie optimale consiste en l'arrangement des phases de feux qui minimise la fonction objectif suivante :

$$\min(G(n_1, n_2, 1), G(n_1, n_2, 2)) \quad .$$

Dans la notation $\alpha|\beta|\gamma$ utilisée habituellement dans la littérature pour décrire les problèmes de planification (voir [Graham *et al.*, 1979, Brucker et Knust, 1999]), notre problème peut être écrit comme suit : $1|r_i, p_i, s_f| \sum C_i$. La surface partagée fait office de « machine » pour le problème de planification ainsi défini ; elle est évidemment unique, d'où le début de la notation, « 1 ». Dans la partie β , « r_i » implique que les véhicules mettent un certain temps (constant) à parcourir la rue, avant d'arriver à hauteur du feu ($r_i = r_{(q_g, g)}$). « p_i » signifie que les utilisateurs requièrent la ressource pendant un temps constant. Autrement dit les véhicules traversent l'intersection en un temps déterminé à l'avance. Ce temps est calculé à partir de la longueur de la zone de conflit et de la vitesse moyenne ($p_i = p_{(q_g, g)}$). « s_f » à son tour signifie que lorsque la ressource change d'utilisateur, elle doit d'abord être reconfigurée pendant un temps connu et constant. Ceci correspond au temps de rouge intégral, auquel nous ajoutons le temps d'accélération du premier véhicule lors d'un changement de phase, ainsi que la portion de temps de feu orange pendant laquelle les véhicules dépendant de la phase précédente ne peuvent s'arrêter en toute sécurité ($s_f = s_{g', g}$). Enfin, « $\sum C_i$ » traduit le fait que le critère à minimiser est la somme des temps de parcours de chaque véhicule ($\sum C_i = G(q_1, q_2, g)$).

4.2.2.B Algorithme de commande

Il est proposé, dans [Potts et Kovalyov, 2000], un algorithme de programmation dynamique pour résoudre ce problème de manière optimale. Puisque le temps d'exécution de cet algorithme est polynomial ($\mathcal{O}(4n^2)$, avec n véhicules), il peut être implémenté pour une application en temps réel. La formule récursive est la suivante :

$$G(q_1, q_2, g) = \min_{g'=1,2} (G(q'_1, q'_2, g') + s_{g', g} + p_{(q_g, g)} + \tau) \quad ,$$

où $q'_f = q_f$ si $f \neq g$, ou alors $q'_f = q_f - 1$ si $f = g$, avec $f = 1, 2$. τ est le temps d'attente du véhicule (q_g, g) .

Si le dernier véhicule de la stratégie précédente provient également de la rue g , alors $s_{g', g} = 0$ puisque $g' = g$. Cela sous-entend que les feux ne changent pas de couleur. Sinon, si le dernier véhicule de la stratégie précédente provient d'une autre rue $g' \neq g$, alors cela implique un changement de phase.

À ce niveau, nous pouvons donner la formule permettant le calcul du temps d'attente τ du véhicule (q_g, g) :

$$\tau = \begin{cases} \max(0, r_{(q_g, g)} - G(q'_1, q'_2, g')) & \text{si } g' = g, \\ \max(0, r_{(q_g, g)} - G(q'_1, q'_2, g') - s_{g', g}) & \text{si } g' \neq g. \end{cases}$$

La formule récursive sélectionne, lors de l'itération k , la meilleure stratégie calculée jusqu'à l'itération $k-1$, pour laquelle le véhicule (q_g, g) est le dernier véhicule à traverser l'intersection. La récursion opère jusqu'à atteindre les conditions initiales.

Théoriquement, pour notre problème, quatre cas peuvent être considérés comme conditions initiales : $G(1, 0, 1)$, $G(0, 1, 2)$, $G(1, 0, 2)$ ou $G(0, 1, 1)$. En pratique, les cas $G(0, 1, 1)$

et $G(1, 0, 2)$ ne sont pas réalistes. En effet, $G(0, 1, 1)$ suppose à la fois qu'il n'y a aucun véhicule dans la rue 1, et qu'un véhicule de la rue 1 traverse l'intersection, ce qui est contradictoire. $G(1, 0, 2)$ traduit la même situation ambiguë pour la rue 2. Ainsi, ces deux cas sont écartés d'emblée des choix disponibles pour l'initialisation.

Soit $s_{0,g}$ le temps de rouge intégral avant que le premier véhicule $(1, g)$ arrive dans la rue g . Ce temps peut être non nul dans un cas réel d'application, lorsque la stratégie d'attribution de la zone de conflit est remise à jour. Les deux conditions initiales admissibles sont définies par :

$$\begin{aligned} G(1, 0, 1) &= s_{0,1} + \max(r_{(1,1)} - s_{0,1}, 0) + p_{(1,1)} \quad , \\ G(0, 1, 2) &= s_{0,2} + \max(r_{(1,2)} - s_{0,2}, 0) + p_{(1,2)} \quad . \end{aligned}$$

Si $s_{0,g} = 0$, ces relations peuvent être simplifiées comme suit :

$$\begin{aligned} G(1, 0, 1) &= r_{(1,1)} + p_{(1,1)} \quad , \\ G(0, 1, 2) &= r_{(1,2)} + p_{(1,2)} \quad . \end{aligned}$$

4.2.2.C Extension de la stratégie de commande des feux

Nous pouvons considérer d'autres éléments d'une situation réelle de trafic. Par exemple, il est possible de prendre en compte les véhicules prioritaires tels que les ambulances, les camions de pompier ou bien les véhicules des forces de l'ordre. Le système de commande du trafic devrait donner une grande priorité à ces véhicules et minimiser leur temps d'attente.

Pour résoudre ce nouveau problème, nous apportons une légère modification à l'algorithme. Ainsi, nous associons à chaque véhicule un poids proportionnel à son degré de priorité. Pour réduire le temps d'attente des véhicules prioritaires, il suffit alors de minimiser leur temps total pondéré de trajet.

Soit $w_{(q_g, g)}$ le poids associé au véhicule (q_g, g) . Redéfinissons $G(q_1, q_2, g)$ comme étant la somme des temps de parcours pondérés des véhicules $(1, 1), \dots, (q_1, 1), (1, 2), \dots, (q_2, 2)$. Nous avons alors :

$$G(q_1, q_2, g) = \min_{g'=1,2} \left(G(q'_1, q'_2, g') + w_{(q_g, g)}(s_{g',g} + p_{(q_g, g)} + \tau) \right) \quad ,$$

où le dernier véhicule (q_g, g) provient, selon cette stratégie, de la rue g . Si $f \neq g$, alors $q'_f = q_f$, sinon $q'_f = q_f - 1$. De plus, si $g = g'$ alors $s_{g',g} = 0$, sinon un changement de phase est opéré. Les autres variables ont la même définition que pour le problème sans pondération. Les initialisations de la stratégie de commande du trafic étendue sont les suivantes :

$$\begin{aligned} G(1, 0, 1) &= w_{(1,1)}(s_{0,1} + \max(r_{(1,1)} - s_{0,1}, 0) + p_{(1,1)}) \quad , \\ G(0, 1, 2) &= w_{(1,2)}(s_{0,2} + \max(r_{(1,2)} - s_{0,2}, 0) + p_{(1,2)}) \quad . \end{aligned}$$

Si $s_{0,g} = 0$, ces relations peuvent être simplifiées comme suit :

$$\begin{aligned} G(1, 0, 1) &= w_{(1,1)}(r_{(1,1)} + p_{(1,1)}) \quad , \\ G(0, 1, 2) &= w_{(1,2)}(r_{(1,2)} + p_{(1,2)}) \quad . \end{aligned}$$

4.2.2.D Exemple simple

Soit une intersection avec rouge intégral telle que présentée en figure 1.2 page 26. Pour chaque direction, nous supposons les arrivées de véhicules suivantes (voir figure 4.6) :

- cinq véhicules dans la rue R_1 , aux dates $0s$, $4s$, $10s$, $15s$ et $21s$;
- quatre véhicules dans la rue R_2 , aux dates $2s$, $8s$, $15s$ et $22s$.

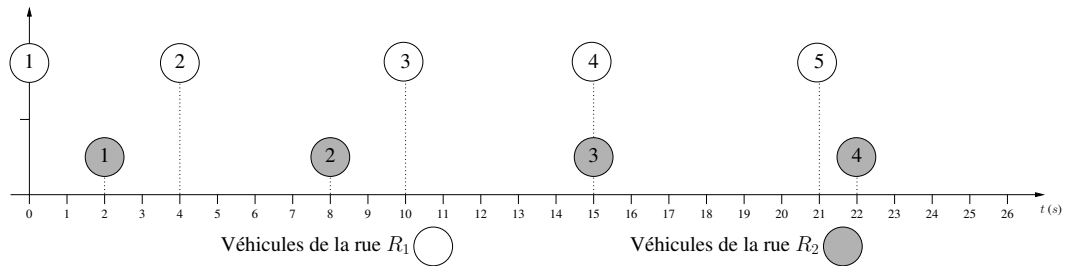


FIGURE 4.6 – Séquences d'arrivées de véhicules dans une intersection avec rouge intégral.

Nous supposons que les véhicules nécessitent deux secondes pour traverser l'intersection, quelle que soit leur provenance : $p_1 = p_2 = 2s$. Nous supposons aussi que la somme du temps de rouge intégral, de feu orange et d'accélération du premier véhicule de chaque file est égale à cinq secondes : $s_1 = s_2 = 5s$. L'algorithme donne alors la solution présentée dans la figure 4.7.

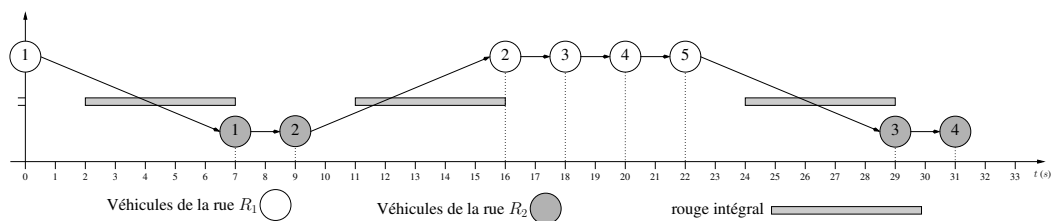


FIGURE 4.7 – Séquence optimale d'attribution de la zone de conflit aux véhicules dans une intersection avec rouge intégral.

Nous observons que la séquence proposée ne scinde pas les groupes de véhicules. Autrement dit, les véhicules qui arrivent à hauteur de l'intersection de manière suffisamment rapprochée profitent de la même phase de feu. Ainsi, la proposition 4.1 page 124 présente une propriété de la solution optimale.

4.3 Validation

Nous validons notre modèle d'intersection avec rouge intégral, ainsi que l'algorithme de commande optimale associé, à travers deux exemples d'application. Dans le premier, nous simulons un carrefour avec un cycle fixe de durée optimale. Nous comparons les résultats avec les performances obtenues lorsque nous réglons le même carrefour avec notre stratégie de commande. Dans le deuxième, nous simulons deux carrefours disposés en cascade que nous synchronisons de manière à générer une onde verte. Nous confrontons alors notre approche, appliquée indépendamment à chacun des carrefours, avec cette technique.

4.3.1 Simulation d'un carrefour

Dans [Wu *et al.*, 2007b], nous considérons une intersection avec rouge intégral (voir figure 1.2 page 26). Nous considérons que les véhicules mettent 2 secondes à traverser l'intersection. Le temps de rouge intégral, auquel nous additionnons les temps d'accélération des véhicules de tête et les temps de feu orange pendant lesquels les derniers véhicules ne peuvent s'arrêter, totalisent 3 secondes. Dans cet exemple, nous n'étudions pas le comportement des véhicules dans les zones de stockage des rues composant l'intersection, mais seulement dans la zone de conflit commune à ces rues.

4.3.1.A Modélisation du système

Le modèle RdP du système étudié est donné par la figure 4.8.

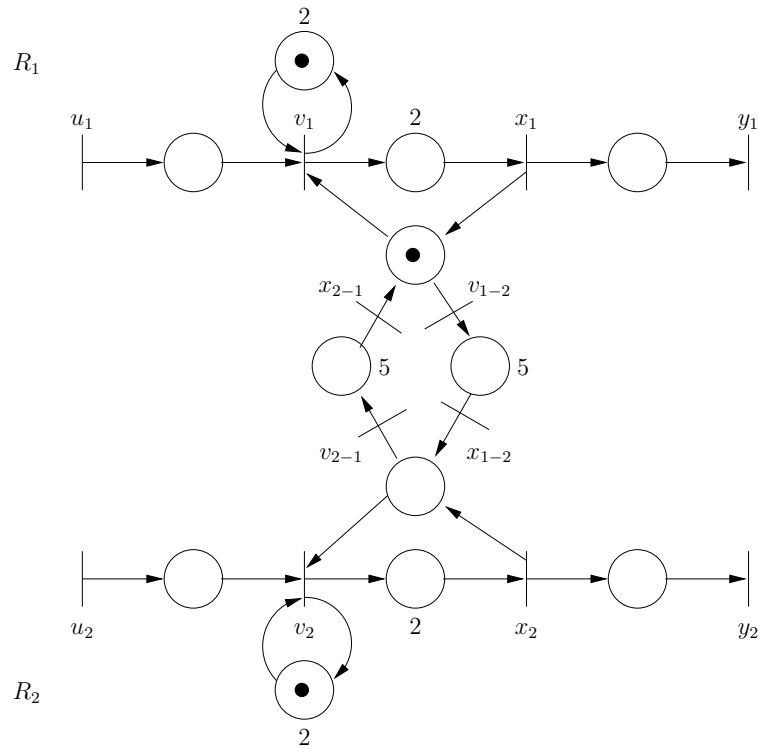


FIGURE 4.8 – Modèle Rdp d'une intersection avec rouge intégral.

Le modèle $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ de la figure 4.8 est donné par les équations suivantes :

$$\begin{aligned}
 X &= \begin{pmatrix} \delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \delta^2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \delta^5 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \delta^5 \end{pmatrix} V \quad , \\
 V &\preceq \begin{pmatrix} \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} V \oplus \begin{pmatrix} \mathbf{e} & \varepsilon \\ \varepsilon & \mathbf{e} \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{pmatrix} U \quad , \\
 Y &= \begin{pmatrix} \mathbf{e} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{e} & \varepsilon & \varepsilon \end{pmatrix} X \quad , \\
 \gamma\delta^* &\preceq \frac{v_1 \bullet}{x_1} \odot \frac{v_{1-2} \bullet}{x_{2-1}} \quad , \\
 \gamma\delta^* &\preceq \frac{v_2 \bullet}{x_2} \odot \frac{v_{2-1} \bullet}{x_{1-2}} \quad .
 \end{aligned}$$

4.3.1.B Stratégie de commande et résultats de simulation

Nous comparons d'abord les résultats obtenus avec notre algorithme de programmation dynamique avec ceux d'un cycle fixe. La durée totale de ce cycle, ainsi que celles de ses phases, est optimisée de la manière suivante [Cohen, 1993] :

- la durée de cycle optimale est donnée par :

$$c_1 = \frac{1.5T + 5}{1 - Y} ,$$

avec $T = 2s + 2P = 2 \times 3 + 2 \times 2$, soit 10 secondes et Y tel que

$$Y = 2 \times \frac{\lambda \times 3600}{2000} ;$$

- la durée de la phase de vert pour chaque rue est donnée par :

$$v = \frac{c_1 - T}{2} .$$

Nous avons réalisé deux simulations. La première considère des débits d'arrivées de véhicules de $\lambda = 0.1veh.s^{-1}$ pour chacune des rues, ce qui donne un cycle optimal de $31.25s$ avec des phases de $10.6s$. La figure 4.9 montre les temps d'attente des véhicules dans chacune des rues. La figure 4.10 montre le nombre de véhicules présents dans les rues.

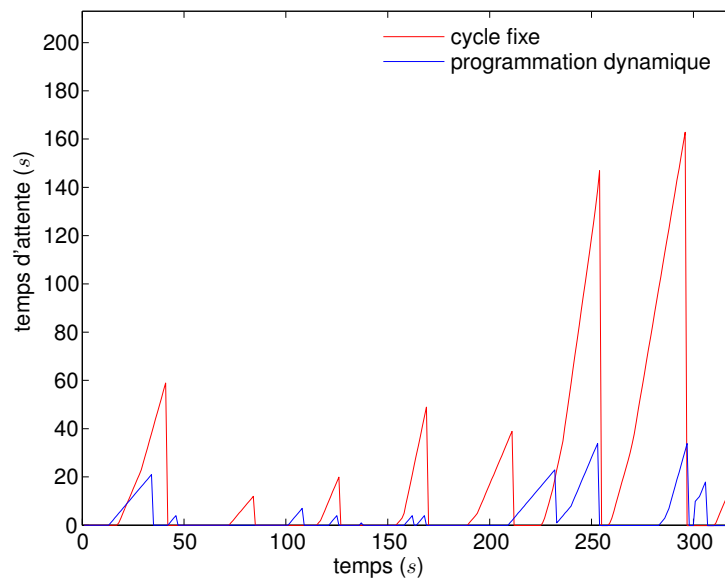


FIGURE 4.9 – Temps d'attente dans une intersection avec rouge intégral (débits d'entrée : $0.1veh.s^{-1}$).

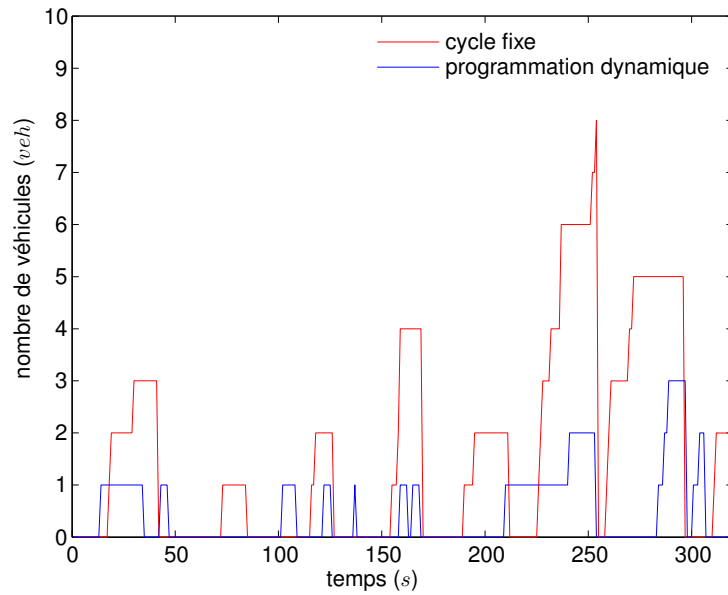


FIGURE 4.10 – Longueur des files d’attente dans une intersection avec rouge intégral (débits d’entrée : $0.1veh.s^{-1}$).

Nous avons également effectué les mêmes simulations, mais avec des débits d’arrivées de $0.2veh.s^{-1}$ pour chaque rue, ce qui donne un cycle optimal de $71.4s$ avec des phases de $30.7s$. La figure 4.11 montre les temps d’attente des véhicules dans chacune des rues. La figure 4.12 montre le nombre de véhicules présents dans les rues.

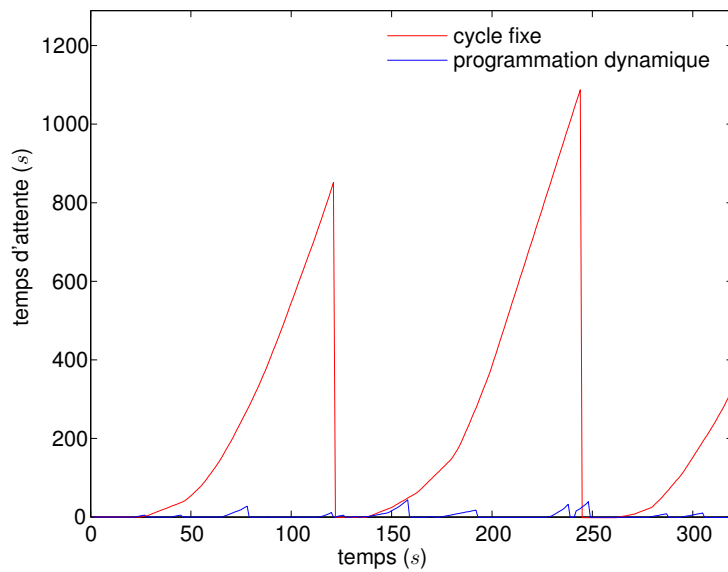


FIGURE 4.11 – Temps d’attente dans une intersection avec rouge intégral (débits d’entrée : $0.2veh.s^{-1}$).

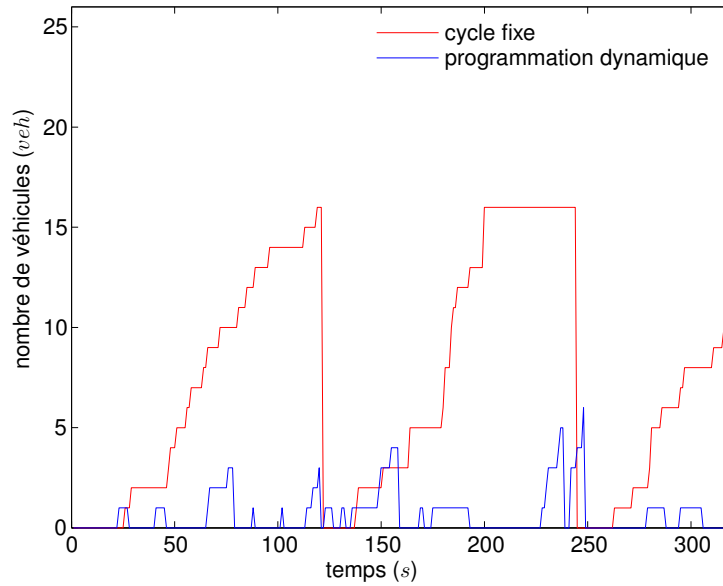


FIGURE 4.12 – Longueur des files d’attente dans une intersection avec rouge intégral (débits d’entrée : $0.2veh.s^{-1}$).

Les résultats de simulations montrent bien que notre algorithme de commande diminue fortement les temps d’attente des véhicules ainsi que les longueurs des files par rapport au cycle fixe.

4.3.2 Simulation de deux carrefours en cascade

Dans [Wu *et al.*, 2007a], nous considérons un système composé de deux intersections en cascade, agencées comme dans la figure 4.13.

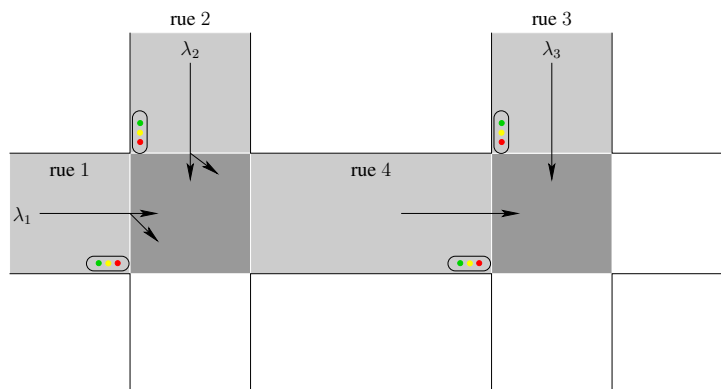


FIGURE 4.13 – Deux intersections en cascade.

Les débits d’arrivées de véhicules depuis les rues 1, 2 et 3 sont donnés respectivement par λ_1 , λ_2 et λ_3 . La moitié des véhicules en provenance des rues 1 et 2 tournent dans la rue 4. Parmi

les véhicules quittant la première intersection, seuls les véhicules se dirigeant vers la rue 4 sont pris en considération. Les véhicules en provenance des rues 3 et 4 vont tout droit.

Nous considérons que les véhicules mettent 2 secondes à traverser les intersections. Le temps de rouge intégral, auquel nous additionnons les temps d'accélération des véhicules de tête et les temps de feu orange pendant lesquels les derniers véhicules ne peuvent s'arrêter, totalisent 3 secondes. La rue 4 a une longueur de 500 mètres et la vitesse maximum est de $50km.h^{-1}$.

4.3.2.A Modélisation du système

L'objectif est ici de comparer les résultats d'une méthode classique de calcul d'onde verte avec notre algorithme de programmation dynamique. Afin de calculer les temps de cycle et des phases réalisant l'onde verte, nous représentons le système à l'aide d'un modèle véhicule-suiveur classique.

Notre approche utilise le modèle de rue présenté dans la sous-section 2.2.1 page 58, combiné avec le modèle d'intersection avec rouge intégral de la sous-section 4.1.4 page 120, comme montré dans la figure 4.14.

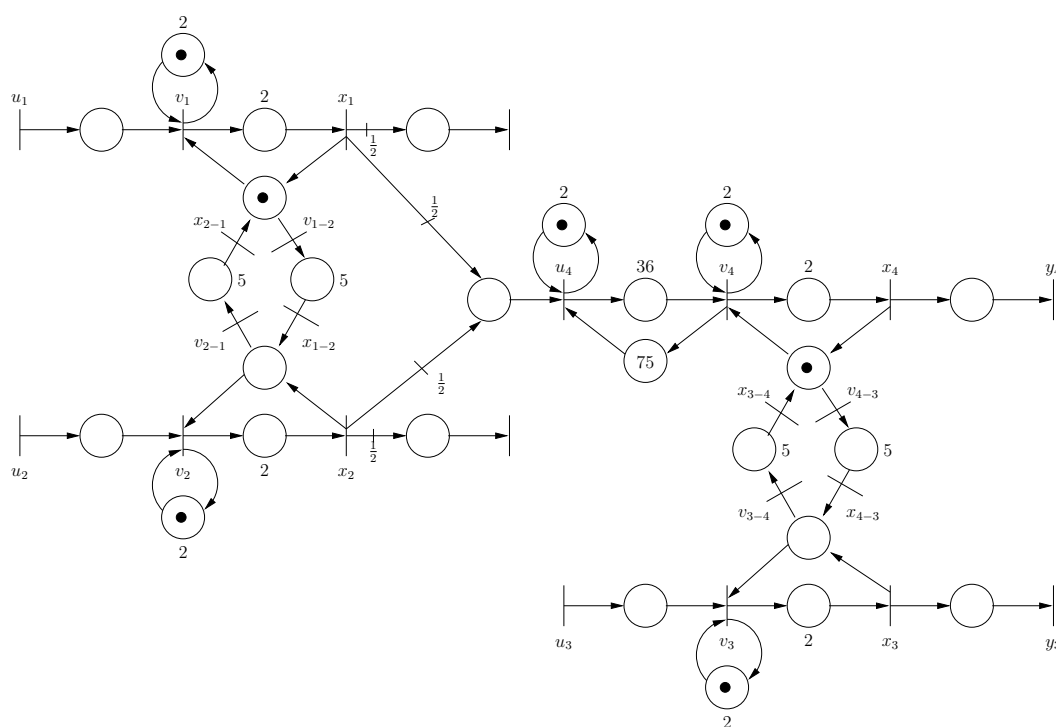


FIGURE 4.14 – Modèle RdP de deux intersections avec rouge intégral en cascade.

Le modèle $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ de la figure 4.14 est donné par les équations suivantes :

$$\begin{aligned}
 X &= \begin{pmatrix} \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \delta^5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \delta^5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^5 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^5 \end{pmatrix} V, \\
 V &\preceq \begin{pmatrix} \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} V \oplus \begin{pmatrix} \mathbf{e} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \mathbf{e} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \mathbf{e} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \mathbf{e} \\ \varepsilon & \varepsilon & \varepsilon & \delta^{36} \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix} U, \\
 \gamma\delta^* &\preceq \frac{v_1 \bullet}{x_1} \odot \frac{v_{1-2} \bullet}{x_{2-1}}, \\
 \gamma\delta^* &\preceq \frac{v_2 \bullet}{x_2} \odot \frac{v_{2-1} \bullet}{x_{1-2}}, \\
 \gamma\delta^* &\preceq \frac{v_3 \bullet}{x_3} \odot \frac{v_{3-4} \bullet}{x_{4-3}}, \\
 \gamma\delta^* &\preceq \frac{v_4 \bullet}{x_4} \odot \frac{v_{4-3} \bullet}{x_{3-4}}.
 \end{aligned}$$

La séquence des tirs de u_4 est définie dans $\overline{\mathbb{Z}}_{\min}$ à travers sa fonction compteur par :

$$u_4(t) = 1u_4(t-2) \oplus \left\lfloor \frac{x_1(t)}{2} \right\rfloor \otimes \left\lfloor \frac{x_2(t)}{2} \right\rfloor \oplus 75v_4(t).$$

4.3.2.B Stratégie de commande

Dans le cadre de la réalisation d'une onde verte, les cycles de feux doivent être fixes, afin d'assurer la synchronisation des feux des deux intersections en cascade. Ainsi, pour cette stratégie de commande, les cycles sont définis comme suit :

Pour la première intersection, la durée de cycle optimale est donnée par [Cohen, 1993] :

$$c_1 = \frac{1.5T + 5}{1 - Y},$$

avec $T = 2s + 2\tau = 2 \times 3 + 2 \times 2$, soit 10 secondes et $Y = y_1 + y_2$ tel que

$$\begin{aligned} y_1 &= \frac{\lambda_1 \times 3600}{2000} , \\ y_2 &= \frac{\lambda_2 \times 3600}{2000} . \end{aligned}$$

Les durées de la phase de vert pour les rues 1 et 2 sont données respectivement par :

$$\begin{aligned} v_1 &= y_1 \frac{c_1 - T}{Y} , \\ v_2 &= y_2 \frac{c_1 - T}{Y} . \end{aligned}$$

De la même manière, nous avons, pour la deuxième intersection :

$$\begin{aligned} c_2 &= \frac{1.5T + 5}{1 - Y} , \\ v_3 &= y_3 \frac{c_2 - T}{Y} , \\ v_4 &= y_4 \frac{c_2 - T}{Y} . \end{aligned}$$

avec $T = 2s + 2\tau = 2 \times 3 + 2 \times 2$, soit 10 secondes et $Y = y_1 + y_2$ tel que

$$\begin{aligned} y_3 &= \frac{\lambda_3 \times 3600}{2000} , \\ y_4 &= \frac{\frac{\lambda_1 + \lambda_2}{2} \times 3600}{2000} . \end{aligned}$$

En revanche, dans notre approche, nous ne prévoyons aucune synchronisation des feux et la notion de cycle est sans objet. Les dates auxquelles les véhicules quittent la première intersection par la rue 4 sont simplement prises comme entrées pour la commande des feux de la deuxième intersection. Nous utilisons l'algorithme de programmation dynamique présenté dans la sous-section 4.2.2 page 126.

4.3.2.C Résultats de simulation

Nous nous intéressons plus particulièrement à la deuxième intersection car elle est en aval. Les figures 4.15 à 4.18 montrent l'évolution des temps moyens d'attente des véhicules.

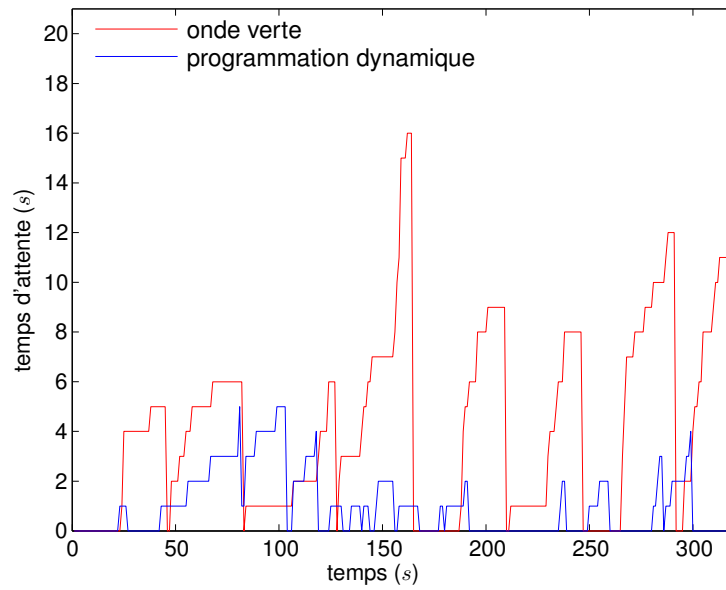


FIGURE 4.15 – Temps d’attente dans l’intersection aval d’un système de deux intersections en cascade ($\lambda_1 = 0.1veh.s^{-1}$, $\lambda_2 = 0.1veh.s^{-1}$, $\lambda_3 = 0.1veh.s^{-1}$).

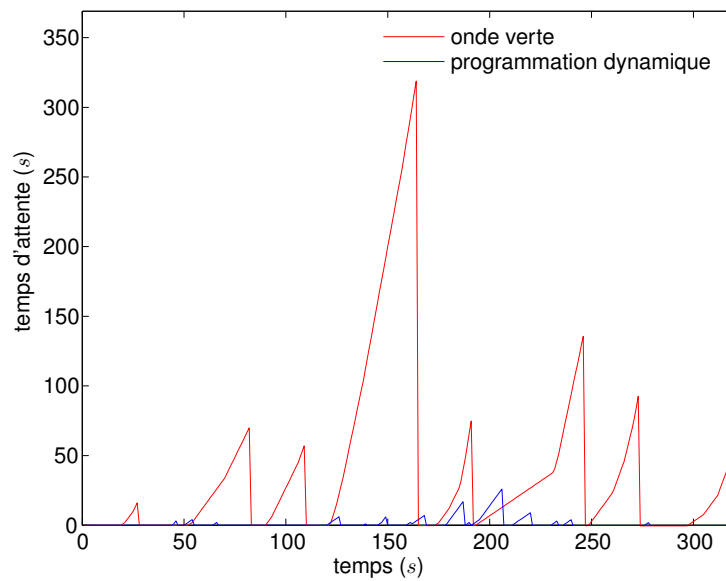


FIGURE 4.16 – Temps d’attente dans l’intersection aval d’un système de deux intersections en cascade ($\lambda_1 = 0.1veh.s^{-1}$, $\lambda_2 = 0.1veh.s^{-1}$, $\lambda_3 = 0.3veh.s^{-1}$).

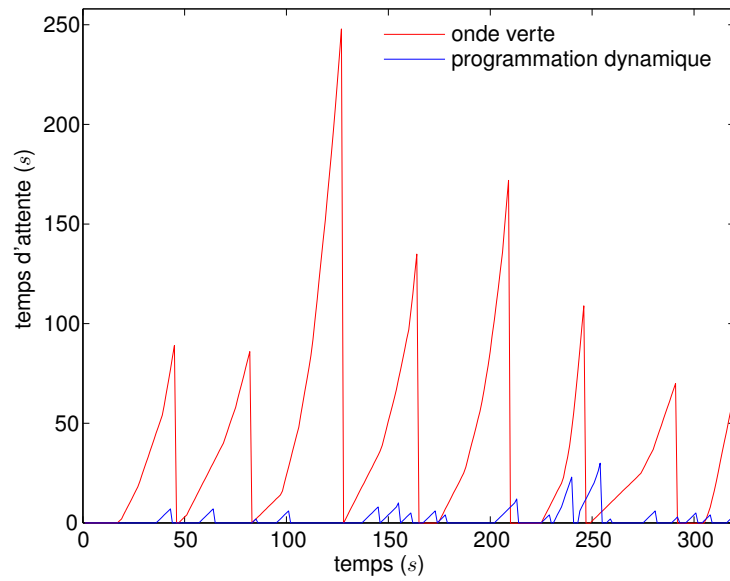


FIGURE 4.17 – Temps d’attente dans l’intersection aval d’un système de deux intersections en cascade ($\lambda_1 = 0.2veh.s^{-1}$, $\lambda_2 = 0.2veh.s^{-1}$, $\lambda_3 = 0.2veh.s^{-1}$).

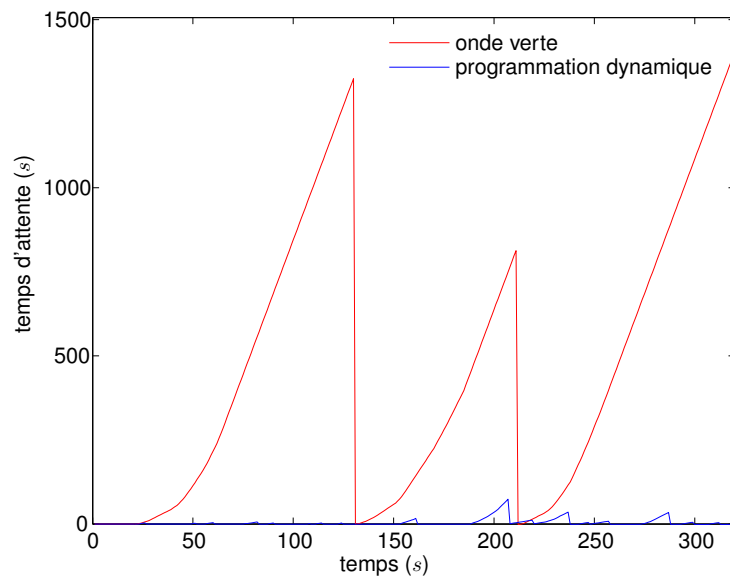


FIGURE 4.18 – Temps d’attente dans l’intersection aval d’un système de deux intersections en cascade ($\lambda_1 = 0.3veh.s^{-1}$, $\lambda_2 = 0.3veh.s^{-1}$, $\lambda_3 = 0.2veh.s^{-1}$).

Les figures 4.19 à 4.22 montrent l’évolution des longueurs de files d’attente de véhicules dans la deuxième intersection.

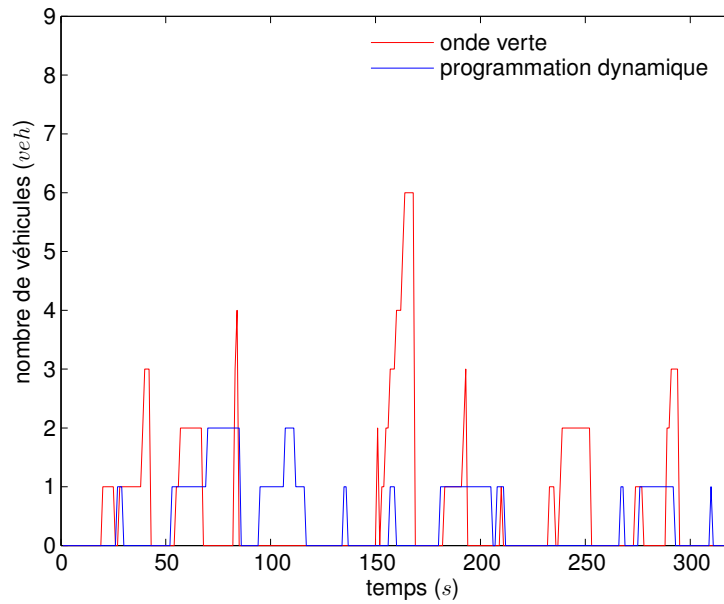


FIGURE 4.19 – Nombre de véhicules dans l’intersection aval d’un système de deux intersections en cascade ($\lambda_1 = 0.1veh.s^{-1}$, $\lambda_2 = 0.1veh.s^{-1}$, $\lambda_3 = 0.1veh.s^{-1}$).

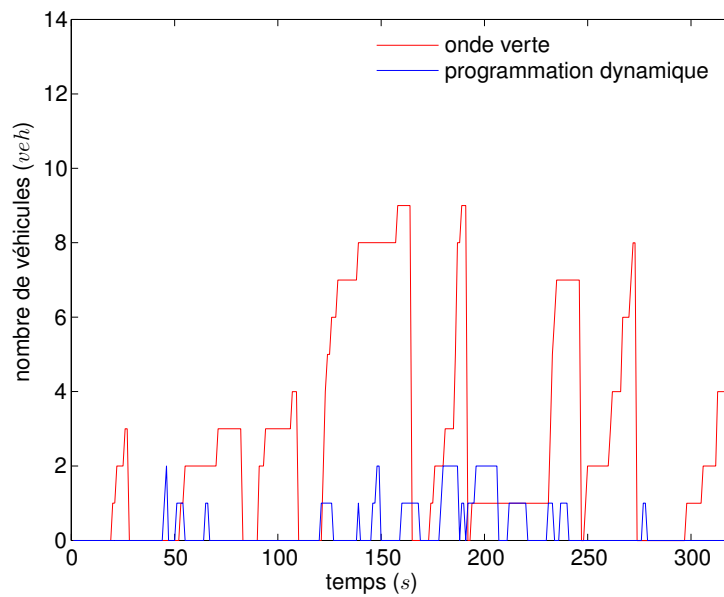


FIGURE 4.20 – Nombre de véhicules dans l’intersection aval d’un système de deux intersections en cascade ($\lambda_1 = 0.1veh.s^{-1}$, $\lambda_2 = 0.1veh.s^{-1}$, $\lambda_3 = 0.3veh.s^{-1}$).

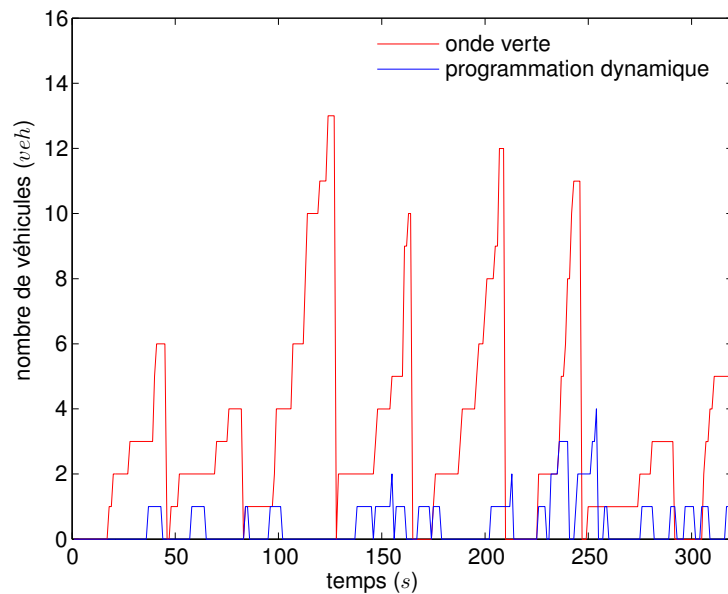


FIGURE 4.21 – Nombre de véhicules dans l’intersection aval d’un système de deux intersections en cascade ($\lambda_1 = 0.2veh.s^{-1}$, $\lambda_2 = 0.2veh.s^{-1}$, $\lambda_3 = 0.2veh.s^{-1}$).

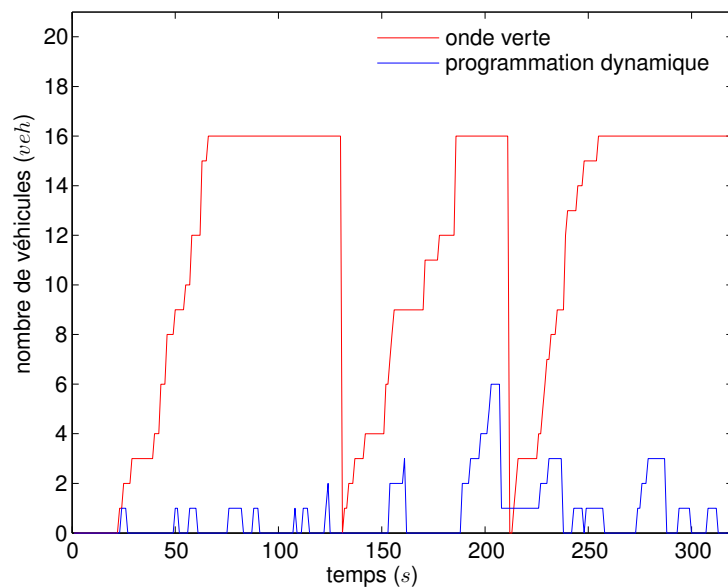


FIGURE 4.22 – Nombre de véhicules dans l’intersection aval d’un système de deux intersections en cascade ($\lambda_1 = 0.3veh.s^{-1}$, $\lambda_2 = 0.3veh.s^{-1}$, $\lambda_3 = 0.2veh.s^{-1}$).

Les simulations montrent que plus le trafic est dense, meilleur est notre algorithme vis-à-vis de l’onde verte. Notamment, les temps d’attente sont fortement diminués, de même que le nombre de véhicules circulant dans les zones de stockage. Ceci est d’autant plus surprenant qu’il est bien connu que l’optimisation locale est moins bonne que l’optimisation globale. Ainsi, ces

simulations nous montrent que la prise en compte des véhicules de manière individuelle donne de meilleurs résultats que lorsque le trafic est considéré comme un flot.

Conclusion

Nous avons proposé, dans ce chapitre, un modèle de trafic transversal, qui peut être utilisé avec les technologies d'aujourd'hui et de demain. Ce modèle est original car il prend en compte les véhicules de manière individuelle, alors que la grande majorité des modèles de trafic éprouvés, ainsi qu'une partie des approches plus récentes, considère le trafic comme un flot.

Nous avons montré que ce modèle, couplé à un algorithme de commande optimale en temps polynomial, donne des résultats probants vis-à-vis des méthodes et techniques les plus utilisées aujourd'hui pour la gestion d'un carrefour à feux. Nous avons également constaté, à travers un exemple simple, que notre optimisation locale donnait des résultats au moins aussi bons que la synchronisation de plans de feux fixes réalisant une onde verte.

Nos travaux ouvrent de nouvelles perspectives de recherche tant au niveau théorique que pratique. En effet, ils doivent être améliorés et enrichis sur plusieurs points. Notamment, il serait intéressant de pouvoir tenir compte des véhicules tournant à gauche. Plus précisément, nous ne modélisons que deux phases pour un carrefour composé de deux rues. Nous devons également étudier plus en profondeur le cas de plusieurs intersections en cascade.

Conclusion générale

Dans ce mémoire, nous nous sommes intéressés à la modélisation des conflits dans l'algèbre des dioïdes dans le but d'apporter une nouvelle vision du trafic urbain au niveau des carrefours. Nous avons d'abord passé en revue le vocabulaire de l'ingénierie du trafic. Puis nous avons rappelé les fondements de sa modélisation et de sa régulation. Nous avons également constaté que certaines approches très récentes sont encore basées sur des principes éprouvés. Ces principes reproduisent incontestablement le comportement du trafic, cependant les nouvelles technologies permettent sans doute d'aller au-delà de leur portée. Notre réflexion a ainsi porté sur la conception d'une nouvelle approche du trafic urbain.

Cette réflexion nous a amené à proposer un modèle basé sur l'algèbre des dioïdes pour décrire le comportement des véhicules dans une rue. Ce modèle puise son originalité dans le fait de ne pas considérer le trafic comme un flot. Il permet au contraire de prendre en compte les véhicules de manière individuelle. En effet, le trafic peut être considéré comme un système discret. D'une part, les véhicules sont des entités entières et indivisibles. D'autre part, le système est régi par des événements tels que les changements de feux.

Pour pouvoir développer une méthode de régulation générique et efficace, nous devons décrire l'évolution du système par un modèle algébrique. Puisque le trafic induit de nombreuses synchronisations, il nous a paru naturel d'utiliser l'algèbre des dioïdes, car elle permet de décrire l'évolution d'un tel système par des équations linéaires. Nous avons montré que notre modèle peut être comparé aux modèles de trafic de la littérature en termes d'évaluation de performance puisqu'il reflète bien les comportements du trafic.

Le modèle étant validé, nous nous sommes attaqués au problème des intersections à deux modes de fonctionnement, à savoir rouge et vert. Ces intersections peuvent être assimilées à un système de régulation simplifié où chaque véhicule est équipé de systèmes de géopositionnement et de régulation de vitesse automatiques. Les intersections induisent par nature des conflits. Or, lorsque nous tentons de décrire les fonctions compteurs ou dateurs des transitions d'un Réseau de Petri comportant des conflits à l'aide d'équations, nous nous heurtons au problème de l'estimation du marquage instantané, rendant impossible l'écriture d'une équation d'évolution récursive.

En posant formellement le Problème de Ressources Partagées Invariantes, nous sommes

parvenus à décrire l'ensemble des évolutions admissibles par les systèmes les incluant. Ceci nous a permis de considérer de manière exhaustive les comportements possibles, en dehors de toute solution particulière. Afin d'illustrer la souplesse de notre modèle en terme de commande, nous avons proposé, pour réguler le trafic au sein d'une intersection simplifiée, deux algorithmes d'ordonnancement inspirés des résultats de la recherche opérationnelle. Chaque approche est complétée par la simulation de deux exemples concrets.

Le résultat observé en simulation représente bien le comportement d'un carrefour simplifié. Par contre, ce modèle n'est pas conçu pour représenter une intersection contemporaine. En effet, la solution optimale suggère de laisser passer alternativement un véhicule de chaque rue sans tenir compte du laps temps perdu entre le vert et le rouge effectifs. Plus précisément, elle montre que le modèle ne prend en considération ni les temps d'accélération des véhicules après un changement de phase, ni les périodes de rouge intégral pendant lesquelles tous les feux sont rouges.

Dans le but de modéliser et réguler les intersections contemporaines, nous avons enrichi le modèle théorique, en définissant un problème plus général, le Problème de Ressources Partagées Invariantes avec Temps de Configuration. Ceci nous a conduit à étendre d'avantage les possibilités de modélisation de l'algèbre des dioïdes. Nous avons également proposé deux algorithmes de commande, qui minimisent les temps d'attente des véhicules en un temps polynomial.

Enfin, le fait de ne plus considérer le trafic comme un flot mais, au contraire, prendre en compte chaque véhicule individuellement, donne des résultats prometteurs. Ceci est étayé par les simulations de deux situations concrètes, dans lesquelles nous comparons les résultats de notre approche avec ceux obtenus par les approches dominantes. En particulier, il est bien connu que la synchronisation des feux de deux intersections en cascade donne de meilleurs résultats qu'une optimisation locale. Pourtant, notre approche appliquée individuellement à chaque intersection a généré des résultats probants vis à vis de la réalisation classique d'une onde verte.

Notre travail ouvre ainsi de nouvelles perspectives de recherche, tant sur les aspects théoriques que pratiques.

- Sur le plan théorique, nous avons étendu la portée de l'algèbre des dioïdes en terme de modélisation. Il est désormais possible de représenter le comportement de la grande majorité des systèmes incluant une ou plusieurs ressources partagées. L'écriture algébrique permet de faciliter la conception de nouvelles techniques de commande, notamment avec la théorie de la résiduation. Toutefois, des efforts doivent être fournis pour développer ces techniques.
- Du point de vue pratique, nous avons proposé une nouvelle vision du trafic qui permet d'améliorer les systèmes de gestion et de régulation actuellement en place, tout en anticipant sur les systèmes à venir. Les bons résultats obtenus pour des cas simples doivent être étendus à des situations plus complexes qu'il est possible de rencontrer dans nos villes.

Liste des tableaux

1.1	Quelques exemples de fonctions permettant d'obtenir le diagramme fondamental. a et b sont des paramètres à définir.	23
1.2	Modèles de simulation microscopiques multimodaux	25
1.3	Commutations des feux de signalisation.	36
2.1	Paramètres du modèle d'une section de rue.	59
2.2	Calcul des paramètres du modèle d'une section de rue.	59
2.3	Valeur des paramètres d'une rue découpée en sections.	64
2.4	Valeur des paramètres du modèle d'une rue découpée en sections.	65
3.1	Sémantique associée à la figure 3.1.	78
3.2	Sémantique utilisée pour le modèle RdP d'une gare de fret.	84
3.3	Dates dues pour la planification des trains dans une gare de fret.	85
3.4	Tableau de planification final d'une gare de fret.	89
3.5	Sémantique du modèle de la figure 3.7.	93
3.6	Informations sur la configuration du carrefour de la figure 3.9.	95
3.7	Résultats de simulation concernant la première fenêtre de temps.	98
4.1	Sémantique des figures 4.1 et 4.2.	115
4.2	Sémantique des figures 4.4 et 4.5	122
4.3	Notations utilisées pour décrire la commande optimale d'un PRPITC.	126

Liste des figures

1.1	Le diagramme fondamental donne le débit en fonction du taux d'occupation . . .	24
1.2	L'intersection : un carrefour simple, croisement de deux rues à sens unique . . .	26
1.3	Différentes conceptions des actions à considérer	31
1.4	Comparaison entre le saut et la sélection de phase	32
1.5	Modélisation d'un réseau urbain par distribution en différé	34
2.1	Exemple de RdP : la transition \mathcal{T}_3 est validée et peut être franchie	49
2.2	Un GET simple	51
2.3	Une section de rue.	58
2.4	Modèle GET d'une section de rue.	59
2.5	Modèle GET d'une section de rue éventuellement non vide au début de la simulation.	60
2.6	Une rue découpée en sections.	64
2.7	Modèle modulaire d'une rue.	64
2.8	Diagrammes fondamentaux du système.	66
2.9	Borne supérieure de temps de séjour en fonction du taux de remplissage.	67
2.10	Modèle GET d'une rue initialement vide.	70
2.11	Réalisation d'un précompensateur GET pour le modèle de la figure 2.10.	71
3.1	Modèle RdP d'un PRPI	78
3.2	Modèle scindé d'un PRPI	79
3.3	Modèle RdP d'une gare de fret.	84
3.4	Diagramme de Gantt d'une gare de fret.	90
3.5	Occupation des voies d'une gare de fret.	90
3.6	État d'utilisation des voies d'une gare de fret.	91
3.7	Modèle RdP d'une intersection élémentaire.	92
3.8	Modèle scindé d'une intersection élémentaire.	93
3.9	Rencontre de deux rues à une intersection élémentaire.	94
3.10	Modèle RdP de deux rues se rencontrant à une intersection élémentaire (1u.t.= 1s).	95
3.11	Modèle RdP de deux rues se rencontrant à une intersection élémentaire (1u.t.= 2s).	100
3.12	Simulation d'un carrefour élémentaire (1u.t.= 2s).	101

3.13	Diagramme fondamental d'un carrefour élémentaire.	102
3.14	Vitesse en fonction du taux d'occupation dans un carrefour élémentaire.	103
3.15	Système urbain en boucle fermée composé de deux rues et d'un carrefour.	104
3.16	Modèle GET d'une rue circulaire, présenté dans [Farhi <i>et al.</i> , 2005].	104
3.17	Modèle RdP d'un carrefour, présenté dans [Farhi <i>et al.</i> , 2005].	105
3.18	Diagramme fondamental d'un système urbain en boucle fermée, présenté dans [Farhi <i>et al.</i> , 2005].	106
3.19	Modèle RdP d'un système urbain en boucle fermée.	106
3.20	Simulation d'un système urbain en boucle fermée.	107
3.21	Diagramme fondamental d'un système urbain en boucle fermée.	108
3.22	Vitesse en fonction du taux d'occupation dans un système urbain en boucle fermée.	108
4.1	Modèle RdP d'un PRPITC.	114
4.2	Modèle scindé d'un PRPITC.	116
4.3	Les deux phases de feux sont séparées par une période de rouge intégral.	120
4.4	Modèle RdP d'une intersection avec rouge intégral.	121
4.5	Modèle scindé d'une intersection avec rouge intégral.	122
4.6	Séquences d'arrivées de véhicules dans une intersection avec rouge intégral.	129
4.7	Séquence optimale d'attribution de la zone de conflit aux véhicules dans une intersection avec rouge intégral.	129
4.8	Modèle RdP d'une intersection avec rouge intégral.	131
4.9	Temps d'attente dans une intersection avec rouge intégral (débits d'entrée : $0.1veh.s^{-1}$).	132
4.10	Longueur des files d'attente dans une intersection avec rouge intégral (débits d'entrée : $0.1veh.s^{-1}$).	133
4.11	Temps d'attente dans une intersection avec rouge intégral (débits d'entrée : $0.2veh.s^{-1}$).	133
4.12	Longueur des files d'attente dans une intersection avec rouge intégral (débits d'entrée : $0.2veh.s^{-1}$).	134
4.13	Deux intersections en cascade.	134
4.14	Modèle RdP de deux intersections avec rouge intégral en cascade.	135
4.15	Temps d'attente dans l'intersection aval d'un système de deux intersections en cascade ($\lambda_1 = 0.1veh.s^{-1}$, $\lambda_2 = 0.1veh.s^{-1}$, $\lambda_3 = 0.1veh.s^{-1}$).	138
4.16	Temps d'attente dans l'intersection aval d'un système de deux intersections en cascade ($\lambda_1 = 0.1veh.s^{-1}$, $\lambda_2 = 0.1veh.s^{-1}$, $\lambda_3 = 0.3veh.s^{-1}$).	138
4.17	Temps d'attente dans l'intersection aval d'un système de deux intersections en cascade ($\lambda_1 = 0.2veh.s^{-1}$, $\lambda_2 = 0.2veh.s^{-1}$, $\lambda_3 = 0.2veh.s^{-1}$).	139
4.18	Temps d'attente dans l'intersection aval d'un système de deux intersections en cascade ($\lambda_1 = 0.3veh.s^{-1}$, $\lambda_2 = 0.3veh.s^{-1}$, $\lambda_3 = 0.2veh.s^{-1}$).	139
4.19	Nombre de véhicules dans l'intersection aval d'un système de deux intersections en cascade ($\lambda_1 = 0.1veh.s^{-1}$, $\lambda_2 = 0.1veh.s^{-1}$, $\lambda_3 = 0.1veh.s^{-1}$).	140

-
- 4.20 Nombre de véhicules dans l'intersection aval d'un système de deux intersections en cascade ($\lambda_1 = 0.1veh.s^{-1}$, $\lambda_2 = 0.1veh.s^{-1}$, $\lambda_3 = 0.3veh.s^{-1}$). 140
- 4.21 Nombre de véhicules dans l'intersection aval d'un système de deux intersections en cascade ($\lambda_1 = 0.2veh.s^{-1}$, $\lambda_2 = 0.2veh.s^{-1}$, $\lambda_3 = 0.2veh.s^{-1}$). 141
- 4.22 Nombre de véhicules dans l'intersection aval d'un système de deux intersections en cascade ($\lambda_1 = 0.3veh.s^{-1}$, $\lambda_2 = 0.3veh.s^{-1}$, $\lambda_3 = 0.2veh.s^{-1}$). 141

Index

A

accélération moyenne, 18
algèbre
 (max, +), 47
 (min, +), 47
application
 bijective, 46
 injective, 46
 isotone, 46
 résiduable, 48
 résiduée, 48
 surjective, 46
arbre de décision, 31
arc, 49
Automate Cellulaire, 19

B

bipartite, 49
borne
 inférieure, 46
 supérieure, 46

C

carrefour, 27
 à feux, 27
coefficient d'équivalence, 17
composition du trafic, 18
compteur, voir fonction compteur
concentration, voir densité
correction neutre, 70
courant de mobiles, 27
coût, 30
cycle, 28
 fixe, 28

D

dateur, voir fonction dateur

débit, 20
demi-treillis
 inférieur, 46
 supérieur, 46
densité, 20
diagramme
 de Gantt, 90
 fondamental, 23
dioïde, 46
 commutatif, 47
 complet, 47
discrétisation
 événementielle, 25
 temporelle, 25
distribution en différé, 34

E

écart de temps inter-véhiculaire, 17
élément
 absorbant, 47
 identité, 47
 neutre, 47
ensemble
 fermé, 46
 ordonné, 46
équation de Bellman, 31
étoile de Kleene, 52

F

feu de circulation, voir signal tricolore
feu tricolore, voir signal tricolore
flot, 19
 stationnaire, 20
fonction
 compteur, 51
 dateur, 51

fonctionnement au plus tôt, 51

G

Graphe d'Événements Temporisés, 51

Graphe Marqué, 51

H

homomorphisme, 53

horizon de planification, 30

horizon glissant, 32

I

infrastructure, 16

injection canonique, 52

intersection, 27

invariant de place, 50

isomorphisme, 53

isovélocique, 23

J

jeton, 49

M

macro-régulation, 29

macroscopique, 19

majorant, 46

marquage, 49

initial, 49

vivant, 50

marque, 49

matrice

causale, 52

périodique, 53

rationnelle, 53

réalisable, 53

matrice de transfert, 52

méthode de régulation, 27

micro-régulation, 28

microscopique, 17

minorant, 46

mobile, 16

mode, 16

modèle

continu, 25

de poursuite, voir modèle véhicule-suiveur

véhicule-suiveur, 19

\oplus -morphisme, 53

\otimes -morphisme, 53

mouvement, 16, 27

multimodal, 24

multiprogrammation à temps fixe, voir programmation horaire

O

onde verte, 29

opérateur de décalage

événementiel, 51

temporel, 51

opération

associative, 47

commutative, 47

distributive, 47

idempotente, 47

P

pas d'échantillonnage, 25

période, 25

phase, 28

piéton, 16

place, 49

amont, 49

aval, 49

plan de feux, 28

Poids Lourd, 16

point

conflictuel, 27

primaire, 27

secondaire, 27

tertiaire, 27

fixe, 69

politique de commande, voir trajectoire de commande

précompensateur, 67

Problème de Complémentarité Linéaire Étendu, 36

problème de ressources partagées, 76

Problème de Ressources Partagées Invariantes, 77

- Problème de Ressources Partagées Invariantes avec Temps de Configuration, 115
- programmation horaire, 28
- projection, 52
causale, 52
- R**
- règle de première date due, 77
- régulation
acyclique, 28
d'écart de TIV, 18
- relation
antisymétrique, 46
fondamentale, 23
réflexive, 46
transitive, 46
- Réseau de Petri, 49
déterministe, 50
mort, 50
ordinaire, 50
 \mathcal{P} -temporisé, 50
stochastique, 50
 \mathcal{T} -temporisé, 50
vivant, 50
- restriction, 52
- route, 27
- S**
- section critique, 78
- séquence de tirs, 50
- séquence d'occurrences, voir séquence de tirs
- série
causale, 52
périodique, 53
rationnelle, 53
- signal tricolore, 27
- sous-dioïde, 47
- stationnaire, 21
- stock and forward, 34
- stratégie de régulation, voir méthode de régulation
- système
à temps contraint, 68
à temps critique, 68
- Système à Ressources Partagées Invariantes, 77
- Système à Ressources Partagées Invariantes avec Temps de Configuration, 115
- Système dynamique à Événements Discrets, 39
- T**
- taux d'occupation, 21
- temporisation, 50
- temps
adaptatif, 29
de cycle, 28
discrétisé, 22
échantillonné, voir temps discrétisé
réel, voir temps adaptatif
- Temps Inter-Véhiculaire, 18
- théorie de la résiduation, 47
- tir de transition, 50
impulsionnel, 53
- Traffic-responsive Urban Control, 34
- trafic, 16
- trajectoire de commande, 30
- transformée
en γ , 51
en δ , 51
- transition, 49
franchie, voir transition tirée
morte, 50
puits, 50
source, 50
tirée, 50
validée, 50
- treillis, 46
- U**
- unité de voiture particulière, 16
- usager, 16
- V**
- vecteur
de sortie, 52
d'entrée, 52
d'état, 52

véhicule, [16](#)

gêné, voir véhicule suiveur

libre, [19](#)

suiveur, [19](#)

Véhicule Léger, [16](#)

vitesse

désirée, voir vitesse libre

instantanée, [18](#)

libre, [23](#)

moyenne

de l'approche microscopique, [18](#)

d'un flot, [23](#)

spatiale, [22](#)

temporelle, [21](#)

Z

zone

de conflit, [27](#)

de stockage, [27](#)

Bibliographie

- [Al Saba *et al.*, 2006] M. Al Saba, J.-L. Boimond et S. Lahaye (2006). Sur la commande des systèmes flexibles de production manufacturière par l’algèbre des dioïdes, Dans *Conférence Internationale Francophone d’Automatique (CIFA)*, Bordeaux, France. Cité page 77.
- [Anonyme, 1991] Anonyme (décembre 1991). *Régulation du trafic routier, feux de circulation, terminologie*. Association Française de Normalisation (AFNOR). Cité page 26.
- [Anonyme, 2002] — (2002). *Instruction interministérielle sur la signalisation routière, Livre I^e*. Ministère de l’intérieur et Ministère de l’équipement, du logement, des transports et de l’espace, chapitre « 6^e partie : feux de circulation permanents ». Cité page 27.
- [Anonyme, 2004] — (septembre 2004). *Journal Officiel*. Direction des Journaux Officiels, chapitre « Décret numéro 2004-998, article 1 », code de la Route (partie réglementaire), article R110-2. Cité page 25.
- [Aw et Rascle, 2000] A. Aw et M. Rascle (2000). Resurrection of « second order » models of traffic flow ?, *SIAM Journal on Applied Mathematics*, volume 60, numéro 3, pp. 916–938. Cité page 22.
- [Baccelli *et al.*, 1992] F. Baccelli, G. Cohen, G.-J. Olsder et J.-P. Quadrat (1992). *Synchronization and linearity, an algebra for discrete event systems*. Wiley. Disponible sur <http://www-rocq.inria.fr/metalau/cohen/SED/book-online.html>. Cité pages 46, 47, 52, 53, 54, 60, 62, 76, 81 et 118.
- [Barriere *et al.*, 1986] J.-F. Barriere, J.-L. Farges et J.-J. Henry (1986). Decentralization vs. hierarchy in optimal traffic control, Dans *5^e IFAC/IFIP/IFORS Symposium on Control in Transportation Systems (CTS)*, Vienna, Austria. Cité page 33.
- [Bhourri, 2002] N. Bhourri (février 2002). Intermodalité : bilan et perspectives des systèmes informatiques, Laboratoire Génie des Réseaux de Transport et Informatique Avancée (GRE-TIA, INRETS), Rapport de convention DTT-INRETS. Cité pages 24 et 25.
- [Biham et Middleton, 1992] O. Biham et A. A. Middleton (1992). Self-organization and a dynamical transition in traffic-flow models, *Physical Review Part A : Atomic, Molecular and Optical Physic*, volume 46, numéro 10, pp. R6124–R6127. Cité page 19.
- [Bingham, 2001] E. Bingham (2001). Reinforcement learning in neurofuzzy traffic signal control, *European Journal of Operational Research*, volume 131, pp. 232–241. Cité page 31.
- [Birkhoff, 1967] G. Birkhoff (1967). *Lattice theory*, 3^e édition. Providence. Cité page 46.

- [Blyth et Janowitz, 1972] T. S. Blyth et M. F. Janowitz (1972). *Residuation theory*. Pergamon Press. Cité page 47.
- [Braban et Boillot, 2003] C. Braban et F. Boillot (mai 2003). *Les systèmes adaptatifs de régulation du trafic en milieu urbain, synthèse*. Institut National de REcherche sur les Transports et leur Sécurité (INRETS), numéro 44. Cité page 28.
- [Bretherton, 1989] R. D. Bretherton (septembre 1989). SCOOT urban traffic control system : philosophy and evaluation, Dans *IFAC/IFIP/IFORS Symposium on Control, Computers, and Communications in Transportation (CCCT)*, pp. 237–239. Cité page 32.
- [Bretherton et Bowen, 1990] R. D. Bretherton et G. T. Bowen (mai 1990). Recent enhancements to SCOOT : SCOOT version 2.4, Dans *3^e IEEE international conference on Road Traffic Control (RTC)*, London, United Kingdom, pp. 95–98. Cité page 32.
- [Brucker, 2006] P. Brucker (février 2006). *Scheduling algorithms*, 5^e édition. Springer-Verlag. Cité page 126.
- [Brucker et Knust, 1999] P. Brucker et S. Knust (1999). Complexity results for single-machine problems with positive finish-start time-lags, *Computing*, volume 63, numéro 4, pp. 299–316. Cité pages 96 et 127.
- [Buisson *et al.*, 1995] C. Buisson, J.-P. Lebacque et J.-B. Lesort (1995). Macroscopic modelling of traffic flow and assignment in mixed networks, Dans *Conference of the International Society for Computing in Civil and Building Engineering (ISCCBE)*, Rotherdam, UK, pp. 1367–1374. Cité page 22.
- [Chandler *et al.*, 1958] R. E. Chandler, R. Herman et E. Montroll (avril 1958). Traffic dynamics : studies in car following, *Operations Research*, volume 6, numéro 2, pp. 165–184. Cité page 19.
- [Chin *et al.*, 1999] D. C. Chin, J. C. Spall et R. H. Smith (1999). Evaluation of system-wide traffic signal control using stochastic optimization and neural networks, Dans *American Control Conference (ACC)*, volume 3, pp. 2188–2194. Cité page 31.
- [Cohen *et al.*, 1991] G. Cohen, S. Gaubert, R. Nikoukhah et J.-P. Quadrat (décembre 1991). Second order theory of min-linear systems and its application to discrete event systems, Dans *IEEE international Conference on Decision and Control (CDC)*, Brighton, England. Cité pages 48, 53, 55, 56, 62 et 80.
- [Cohen *et al.*, 1989] G. Cohen, P. Moller, J.-P. Quadrat et M. Viot (1989). Algebraic tools for the performance evaluation of discrete event systems, *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, volume 77, numéro 1, pp. 39–58. Cité page 51.
- [Cohen, 1993] S. Cohen (1993). *Ingénierie du trafic routier. Éléments de théorie du trafic et applications*, nouvelle édition. Presses de l'École Nationale des Ponts et Chaussées. Cité pages 16, 17, 19, 28, 58, 66, 132 et 136.
- [Corréia, 2004] A. Corréia (juillet 2004). Étude des graphes d'événements temporisés dans l'algèbre des dioïdes : synthèse d'un modèle de référence avec contrainte de temps critique, Mémoire de master, École Centrale de Nantes, Université de Nantes, France. Cité pages 56 et 67.

- [Corréia *et al.*, 2005] A. Corréia, A. Abbas Turki, R. Bouyekhf et A. El Moudni (septembre 2005). A (max, +)-linear model for the analysis of urban traffic networks, Dans *10^e IEEE international conference on Emerging Technologies and Factory Automation (ETFA)*, volume 2, Catania, Italy. Cité page 58.
- [Corréia *et al.*, 2006a] — (août 2006). Controlling a model of traffic lights based on dioids, Dans *11^e IFAC/IFIP/IFORS Symposium on Control in Transportation Systems (CTS)*, Delft, The Netherlands. Cité page 96.
- [Corréia *et al.*, 2006b] — (mai 2006). Modelling urban intersections in dioid algebra, *WSEAS transactions on systems*, volume 5, pp. 1223–1228. Cité page 93.
- [Corréia *et al.*, 2007a] — (juillet 2007). Controlling a dioid model of traffic light : a scheduling problem, Dans *European Control Conference (ECC)*. Cité page 99.
- [Corréia *et al.*, 2007b] — (2007). A dioid model for invariant resource sharing problems, à paraître dans *IEEE Transactions on Systems, Man and Cybernetics (SMC) – part A : Systems and Humans*. Cité page 78.
- [Corréia *et al.*, 2007c] — (août 2007). Modeling a flow-shop with setup times in dioid algebra, Dans *IEEE international conference on Service Operations and Logistics, and Informatics (SOLI)*, Philadelphia, USA. Cité page 115.
- [Corréia *et al.*, 2006c] A. Corréia, S. Lahaye et B. Cottenceau (mai 2006). Commande de graphes d'événements temporisés partiellement commandables avec contraintes temporelles, Dans *Conférence Internationale Francophone d'Automatique (CIFA)*, Bordeaux, France, actes sur CD-ROM. Cité pages 56, 57, 67 et 68.
- [Cottenceau, 1999] B. Cottenceau (1999). Contribution à la commande de systèmes à événements discrets : synthèse de correcteurs pour les graphes d'événements temporisés dans les dioïdes, Thèse de doctorat, Institut des Sciences et Techniques pour l'Ingénieur d'Angers (ISTIA), Université d'Angers, France. Cité pages 67 et 85.
- [Cottenceau *et al.*, 2001] B. Cottenceau, L. Hardouin, J.-L. Boimond et J.-L. Ferrier (mars 2001). Model reference control for timed event graphs in dioids, *Automatica*, volume 37, pp. 1451–1458. Cité pages 52 et 67.
- [Cremer et Ludwig, 1986] M. Cremer et J. Ludwig (1986). A fast simulation model for traffic flow on the basis of Boolean operations, *Mathematics and Computers in Simulation*, volume 28, pp. 297–303. Cité page 19.
- [Cunninghame-Green, 1979] R. Cunninghame-Green (1979). Minimax algebra, Dans *Lecture notes in economics and mathematical systems*. Springer-Verlag, volume 166. Cité page 46.
- [Davey et Priestley, 1990] B. A. Davey et H. A. Priestley (1990). *Introduction to lattices and order*. Cambridge University Press. Cité pages 46 et 69.
- [David et Alla, 1992] R. David et H. Alla (1992). *Du GraFCET aux réseaux de Petri*. Hermès. Cité pages 38 et 76.
- [De Schutter et De Moor, 1997] B. De Schutter et B. De Moor (décembre 1997). Optimal traffic light control for a single intersection, Dans *International symposium on NonLinear Theory and its Applications (NOLTA)*, Honolulu, Hawaii, pp. 1085–1088. Cité pages 35 et 36.

- [Desrochers et Al-Jaar, 1995] A. A. Desrochers et R. Y. Al-Jaar (1995). *Applications of Petri nets in manufacturing systems : modeling, control, and performance analysis*. Press of the Institute of Electrical and Electronics Engineers (IEEE). Cité page 76.
- [Di Taranto et Mauro, 1989] C. Di Taranto et V. Mauro (1989). *UTOPIA*, Dans *6^e IFAC/IFIP/IFORS Symposium on Control in Transportation Systems (CTS)*, Paris, France, pp. 575–597. Cité page 33.
- [Diakaki, 1999] C. Diakaki (1999). *Integrated control of traffic flow in corridor networks*, Thèse de doctorat, Technical University of Crete, Chania, Greece. Cité page 34.
- [Dinopoulou *et al.*, 2000] V. Dinopoulou, C. Diakaki et M. Papageorgiou (octobre 2000). *Simulation investigations of the traffic-responsive urban control strategy TUC*, Dans *IEEE Intelligent Transportation Systems Conference (ITS)*, Dearborn (MI), USA. Cité pages 33, 34 et 35.
- [Dion et Hellinga, 2001] F. Dion et B. Hellinga (2001). *A methodology for obtaining signal coordination within a distributed real-time network signal control system with transit priority*, Dans *80^e Annual Meeting of the Transportation Research Board (TRB)*. Cité page 32.
- [Dion et Hellinga, 2002] — (2002). *A rule-based real-time traffic responsive signal control system with transit priority : Application to an isolated intersection*, *Transportation Research Part B : Methodological*, volume 36, pp. 325–343. Cité page 32.
- [Farhi *et al.*, 2005] N. Farhi, M. Goursat et J.-P. Quadrat (décembre 2005). *Derivation of the fundamental traffic diagram for two circular roads and a crossing using minplus algebra and Petri net modeling*, Dans *44^e IEEE international Conference on Decision and Control (CDC)*, Sevilla, Spain. Cité pages 40, 75, 103, 104, 105, 106, 109 et 148.
- [Farhi *et al.*, 2007] — (juillet 2007). *Fundamental traffic diagrams of elementary road networks*, Dans *European Control Conference (ECC)*, Kos, Greece, pp. 4944–4951. Cité pages 40 et 75.
- [Foraste et Scemama, 1987] B. Foraste et G. Scemama (juillet 1987). *An « expert system » approach to congestion*, *Institut National de REcherche sur les Transports et leur Sécurité (INRETS)*, volume 2, pp. 35–40. Cité page 29.
- [Fukui et Ishibashi, 1996] M. Fukui et Y. Ishibashi (1996). *Traffic flow in 1D cellular automata model including cars moving with high speed*, *Journal of Physics Society*, volume 65, pp. 1859–1870. Cité page 19.
- [Gantt, 1919] H. L. Gantt (août 1919). *Organizing for work*, *Industrial Management*, volume 58, pp. 89–93. Cité page 89.
- [Gartner, 1983] N. H. Gartner (1983). *OPAC : a demand-responsive strategy for traffic signal control*, *Transportation Research Record (journal of the Transportation Research Board)*, volume 906, pp. 75–81. Cité page 33.
- [Gartner *et al.*, 1995] N. H. Gartner, C. Stamatiadis et P. J. Tarnoff (1995). *Development of advanced traffic signal control strategies for intelligent transportation systems : multilevel design*, *Transportation Research Record (journal of the Transportation Research Board)*, volume 1494, pp. 98–105. Cité page 33.

- [Gartner *et al.*, 1999] N. H. Gartner, P. J. Tarnoff et C. M. Andrews (1999). Evaluation of the optimized policies for adaptive signal control strategy, *Transportation Research Record (journal of the Transportation Research Board)*, volume 1683, pp. 105–114. Cité page 33.
- [Gaubert, 1992] S. Gaubert (1992). Théorie des systèmes linéaires dans les dioïdes, Thèse de doctorat, Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt, France. Cité page 55.
- [Gaubert, 1995] — (décembre 1995). Performance evaluation of $(\max, +)$ automata, *IEEE Transactions on Automatic Control*, volume 40, numéro 12, pp. 2014–2025. Cité page 62.
- [Gaubert et Mairesse, 1999] S. Gaubert et J. Mairesse (avril 1999). Modeling and analysis of timed Petri nets using heaps of pieces, *IEEE Transactions on Automatic Control*, volume 44, numéro 4, pp. 683–697. Cité page 77.
- [Gazis *et al.*, 1961] D. C. Gazis, R. Herman et R. W. Rothery (1961). Non-linear car-follow-leader models of traffic flow, *Operations Research*, volume 9, numéro 4, pp. 545–567. Cité page 19.
- [Gerlough, 1956] D. L. Gerlough (1956). Simulation of freeway traffic by an electronic computer, Dans *Annual Meeting of the Transportation Research Board (TRB)*, volume 35, Washington DC, USA, pp. 543–547. Cité page 19.
- [Ghosh et Gupta, 1997] J. B. Ghosh et J. N. D. Gupta (1997). Batch scheduling to minimize maximum lateness, *Operations research letters*, volume 21, pp. 77–80. Cité page 126.
- [Giua *et al.*, 1992] A. Giua, F. Di Cesare et M. Silva (octobre 1992). Generalized mutual exclusion constraints on nets with uncontrollable transitions, Dans *IEEE international conference on Systems, Man and Cybernetics (SMC)*, volume 2, Chicago, IL, USA, pp. 974–979. Cité page 76.
- [Graham *et al.*, 1979] R. L. Graham, E. L. Lawler, J. K. Lenstra et A. H. G. Rinnooy Kan (1979). Optimization and approximation in deterministic sequencing and scheduling : a survey, *Annals of discrete mathematics*, volume 5, pp. 287–326. Cité pages 76, 96 et 127.
- [Hadamard, 1893] J. Hadamard (1893). Résolution d’une question relative aux déterminants, *Bulletin des Sciences Mathématiques*, volume 17, pp. 240–246, rédigé par M. Darboux. Cité page 61.
- [Henry et Farges, 1989] J.-J. Henry et J.-L. Farges (1989). PRODYN, Dans *6^e IFAC/IFIP/IFORS Symposium on Control, Computers, and Communications in Transportation (CCCT)*, Paris, France. Cité page 33.
- [Henry *et al.*, 1983] J.-J. Henry, J.-L. Farges et J. Tuffal (1983). The PRODYN real time traffic algorithm, Dans *4^e IFAC/IFIP/IFORS Symposium on Control in Transportation Systems (CTS)*, Baden-Baden, Germany, pp. 307–312. Cité page 33.
- [Herman et Potts, 1959] R. Herman et R. B. Potts (1959). Single lane traffic theory and experiment, Dans *Symposium on theory of traffic flow*, pp. 147–157. Cité page 19.
- [Houssin, 2006] L. Houssin (décembre 2006). Contribution à la commande des systèmes $(\max, +)$ -linéaires : applications aux réseaux de transport, Thèse de doctorat, Institut des Sciences et Techniques pour l’Ingénieur d’Angers (ISTIA), Université d’Angers, France. Cité page 69.

- [Hua et Faghri, 1995] J. Hua et A. Faghri (juillet 1995). Development of neural signal control system – toward intelligent traffic signal control, *Transportation Research Record (journal of the Transportation Research Board)*, volume 1497, pp. 53–61. Cité page 31.
- [Jackson, 1955] J. R. Jackson (juillet 1955). Scheduling a production line to minimize maximum tardiness, University of California, Los Angeles, Research report 43. Cité pages 77, 85 et 124.
- [Kamata et Oda, 1991] J. Kamata et T. Oda (1991). *Concise encyclopædia of traffic and transportation systems*. Pergamon Press, chapitre « Detectors for road traffic », pp. 96–101. Cité page 16.
- [Kumar et Garg, 1995] R. Kumar et V. K. Garg (1995). Extremal solutions of inequations over lattices with applications to supervisory control, *Theoretical computer science*, volume 148, pp. 67–92. Cité page 69.
- [Lahaye et al., 2004] S. Lahaye, B. Cottenceau et A. Corrêia (2004). Commande de graphes d'événements temporisés avec contrainte de temps critique, Dans *Conférence Internationale Francophone d'Automatique (CIFA)*. Cité pages 56, 57 et 67.
- [Lamport, 1978] L. Lamport (juillet 1978). Time, clocks, and the ordering of events in a distributed system, *Association for Computing Machinery (ACM)*, volume 7, numéro 21, pp. 558–565. Cité page 76.
- [Lassez et al., 1982] J.-L. Lassez, V. L. Nguyen et L. Sonenberg (1982). Fixed point theorems and semantics : a folk tale, *Information Processing Letters*, volume 14, numéro 3, pp. 112–116. Cité page 69.
- [Lawler, 1977] E. L. Lawler (1977). A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness, *Annals of Discrete Mathematics*, volume 1, pp. 331–342. Cité pages 96, 97 et 165.
- [Lee et Lee-Kwang, 1999] J. H. Lee et H. Lee-Kwang (1999). Distributed and cooperative fuzzy controllers for traffic intersections group, *IEEE Transactions on Systems, Man and Cybernetics (SMC) – part C : Applications and Reviews*, volume 29, numéro 2, pp. 263–271. Cité page 31.
- [Leung et al., 2001] A. Leung, K. V. Palem et A. Pnueli (2001). Scheduling time-constrained instructions on pipelined processors, *Programming languages and systems*, volume 23, numéro 1, pp. 73–103. Cité pages 98, 99 et 166.
- [Lhommeau, 2003] M. Lhommeau (2003). Étude de systèmes à événements discrets dans l'algèbre $(\max, +)$: Synthèse de correcteurs robustes dans un dioïde d'intervalles. Synthèse de correcteurs en présence de perturbations., Thèse de doctorat, Institut des Sciences et Techniques pour l'Ingénieur d'Angers (ISTIA), Université d'Angers, France. Cité pages 52 et 67.
- [Lighthill et Whitham, 1955] M. J. Lighthill et G. B. Whitham (1955). On kinematic waves, II : a theory on traffic flow on roads, *Proceedings of the Royal Society*, volume 229A, pp. 317–345. Cité page 22.
- [List et Cetin, 2004] G. F. List et M. Cetin (septembre 2004). Modeling traffic signal control using Petri nets, *IEEE Transactions on Intelligent Transportation Systems*, volume 5, numéro 3, pp. 177–187. Cité pages 38 et 39.

- [Lotito *et al.*, 2005] P. Lotito, E. Mancinelli et J.-P. Quadrat (2005). A min-plus derivation of the fundamental car-traffic law, *IEEE Transactions on Systems, Man and Cybernetics (SMC) – part A : Systems and Humans*, volume 50, numéro 5, pp. 699–705. Cité page 75.
- [Maia *et al.*, 2003] C. A. Maia, L. Hardouin, R. Santos-Mendes et B. Cottenceau (décembre 2003). Optimal closed-loop control of timed event graphs in dioids, *IEEE Transactions on Automatic Control*, volume 48, numéro 12, pp. 2284–2287. Cité page 67.
- [Mancinelli *et al.*, 2001] E. Mancinelli, G. Cohen, S. Gaubert, J.-P. Quadrat et E. Rofman (septembre 2001). On traffic light control of regular towns, Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt, France, Rapport technique. Cité pages 38, 39, 40 et 75.
- [Mirchandani et Head, 2001] P. Mirchandani et L. Head (2001). A real-time traffic signal control system : architecture, algorithms and analysis, *Transportation Research Part C : Emerging Technologies*, volume 9, pp. 415–432. Cité page 33.
- [Moody et Antsaklis, 2000] J. O. Moody et P. J. Antsaklis (mars 2000). Petri net supervisors for DES with uncontrollable and unobservable transitions, Dans *IEEE Transactions on Automatic Control*, volume 45, numéro 3, pp. 462–476. Cité page 76.
- [Moody *et al.*, 1995] J. O. Moody, P. J. Antsaklis et M. D. Lemmon (octobre 1995). Automated design of a Petri net feedback controller for a robotic assembly cell, Dans *IEEE international conference on Emerging Technologies and Factory Automation (ETFA)*, volume 2, Paris, France, pp. 117–128. Cité page 76.
- [Morrish, 1980] D. W. Morrish (septembre 1980). Area traffic in Bordeaux : a contrast with British practice, *Traffic Engineering and Control*, volume 21, numéro 8, pp. 433–437. Cité page 29.
- [Murata, 1989] T. Murata (avril 1989). Petri nets : properties, analysis and applications, *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, volume 77, numéro 4, pp. 541–580. Cité pages 38, 49, 59 et 76.
- [Naimi *et al.*, 1996] M. Naimi, M. Trehel et A. Arnold (1996). A $\log(N)$ distributed mutual exclusion algorithm based on path reversal, *Journal of parallel and distributed computing*, volume 34, pp. 1–13. Cité page 76.
- [Newell, 1998] G. F. Newell (1998). The rolling horizon scheme of traffic control, *Transportation Research Part A : Policy and Practice*, volume 32, pp. 39–44. Cité page 30.
- [Niittymäki, 2001] J. Niittymäki (2001). Installation and experience of field testing a fuzzy signal controller, *European Journal of Operational Research*, volume 131, pp. 273–281. Cité page 31.
- [Papageorgiou et Schmidt, 1991] M. Papageorgiou et G. Schmidt (1991). *Concise encyclopædia of traffic and transportation systems*. Pergamon Press, chapitre « Freeway traffic modeling », pp. 162–167. Cité page 22.
- [Park *et al.*, 2000] B. B. Park, C. J. Messer et T. Urbanik (2000). Enhanced genetic algorithm for signal-timing optimization of oversaturated intersections, *Transportation Research Record (journal of the Transportation Research Board)*, volume 1727, pp. 32–41. Cité page 31.

- [Payne, 1971] H. J. Payne (1971). Models of freeway traffic and control, *Mathematical Models of Public Systems*, volume 28, numéro 1, pp. 51–61. Cité page 22.
- [Petri, 1962] C. A. Petri (1962). Kommunikation mit Automaten, Thèse de doctorat, Institut für Instrumentelle Mathematik, Bonn. Cité pages 38 et 49.
- [Pipes, 1953] L. A. Pipes (1953). An operational analysis of traffic dynamics, *Journal of Applied Physics*, volume 3, numéro 24, pp. 274–281. Cité page 19.
- [Porche, 1998] I. R. Porche (1998). Dynamic traffic control : decentralized and coordinated methods, Thèse de doctorat, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan. Cité page 32.
- [Porche *et al.*, 1996] I. R. Porche, M. Sampath, R. Sengupta, Y.-L. Chen et S. Lafortune (1996). A decentralized scheme for real-time optimization of traffic signals, Dans *IEEE International Conference on Control Applications*. Cité page 32.
- [Potts et Kovalyov, 2000] C. N. Potts et M. Y. Kovalyov (janvier 2000). Scheduling with batching : a review, *European journal of operational research*, volume 120, numéro 2, pp. 228–249, invited review. Cité pages 124, 126 et 127.
- [Reuschel, 1950] A. Reuschel (1950). Fahrzeugbewegungen in der Kolonne bei gleichförmig beschleunigtem oder verzögertem Leitfahrzeug, *Zeitschrift des Österreichischen Ingenieur und Architektur Vereines*, volume 95, pp. 59–62 et 73–77. Cité page 19.
- [Scemama, 2006] G. Scemama (2006). CLAIRE++, un système indépendant de supervision de la congestion du trafic routier, fiche de présentation, Laboratoire Génie des Réseaux de Transport et Informatique Avancée (GRETIA, INRETS). Cité page 28.
- [Sen et Head, 1997] S. Sen et K. L. Head (1997). Controller optimization of phases at an intersection, *Transportation Science*, volume 3, pp. 5–17. Cité page 33.
- [Shelby, 2004] S. G. Shelby (2004). Single intersection evaluation of real-time adaptative traffic signal control algorithms, Dans *84^e Annual Meeting of the Transportation Research Board (TRB)*. Cité page 30.
- [Sims et Finlay, 1984] A. G. Sims et A. B. Finlay (1984). SCATS, Splits and Offsets Simplified (S.O.S.), Dans *Proceedings of Australian Road Research Board Group (ARRB)*, volume 12, numéro 4. Cité page 32.
- [Tarski, 1955] A. Tarski (1955). A lattice-theoretical fixpoint theorem and its applications, *Pacific journal of mathematics*, volume 5, numéro 2, pp. 285–309. Cité page 69.
- [Tolba, 2004] C. Tolba (2004). Contribution à l'utilisation des réseaux de Petri pour la modélisation et la régulation du trafic urbain et inter-urbain, Thèse de doctorat, Université de Technologie de Belfort-Montbéliard (UTBM). Cité pages 38, 39 et 58.
- [Tolba *et al.*, 2001] C. Tolba, D. Lefebvre, P. Thomas et A. El Moudni (2001). Continuous Petri nets models for the analysis of traffic urban networks, Dans *IEEE international conference on Systems, Man and Cybernetics (SMC)*, Tucson, Arizona, pp. 1323–1328. Cité page 39.
- [Tolba *et al.*, 2003] — (2003). Performances evaluation of the traffic control in a single crossroad by Petri nets, Dans *IEEE international conference on Emerging Technologies and Factory Automation (ETFA)*, volume 2, Lisbon, Portugal, pp. 157–160. Cité page 39.

- [Tolba *et al.*, 2004] — (novembre 2004). Approche multi-modèles pour la commande des feux de trafic, Dans *Conférence Internationale Francophone d'Automatique (CIFA)*, Douz, Tunisie. Cité page 39.
- [Transportation Research Board, 2000] Transportation Research Board (2000). *Highway capacity manual*, 4^e édition. Transportation Research Board. Cité pages 15, 22 et 28.
- [Trouillet et Benasser, 2002] B. Trouillet et A. Benasser (octobre 2002). Cyclic scheduling problems with assemblies : an approach based to the search of an initial marking in a marked graph, Dans *IEEE international conference on Systems, Man and Cybernetics (SMC)*, volume 3, Hammamet, Tunisia. Cité page 76.
- [Trouillet et Gentina, 2003] B. Trouillet et J.-C. Gentina (2003). Sur la transformation d'une classe de réseau de Petri en graphe d'événements pondéré par intégration de contraintes, Dans *Modélisation des Systèmes Réactifs (MSR)*. Cité page 76.
- [van den Boom et De Schutter, 2006] T. J. J. van den Boom et B. De Schutter (2006). Modelling and control of discrete event systems using switching max-plus-linear systems, *Control engineering practice*, volume 14, numéro 10, pp. 1199–1211. Cité pages 77, 78 et 115.
- [van Egmond et Olsder, 1998] R.-J. van Egmond et G.-J. Olsder (août 1998). The $(\max, +)$ algebra applied to synchronization of traffic light processes, Dans *International Workshop On Discrete Event Systems (WODES)*, Cagliari, Italy, pp. 193–198. Cité page 75.
- [Wardrop, 1952] J. G. Wardrop (1952). Some theoretical aspects of road traffic research, *Proceedings of the Institute of Civil Engineers (ICE)*, volume 2, numéro 1, pp. 325–378. Cité page 22.
- [Wonham, 1979] W. M. Wonham (1979). *Linear multivariable control : a geometric approach*. Springer-Verlag. Cité page 52.
- [Wu *et al.*, 2007a] J. Wu, A. Abbas-Turki, A. Corréia et A. El Moudni (novembre 2007). Controlling a discrete model of two cascading intersections, Dans *Workshop international : Logistique & Transport (LT)*, Hammamet, Tunisie. Cité page 134.
- [Wu *et al.*, 2007b] — (août 2007). Discrete intersection signal control, Dans *IEEE international conference on Service Operations and Logistics, and Informatics (SOLI)*, Philadelphia, USA. Cité page 130.
- [Zad *et al.*, 1999] S. H. Zad, R. H. Kwong et W. M. Wonham (1999). Supremum operators and computation of supremal elements in system theory, *SIAM Journal on Control and Optimization (SICON)*, volume 37, numéro 3, pp. 695–709. Cité page 69.

Annexes

Algorithme de [Lawler, 1977]

Objectif : minimiser le retard maximum $J = \max_{i \in \mathcal{T}}(g_i - d_i, 0)$, avec \mathcal{T} un ensemble de véhicules, g_i la date planifiée et d_i la date « due » du début de la traversée du véhicule i à travers l'intersection. Dans notre cas, les dates « dues » sont artificielles, elles correspondent aux dates de passage au plus tôt des véhicules à travers le carrefour (*i.e.* $\hat{V} = F^*(LX \oplus QU)$ sans considérer la contrainte relative à l'intersection). Nous considérons toutefois la contrainte d'antériorité, à savoir qu'un véhicule ne peut pas prétendre traverser le carrefour avant le véhicule qui le précède. L'ordre des véhicules est reflété par l'ordre des exposants de γ dans les séries de \hat{V} .

Algorithme 4.1 (Algorithme de [Lawler, 1977]) Soit N l'ensemble des véhicules non planifiés. Initialiser N de telle sorte qu'il contienne tous les véhicules. Tant que N est non vide, faire :

1. Calculer $\tau = \sum_{i \in N} p_i$, avec p_i le temps de traversée du véhicule i .
2. Construire l'ensemble S des véhicules qui n'ont pas de précédent dans N . Concrètement, dans notre cas, S contient le véhicule de tête de chaque file d'attente.
3. Calculer $g_j = \min_{i \in S} \tau - p_i$. j est le véhicule planifié à cette itération. Ajuster N : $N = N - \{j\}$.

À la fin de l'algorithme, nous retrouvons facilement les séries décrivant les tirs des transitions de commande v_r pour chaque rue r :

$$v_r = \bigoplus_{j=0}^n \gamma^j \delta^{g_j} \oplus \gamma^{n+1} \delta^* ,$$

avec $j \in [0, n]$ les véhicules de la rue r à planifier.

Code source Scilab de l'algorithme inspiré de [Leung et al., 2001]

```

1  //////////////- INITIALIZE -//////////////////////

N=2*maxcycles; // this constant must be arbitrarily greater than the total makespan
maxdensity=0.15; // common value in litterature

5  nbres=1; // overwrite number of available resources since this algorithm can only handle one resource
[nblanes,lastinputdate]=size(U); // extract useful information
queuedelay=[];
queuecapacity=[];
10  Vnoconflict=list();
release=list();
deadline=list();
X=[];
V=[];
15  Y=[];
t=1;

for lane=1:nblanes
// compute delay on place of queue
20  queuedelay(lane)=int(queuelength(lane)/queuespeed(lane));

// compute queuecapacity
queuecapacity(lane)=int(queuelength(lane)*maxdensity);

25  // convert counters into daters
Udater=[];
k=1;
for t=1:lastinputdate
    while k<=U(lane,t)
        Udater(k)=t;
        k=k+1;
    end
end

30  // compute ideal (unreachable) firing dates for transitions V

```

```

Vnoconflict(lane)=[];
Xdater=[];
Ydater=[];
for k=1:length(Udater)
    Xdater(k)=max(FILLV(Xdater,k-1)+2,Udater(k),FILLV(Vnoconflict(lane),k-queuecapacity(lane)));
    Vnoconflict(lane)(k)=max(FILLV(Vnoconflict(lane),k-1)+2,
        Xdater(k-Minit(lane))+queuedelay(lane),FILLV(Ydater,k-1));
    Ydater(k)=Vnoconflict(lane)(k)+critdelay(lane);
end
40
// compute "modified release dates"
release(lane)=[Vnoconflict(lane)(1)];
for k=2:Minit(lane)+U(lane,lastinputdate)
    release(lane)(k)=max(Vnoconflict(lane)(k),release(lane)(k-1)+2+critdelay(lane));
50
end
// compute "modified (pseudo-)deadlines"
deadline(lane)=[];
for k=1:Minit(lane)+U(lane,lastinputdate)
    deadline(lane)(k)=N-(Minit(lane)+U(lane,lastinputdate)-k)*(2+critdelay(lane));
55
end
// set some variables
cursor(lane)=1; // in order to retrieve current token's modified release date and deadline for each lane
maxlocalwaitingtime(lane)=0; // we do not consider initial tokens for statistics on waiting times
lastoutdate(lane)=1; // saved only not to explore all time scale during max local waiting time update
end
60
////////// - RUN ALGORITHM -//////////
continue=%T;
resourceusage=[];
t=1; // time origin is t=1 for technical reasons (lists and matrices have first index value equal to 1)
while continue
    // build list of candidates
    eligibles=[];
    for lane=1:nblanes
70

```

```

75   if cursor(lane)<=length(release(lane))
      if release(lane)(cursor(lane))<=t
          eligibles=[eligibles;lane];
      end
    end
  end
80   if length(eligibles)>0 // if at least one token enables a V-transition
      otherVs=e;
      Ys=e;
      chosen=eligibles(1);
      for currentel=2:length(eligibles)
          if deadline(eligibles(currentel))(cursor(eligibles(currentel)))
              < deadline(chosen)(cursor(chosen))
                  chosen=eligibles(currentel);
          end
      end
90   end
  end
  for lane=1:nblanes
      X(lane,t)=min(FILLM(X,lane,t-2)+1,
100     FILLM(V,lane,t)+queuecapacity(lane)-Minit(lane),FILLM(U,lane,t));
      Y(lane,t)=FILLM(V,lane,t-critdelay(lane)); // OK to place it here since every critdelay>=1
      if length(eligibles)==0
          V(lane,t)=FILLM(V,lane,t-1); // control remains unchanged
          X(lane,t)=min(FILLM(X,lane,t-2)+1,
105     FILLM(V,lane,t)+queuecapacity(lane)-Minit(lane),FILLM(U,lane,t));
      else // if at least one token enables a V-transition
          Ys=otimes(Ys,Y(lane,t)); // this is valid as we have every critdelay>=1
          if lane<>chosen
              V(lane,t)=FILLM(V,lane,t-1); // this is a valid "shortcut" for computation if
              // we allow all available resources to one user
              // at each step or if there is only one resource
              X(lane,t)=min(FILLM(X,lane,t-2)+1,
                  FILLM(V,lane,t)+queuecapacity(lane)-Minit(lane),FILLM(U,lane,t));
              otherVs=otimes(otherVs,V(lane,t));
          end
      end
  end

```



```

110         end
        end
        if length(eligibles)>0 // if at least one token enables a V-transition
            lastV=FILLM(V,chosen,t-1);
            V(chosen,t)=min(FILLM(V,chosen,t-2)+1,
                Minit(chosen)+FILLM(X,chosen,t-queuedelay(chosen)),Ys+nbres-otherVs);
            X(chosen,t)=min(FILLM(X,chosen,t-2)+1,
                V(chosen,t)+queucapacity(chosen)-Minit(chosen),FILLM(U,chosen,t));
120         if V(chosen,t)>lastV // if a token begins to use resource at date t
            cursor(chosen)=cursor(chosen)+1; // once a token has left queue, "point" on next one
            // update maximum local waiting time
            while Minit(chosen)+FILLM(U,chosen,lastoutdate(chosen))<V(chosen,t)
                lastoutdate(chosen)=lastoutdate(chosen)+1;
125         end
            // waiting time including time to enter the queue in case of saturation
            maxlocalwaitingtime(chosen)=
                max(maxlocalwaitingtime(chosen),t-lastoutdate(chosen));
130         for i=t:t+critdelay(chosen), resourceusage(i)=1;end
        end
        // compute wether simulation has ended (ie every token has left the system)
        continue=%F;
        for lane=1:nblanes
135             if Y(lane,t)<Minit(lane)+U(lane,lastinputdate), continue=%T;break;end
        end
        i=length(resourceusage)+1;
140         while length(resourceusage)<t, resourceusage(i)=0;i=i+1;end
        t=t+1;
        if t>maxcycles, continue=%F;end // watchdog
145     end

```

```
// extend U over time scale (in order to plot it with same scale as V and Y)
while lastinputdate+1<t
  for lane=1:nblanes, U(lane,lastinputdate+1)=U(lane,lastinputdate);end
  lastinputdate=lastinputdate+1;
end
150
```


Résumé

Le trafic routier entraîne de nombreux effets néfastes, dont la pollution, l'insécurité et la congestion. Dans la gestion de l'espace urbain alloué aux véhicules, les carrefours à feux sont des lieux stratégiques. Bien que de nombreux travaux de recherche proposent des modélisations et des commandes des feux, les outils les plus éprouvés actuellement arrivent à leurs limites. Dans ce mémoire, nous proposons une nouvelle vision du trafic. Nous considérons les carrefours à feux comme des systèmes discrets dans l'espace et le temps. Plus précisément, nous proposons un modèle analytique du trafic urbain dans l'algèbre des dioïdes. Afin de valider ce modèle, nous vérifions, à travers son application à des segments de rue, qu'il permet de représenter les deux états du trafic, à savoir fluide et congestionné. En outre, il fournit les différentes grandeurs macroscopiques et microscopiques du trafic.

La généralisation du modèle aux carrefours pose le problème de la modélisation de conflits par l'algèbre des dioïdes. Or, les travaux proposés dans la littérature ne nous permettent pas de nous affranchir de cette difficulté d'une manière adaptée au problème étudié. Ainsi, nous définissons deux modélisations analytiques. Nous prouvons que ces modèles décrivent formellement tout système constitué d'un nombre invariant de ressources à partager entre des utilisateurs. Ainsi, ce nouveau formalisme nous conduit vers une assimilation du carrefour à un problème de ressources partagées. En adaptant des algorithmes d'ordonnancement aux modèles, nous proposons plusieurs commandes en fonction des hypothèses retenues.

Afin de valider notre approche, nous l'appliquons au problème de régulation du trafic. Les simulations sur un carrefour élémentaire montrent que la prise en considération des véhicules d'une manière individuelle permet d'améliorer sensiblement la fluidité du trafic. Ceci nous a encouragé à généraliser l'approche pour deux carrefours successifs pour la comparer aux ondes vertes. Bien que notre commande optimise localement la circulation, elle donne des résultats comparables.

Mots-clés : carrefour à feux, régulation de trafic, Systèmes dynamiques à Événements Discrets, algèbre des dioïdes, ordonnancement, recherche opérationnelle.

Abstract

Road traffic leads to many issues including pollution, insecurity and congestion. In the management of the urban space allocated to vehicles, intersections with traffic lights are strategic locations. Although many studies suggest models and controls of traffic lights, the most popular tools currently reach their limits. In this thesis, we propose a new vision of traffic. We consider junctions with traffic signals as discrete systems in space and time. Specifically, we propose an analytical model of urban traffic in dioid algebra. In order to validate the model, we check, through its application to segments of a street, that it is capable of representing the two states of traffic, namely fluid and congested. In addition, it provides various macroscopic and microscopic traffic indicators.

The generalization of the model to crossroads involves the problem of modeling conflicts in dioid algebra. Much work proposed in the literature do not allow us to avoid this issue adequately to the studied problem. Thus, we define two analytical models. We prove that these formal models describe any system that consists of an invariant number of resources to be shared between users. Thus, this new formalism leads us consider crossroads as problems of shared resources. By adapting scheduling algorithms to models, we propose several controls depending on considered assumptions.

In order to validate our approach, we apply it to the problem of traffic management. Simulations on an elementary crossroad show that considering vehicles as individual entities significantly improves the traffic fluidity. This encouraged us to generalize the approach to two successive intersections in order to compare it to green waves. Although our control optimizes traffic locally, it gives comparable results.

Keywords: traffic lights, traffic control, Discrete Event Dynamic Systems, dioid algebra, scheduling, operations research.