



HAL
open science

Régulation adaptative multi-objectif et multi-mode aux carrefours à feux

Yann Dujardin

► **To cite this version:**

Yann Dujardin. Régulation adaptative multi-objectif et multi-mode aux carrefours à feux. Autre [cs.OH]. Université Paris Dauphine - Paris IX, 2013. Français. NNT : 2013PA090010 . tel-00904781

HAL Id: tel-00904781

<https://theses.hal.science/tel-00904781>

Submitted on 15 Nov 2013

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

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

UNIVERSITE PARIS-DAUPHINE
ECOLE DOCTORALE DE DAUPHINE

Laboratoire : LAMSADE

THESE

Pour l'obtention du titre de :
DOCTEUR EN INFORMATIQUE
(arrêté du 7 août 2006)

Spécialité : Aide à la décision

Présentée et soutenue publiquement par :

Yann DUJARDIN

le 24 juin 2013

Régulation adaptative multi-objectif et multi-mode aux carrefours à feux

JURY

Directeur de Thèse : Daniel VANDERPOOTEN
Professeur à l'Université Paris-Dauphine

Rapporteurs : Christian ARTIGUES
Directeur de Recherche CNRS au LAAS

Patrice PERNY
Professeur à l'Université Pierre et Marie Curie

Suffragants : Florence BOILLOT
Chargée de Recherche à l'IFSTTAR

Kathrin KLAMROTH
Professeur à la Bergische Universität Wuppertal (Allemagne)

Remerciements

Je remercie Daniel Vanderpooten, mon directeur de thèse, grâce à qui j'ai pu appréhender le métier de chercheur sous de nombreux aspects, et qui a su me montrer les difficultés, parfois insoupçonnées, inhérentes à ce dernier. De plus, son esprit critique acéré m'a permis d'élever sensiblement, à mon sens, la qualité de mes travaux. Enfin, je dois rendre hommage au fait que, sur quatre ans et demi de rencontres régulières, pas une seule réunion ne prit fin sans que tous les points à traiter n'eussent pas tous été abordés (ce qui peut prendre un temps certain).

Je remercie également Florence Boillot, mon encadrante IFSTTAR (ex INRETS), pour s'être investie aussi durablement et efficacement dans le suivi de ma thèse, notamment sur la partie transport. Grâce à ses remarques régulières et pertinentes, j'ai pu progressivement me rendre compte de l'ampleur des contraintes liées à la tâche ambitieuse que représente l'application d'une théorie à un cas pratique.

Je tiens à remercier Christian Artigues et Patrice Perny pour avoir accepté d'être rapporteurs de cette thèse et pour l'intérêt qu'ils portent à ce travail.

Je remercie Kathrin Klamroth pour avoir bien voulu faire partie de mon jury de thèse en tant qu'examineur.

Je remercie les membres (ou anciens membres) de l'équipe "Carrefours intelligents" du GRETTIA :

Jean-Claude Pierrelée, pour tout ce qu'il a pu faire pour moi pendant mes années de thèse, toujours avec efficacité et professionnalisme. Son recul et sa longue expérience dans le monde de la recherche m'ont également beaucoup apporté.

Mohamed Rahal, avec qui, durant toute la seconde moitié de ma thèse, j'ai pu échanger sur des sujets aussi passionnants que variés, outre les sujets d'ordre purement professionnel, comme les mathématiques, la bio-informatique, l'informatique musicale, et même l'astrophysique ! Cela m'a été très profitable de pouvoir côtoyer quelqu'un au

quotidien qui possède une réelle curiosité intellectuelle, un grand enthousiasme pour tout ce qui touche, de près ou de loin, à la recherche scientifique, et globalement un dynamisme dans tout ce qu’il entreprend. J’ai pu, de plus, grâce à lui, découvrir un peu de son domaine de prédilection (l’analyse de données), et j’espère lui avoir fait découvrir quelques facettes du mien en échange.

Pierre Vinant, qui s’est intéressé de près à divers aspects techniques de mes travaux (partie transport) pendant une bonne première moitié de thèse, et dont le souci du détail m’a aidé à mieux appréhender les différents mécanismes et les différentes pratiques en jeu dans la régulation du trafic routier.

Je remercie tous les doctorants du LAMSADE que j’ai pu connaître et avec qui j’ai pu échanger. Je souhaiterais remercier en particulier mon “frère de thèse”, Renaud Lacour. En ce qui le concerne, je ne pourrai être exhaustif étant donné le grand nombre d’éléments qui ont, d’une manière ou d’une autre, joués un rôle positif dans ma thèse, et dont je ne pourrai qu’effleurer l’étendue en disant qu’évidemment, de nombreuses discussions ayant pour objet l’informatique en général, et portant sur l’optimisation multi-objectif en particulier, ont jalonné nos années de thèse, et m’ont sans nul doute permis d’accroître mes connaissances et d’élargir ma vision du domaine. De plus, je dois ajouter qu’il est très fort pour dénicher les derniers outils à la pointe pour faciliter la vie du thésard (et du chercheur en général), et qu’il m’a fait profiter de cette qualité à de nombreuses reprises durant la thèse. Je remercie également les doctorants qui ont bien voulu adhérer à l’idée d’une “sortie brasserie” de temps à autre, favorisant ainsi les échanges dans un cadre non académique : Amine, Édouard, Florian, Lies, Lydia et Romain.

Je remercie, en général, tous les membres du LAMSADE et du GRETTIA que j’ai pu rencontrer et en particulier ceux avec qui j’ai pu entretenir des conversations, que ce soit sur des sujets d’ordre professionnel ou non.

Table des matières

Principales notations	8
Introduction	13
1 Concepts et problématique	17
1.1 La régulation du trafic aux carrefours à feux	17
1.1.1 Notions fondamentales concernant le trafic aux carrefours à feux	17
1.1.2 Plans de feux, fonctionnement cyclique	21
1.1.3 Modélisations microscopique et macroscopique, mesure des gran- deurs physiques associées à l'utilisation de capteurs	22
1.1.4 Caractéristiques essentielles d'un système de régulation	24
1.2 Concepts d'optimisation multi-objectifs	28
1.2.1 Formulation	28
1.2.2 Dominance, efficacité	29
1.2.3 Point idéal, point nadir	30
1.2.4 Paramètres préférentiels, fonction d'agrégation	31
1.2.5 Procédures interactives	35
1.2.6 Procédures interactives temps réel	36
1.3 Problématique de la thèse	38
1.3.1 Les enjeux	39
1.3.2 Les critères	39
1.4 Conclusion du chapitre	40
2 État de l'art des systèmes et des méthodes d'optimisation utilisées dans la régulation du trafic	41

2.1	Systèmes non-adaptatifs	41
2.1.1	Systèmes mono-objectifs	42
2.1.2	Systèmes multi-objectifs	45
2.2	Systèmes semi-adaptatifs	48
2.2.1	Approches mono-objectifs	48
2.2.2	Approches multi-objectifs	51
2.3	Systèmes adaptatifs	62
2.3.1	Approches mono-objectifs	62
2.3.2	Approches multi-objectifs	65
2.4	Conclusion du chapitre	65
3	Modélisation du trafic, présentation du système et des modèles, évolution des paramètres préférentiels	67
3.1	Modélisation du trafic sur un carrefour	68
3.1.1	Hypothèses sur le carrefour et sur les données disponibles en temps réel	68
3.1.2	Description du carrefour	69
3.1.3	La modélisation des usagers	69
3.1.4	Modélisation des files d'attente (VP)	70
3.1.5	Modélisation des bus	73
3.2	Présentation générale du système	76
3.2.1	Description générale du système	77
3.2.2	Évolution de l'instance	78
3.2.3	Formulation générale de la fonction objectif commune aux trois modèles	79
3.2.4	Notations utilisées pour les trois modèles	80
3.3	Modèle "par états"	81
3.3.1	Présentation	81
3.3.2	Variables	82
3.3.3	Contraintes	82
3.3.4	Critères	85
3.4	Modèle "par dates"	88
3.4.1	Présentation	88

3.4.2	Constantes	90
3.4.3	Variables	90
3.4.4	Contraintes	91
3.4.5	Critères	95
3.5	Comparaison des modèles “par états” et “par dates”	102
3.5.1	Hypothèses de travail, représentation de l’environnement	102
3.5.2	Estimation du nombre de variables et de contraintes pour le carrefour C	103
3.5.3	Formulation des critères	103
3.5.4	Extension sur une vision régionale (quelques carrefours)	104
3.6	Modèle hybride	106
3.6.1	Modification des variables par dates	106
3.6.2	Liaisons entre les dates de commutation des feux et les états des feux.	107
3.6.3	Choix de formulations	108
3.6.4	Comparaison en terme de nombre de variables et de contraintes, avec les autres modèles	109
3.7	Formulation de deux types de méthodes interactives pour la régulation de trafic	111
3.7.1	Cas où le décideur n’a pas d’expertise particulière sur les valeurs des critères, mais sur d’autres indicateurs	111
3.7.2	Cas où le décideur dispose de connaissances sur les valeurs des critères	121
3.8	Conclusion du chapitre	132
4	Comparaison des modèles sur une itération et résultats sur l’utilisation du modèle hybride en temps réel	135
4.1	Comparaison des modèles sur une seule itération	137
4.1.1	Méthodologie	137
4.1.2	Résultats	139
4.2	Optimisation sur plusieurs itérations	145
4.2.1	Fréquence des optimisations	145
4.2.2	Méthodologie	148

4.2.3	Résultats	158
4.2.4	Conclusions et remarques sur les résultats obtenus	186
4.3	Conclusion du chapitre	187
	Conclusion générale	189
	Bibliographie	192
	Annexes	200
	Annexe A : Contraintes additionnelles pour les bus (modèle par dates) . . .	201
	Annexe B : Respect des exigences du mécanisme de régulation pour la procé- dure interactive sans expertise du décideur dans le cas général	202
	Annexe C : Avantages du changement de phasage dans un cas simple	209

Principales notations

Notations relatives aux modélisations/modèles

Ensembles

- F ensemble des feux du carrefour considéré, désigne aussi l'ensemble des tronçons sur le carrefour, on note $N_F = |F|$
- $S = \{1, \dots, H\}$ ensemble discret des instants futurs considérés jusqu'à l'horizon de temps H
- $S' = \{1, \dots, H - 1\}$
- $C = \{1, \dots, N_C\}$ ensemble des commutations possibles, où N_C représente le nombre de commutations (modèle par dates)
- $C' = \{1, \dots, N_C - 1\}$ (modèle par dates)
- $C_f = \{1, \dots, N_{C_f}\}$ ensemble des commutations possibles sur le feu $f \in F$, où N_{C_f} représente le nombre de commutations sur le feu $f \in F$ (modèle hybride)
- $C'_f = \{1, \dots, N_{C_f} - 1\}$ (modèle hybride)
- B ensemble des bus pris en compte au niveau de l'itération courante

Variables

- $e_{s,f}$ état du feu $f \in F$ à l'instant $s \in S$ (vaut 1 si le feu est vert et 0 s'il est rouge)
- $l_{s,f}$ longueur de la file sur le tronçon $f \in F$ à l'instant $s \in S$
- t_c date de la $c^{\text{ième}}$ commutation, $c \in C$ (modèle par dates)
- $x_{c,f}$ état du feu $f \in F$ entre la $c^{\text{ième}}$ et la $(c + 1)^{\text{ième}}$ commutation, $c \in C'$ (modèle par dates)
- $t_{c,f}$ date de la $c^{\text{ième}}$ commutation du feu $f \in F$ (modèle hybride)

- $l_{c,f}$ longueur de la file sur le tronçon $f \in F$ entre la $c^{i\grave{e}me}$ et la $(c+1)^{i\grave{e}me}$ commutation

Critères

- TA temps total d'attente
- NA nombre d'arrêts
- ECB erreur totale de consigne pour les bus
- VP véhicules particuliers
- TC transports en commun

Constantes

- u unité de temps du système
- H horizon de temps sur lequel on travaille, en unités de temps
- $QA_{s,f}$ débit des arrivés des véhicules particuliers sur le tronçon $f \in F$, en véhicules par unité de temps, à l'instant $s \in S$
- QA_f débit sur le tronçon $f \in F$ à l'instant présent
- QS_f débit de saturation, en véhicules par seconde, du feu $f \in F$
- t_0 date de la dernière commutation ayant eu lieu avant l'instant présent
- $t_{0,f}$ date de la dernière commutation ayant eu lieu sur le feu $f \in F$ avant l'instant présent
- $l_{0,f}$ longueur de la file, en nombre de véhicules, sur le tronçon $f \in F$: à l'instant présent pour le modèle par état, entre t_0 et l'instant présent pour le modèle par dates, et entre $t_{0,f}$ et l'instant présent pour le modèle hybride
- D_V^{min} , D_V^{max} , D_R^{min} et D_R^{max} les durées respectivement minimum et maximum des états respectivement verts et rouges, des feux
- T_b date d'arrivée du bus $b \in B$
- TAB_b , TAB_b^0 et EC_b respectivement le temps que le bus $b \in B$ passe sur le carrefour, sa consigne de temps d'attente, et son erreur de consigne
- d_{it} temps alloué à une itération

Notations relatives à l'optimisation multi-objectif

Ensembles et critères

- X ensemble des actions potentielles
- p nombre de critères considérés
- $f_j, j \in \{1, \dots, p\}$ critère d'indice j (on dira aussi critère $j \in \{1, \dots, p\}$)
- $Z \subseteq \mathbb{R}^p$ espace des critères
- $Z_X = \{z \in Z : \forall j \in \{1, \dots, p\}, z_j = f_j(x), x \in X\}$ l'image de X dans Z

Points et fonctions remarquables

- z point de Z (si de plus $z \in Z_X$ alors il existe $x \in X$ tel que $z_j = f_j(x), j = 1, \dots, p$, la notation $f_j(x)$ n'étant utilisée qu'occasionnellement, à laquelle on préfère z_j)
- \bar{z} point de référence
- z^* point idéal
- z_* point anti-idéal
- \hat{z} point de l'espace des critères correspondant à la solution courante
- $g_{\bar{z}, \lambda}(z) = \max_{j=1..p} \{\lambda_j (z_j - \bar{z}_j)\}, z \in Z_X, \lambda \in (\mathbb{R}_+^*)^p$: fonction à base de point de référence classique
- $f_{\bar{z}, \lambda}(z) = \max_{j=1..p} \{\lambda_j (z_j - \bar{z}_j)\} + \rho \sum_{j=1}^p \lambda_j (z_j - \bar{z}_j), z \in Z_X, \rho > 0$: fonction à base de point de référence, forme augmentée de $g_{\bar{z}, \lambda}(z)$
- $h_{\bar{z}, \lambda}$: vaut $g_{\bar{z}, \lambda}$ ou $f_{\bar{z}, \lambda}$

Notations relatives aux méthodes interactives

- $\tau_A = \tau$ taux des arrivées courant
- $z_{profil} = z_{profil}(\tau)$ point de l'espace des critères traduisant la situation du trafic
- m point de l'espace des critères dont la coordonnée j minore $f_j, j = 1..p$
- M point de l'espace des critères dont la coordonnée j majore $f_j, j = 1..p$
- \mathbf{u} est tel que $u_j = \frac{1}{\lambda_j}, j = 1, \dots, p$ (\mathbf{u} est un vecteur directeur de la droite de projection de \bar{z} sur $ND(Z_X)$)
- \tilde{z} point de Z tel que $\mathbf{u}\mathbb{R} = (\bar{z}, \tilde{z})\mathbb{R}$ (droite de projection)
- q nombre de seuils de préférence

- $s_{i,j}$ seuil de préférence i relatif au critère j , $i = 1, \dots, q$, $j = 1, \dots, p$
- Z_{i_1, i_2, \dots, i_p} zone d'indices $i_1, \dots, i_p \in \{1, \dots, q\}$

Introduction

Cette thèse s’inscrit dans le thème de recherche IFSTTAR “Carrefour Intelligent”, qui vise la gestion des carrefours à feux dans plusieurs domaines, en particulier celui de la régulation.

On peut voir le problème de la régulation du trafic aux carrefours à feux comme un problème de gestion des accès à un espace commun, le carrefour, par différents types d’usagers. Les différents usagers ne peuvent y pénétrer simultanément pour des raisons de sécurité. Les règles qui régissent ces accès sont, en général, précisément fixées dans les pays développés. Les normes correspondantes sont alors déterminées par les autorités de ce pays. Nous nous placerons dans ce cadre.

Depuis un certain nombre d’années, grâce aux avancées technologiques dans le domaine des réseaux de transports terrestres, il est possible de disposer de mesures fiables et précises des grandeurs physiques relatives à l’état du trafic routier en *temps réel*. La littérature s’est, par conséquent, progressivement enrichie de méthodes et de systèmes permettant de réguler le trafic en temps réel, à partir de ces mesures. Bien sûr, la nature des mesures et la façon de les utiliser pour réguler le trafic peuvent grandement différer d’un système à l’autre, et nous essaierons d’en rendre compte dans cette thèse (chapitre 2). Cependant, la caractéristique temps réel est bien un trait commun et intéressant des méthodes et systèmes actuels. Elle fera partie intégrante de la problématique de la thèse.

L’objectif même de la régulation du trafic à un carrefour à feux pouvant différer suivant le point de vue qu’on adopte, il convient de définir comment nous l’abordons. Nous avons considéré trois enjeux naturels que sont la *fluidité* du trafic, le *confort* des usagers et l’*environnement*. De plus, nous avons pris en compte le fait qu’il existe

plusieurs types d’usagers, comme les *véhicules particuliers* (VP) et les *transports en commun* (TC).

La multiplicité des enjeux et des types d’usagers amènent à envisager plusieurs critères à considérer pour la régulation. La plupart des approches de la littérature ne s’appuient que sur un seul critère. Il s’agit, le plus souvent, d’un critère relatif au temps d’attente des véhicules particuliers. En ce qui nous concerne, on considérera trois critères : le *temps d’attente* (TA) et le *nombre d’arrêts* (NA) pour les véhicules particuliers, et un critère spécifique pour les *transports en commun*. Chaque critère ne correspond pas exactement à un enjeu particulier, mais peut couvrir un ensemble d’enjeux. Ces critères ainsi que leurs liens avec les enjeux seront détaillés au chapitre 1. Le problème qui consiste à optimiser “simultanément” ces critères est un *problème d’optimisation multi-objectif*. Dans notre cas, on va chercher à *minimiser* ces critères.

L’optimisation multi-objectif s’intéresse à la résolution de tels problèmes. Contrairement au cas de l’optimisation mono-objectif, où il existe une seule valeur optimale du seul critère en jeu, dans le cas de problèmes multi-objectifs, il peut exister un ensemble plus ou moins important de “meilleurs compromis” entre les critères, représentant tous un optimum pour le problème.

Deux grands axes classiques dans l’étude de tels problèmes sont :

1. La recherche d’une “bonne” représentation de l’ensemble des meilleurs compromis
2. La recherche d’un meilleur compromis au sens de préférences spécifiques

Dans la très grande majorité des approches de la littérature où plusieurs critères sont considérés, les critères sont agrégés *a priori*, transformant le problème, au départ multi-objectif, en un problème mono-objectif résoluble par les méthodes d’optimisation mono-objectif classiques. Dans la thèse, au contraire, nous avons traité le problème réellement comme un problème multi-objectif, en s’attachant à le résoudre selon le deuxième axe indiqué ci-dessus.

Plusieurs modèles ont été développés durant la thèse, chacun visant à résoudre ce problème. Ces modèles sont tous des programmes linéaires mixtes.

Le document est organisé comme suit.

Le chapitre 1 introduit les concepts fondamentaux, dans le domaine du trafic et dans le domaine de l'optimisation, nécessaires à la bonne compréhension du rapport. Le chapitre introduit également la problématique.

Le chapitre 2 est consacré à l'état de l'art des méthodes d'optimisation utilisées dans les systèmes de régulation en milieu urbain.

Le chapitre 3 concerne à la fois la modélisation générale du trafic avec les différents types d'acteurs, la description du système élaboré pour la thèse avec les formulations détaillées des modèles et la politique évolutive de régulation.

Enfin, le chapitre 4 regroupe les résultats obtenus en ce qui concerne la comparaison des modèles élaborés et l'utilisation du système en temps réel.

Enfin, la dernière partie contient la conclusion et les perspectives.

Chapitre 1

Concepts et problématique

Ce chapitre est composé de trois sections. La section 1.1 présente les concepts fondamentaux liés au domaine de la régulation du trafic. La section 1.2 présente les concepts fondamentaux liés au domaine de l'optimisation multi-objectif. Enfin, la section 1.3 présente la problématique de la thèse.

1.1 La régulation du trafic aux carrefours à feux

Nous présentons ici à la fois les notions fondamentales du trafic aux carrefours à feux, et ce que l'”optimisation” de la gestion des feux d'un carrefour peut recouvrir comme sens dans ce contexte. Nos choix quant à la définition de cette optimisation, ce que cela implique pour la modélisation du problème et de sa mise en œuvre, seront également abordés.

1.1.1 Notions fondamentales concernant le trafic aux carrefours à feux

1.1.1.1 Aspects géométriques

Selon la terminologie française en terme de trafic routier (cf. (AFNOR, 1991)), concernant en particulier les carrefours à feux, une *voie de circulation* désigne une “subdivision transversale de la chaussée ayant une largeur suffisante pour permettre la circulation d'une file de véhicules”. Les voies de circulation sont, de plus, orientées.

Un *couloir*, toujours selon la même source, est un ensemble de voies d'une chaussée, parallèles, de même sens et contiguës.

Un carrefour est un espace où se rencontrent plusieurs couloirs d'entrée ou de sortie de véhicules, pouvant contenir chacun plusieurs voies. Chaque couloir entrant possède au moins un feu. Les véhicules marquent l'arrêt à une *ligne de feu*.

On appelle C le carrefour présenté en figure 1.1, où les voies entrantes sont numérotées de 1 à 5 dans le sens trigonométrique, et possèdent toutes leur propre feu. Il y a donc 5 feux. Les flèches représentent les directions possibles des véhicules après le passage du feu. Les segments rouges représentent les lignes de feu.

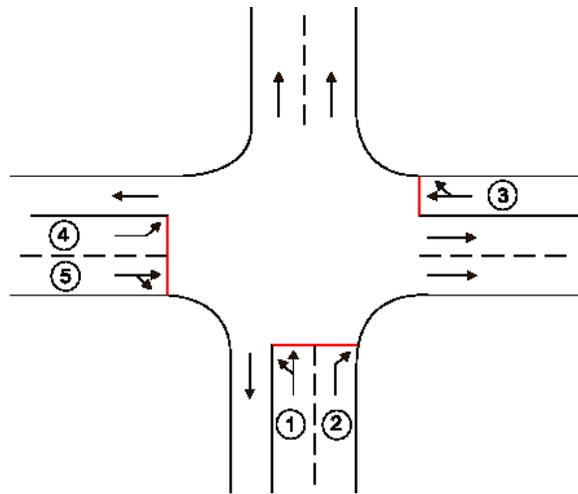


FIGURE 1.1 – Carrefour C

Pour chaque ensemble de voies entrantes *relatives au même feu*, on appelle *tronçon* une section longitudinale de cet ensemble. Un tronçon possède une entrée et une sortie. L'entrée correspond à la délimitation géométrique du tronçon en amont, et la sortie sa délimitation en aval. S'il n'y a pas d'indication particulière, et notamment sur les réseaux de carrefours, l'entrée d'un tronçon est en général située au niveau du carrefour ou de l'intersection amont. Cependant, dans le cas où certaines hypothèses supplémentaires sont à prendre en compte, il peut être judicieux de considérer l'entrée d'un tronçon de manière différente. Ce sera en particulier notre cas, dans le cadre de la thèse (voir 3.1). On appelle longueur du tronçon la distance séparant l'entrée de la sortie.

L'éventuelle zone piétonne ne fait pas partie du tronçon dont la sortie est confondue avec la ligne de feu (figure 1.2).

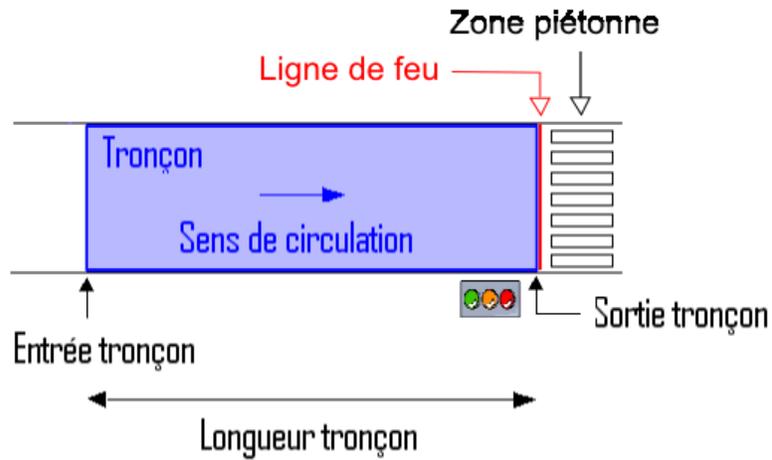


FIGURE 1.2 – Tronçon et éléments périphériques

Sur la figure 1.3, les tronçons du carrefour C sont représentés par des rectangles bleus. Sur cet exemple, chaque tronçon correspond à une unique voie entrante. Ceci est dû au fait que chacune d'elle possède un feu. Il s'agit de feux directionnels.

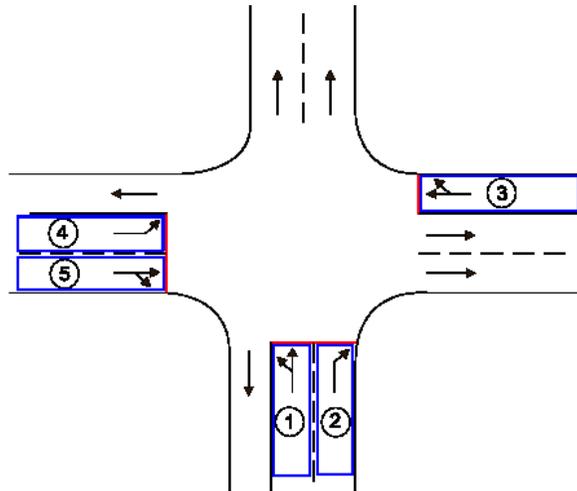


FIGURE 1.3 – Tronçons du carrefour C

1.1.1.2 Antagonismes

On appelle “mouvement directionnel” une trajectoire de véhicules traversant le carrefour, ayant pour origine un couloir d’entrée bien déterminé et pour destination un

même couloir de sortie.

Sur la figure 1.4, on peut voir de tels mouvements. Par exemple les véhicules provenant du tronçon 2 ou du tronçon 4 n'ont qu'une seule possibilité de mouvement. Les deux mouvements correspondants sont représentés en tiretés. En revanche, les véhicules provenant du tronçon 1, par exemple, ont le choix entre deux mouvements : aller tout droit ou tourner à gauche.

Pour des raisons de sécurité, certaines paires de mouvements sont interdites. Bien que des règles générales existent concernant la détermination de ces interdictions, leur élaboration est à la discrétion de l'exploitant du carrefour.

En ce qui concerne la régulation du trafic, l'information utile est l'ensemble des incompatibilités entre les différents feux du carrefour, c'est-à-dire l'ensemble des paires de feux qui ne peuvent laisser passer simultanément des véhicules, que l'on peut déduire des interdictions portant sur les paires de mouvements directionnels. Par exemple, sur la figure 1.4, les feux relatifs aux tronçons 1 et 3 seront automatiquement considérés comme antagonistes à cause des mouvements directs, qui se croisent perpendiculairement, ce qui est absolument proscrit.

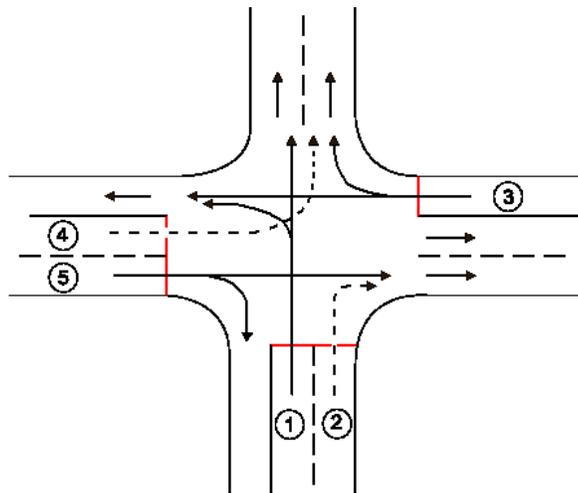


FIGURE 1.4 – Mouvements possibles en provenance de chaque feu

Mais l'exploitant peut aussi ajouter des antagonismes s'il les juge primordiaux pour la sécurité. Par exemple le feu 2 et le feu 5 peuvent être déclarés antagonistes à cause du mouvement direct de 5, s'il est avéré que le débit sur le tronçon 5 est élevé. Les critères pris en compte sont par exemple le manque de visibilité ou encore les vitesses

sur les tronçons.

On peut représenter par un graphe non orienté, l'ensemble de ces antagonismes.

Chaque sommet du graphe représente un feu. Pour chaque paire de sommets, il existe une arête entre les deux sommets si et seulement si les deux feux sont considérés comme antagonistes.

Le graphe correspondant au carrefour C (figure 1.1) est le graphe G suivant (figure 1.5) :

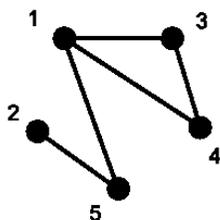


FIGURE 1.5 – Graphe G des antagonismes pour le carrefour C

La matrice d'adjacence de ce graphe des antagonismes est appelée *matrice des antagonismes*.

1.1.2 Plans de feux, fonctionnement cyclique

Pour un feu donné, on appelle *état* la couleur dans laquelle se trouve, à un instant donné le signal commandé en ligne. Pour les feux tricolores, en France, les états possibles sont vert, jaune et rouge (AFNOR, 1991). La majeure partie des carrefours en France, fonctionnent suivant un *plan de feux*. Un plan de feux est un “ensemble de données définissant complètement le déroulement des états de toutes les lignes de commande des signaux d’un carrefour à feux” (AFNOR, 1991). Cet ensemble de données ne peut être déterminé que pour un fonctionnement “cyclique” du carrefour, c’est-à-dire tel qu’il “peut être décrit par une séquence d’états prédéfinis de l’ensemble des feux du carrefour” (AFNOR, 1991). Nous donnons ici la terminologie associée à la conception de plans de feux.

On rappelle qu'un stable, pour un graphe donné, est un ensemble de sommets deux à deux non-adjacents, et qu'un stable maximal est un stable qui n'est contenu strictement dans aucun autre stable. Les stables maximaux pour le graphe G sont les ensembles :

$$\{1, 2\}, \{2, 3\}, \{2, 4\}, \{3, 5\}, \{4, 5\}$$

Le *phasage* d'un carrefour peut être vu comme une séquence $S = (S_1, \dots, S_k)$, $k \leq n$, de stables maximaux de son graphe G associé, où n est le nombre de sommets de G .

Cette séquence est telle que chaque sommet du graphe doit apparaître dans au moins un des ses éléments.

Les éléments de S sont appelés *phases de trafic*, ou plus simplement *phases* (AFNOR, 1991). A chaque phase est associée une durée. Pendant cette durée, seuls les feux dont les sommets correspondants appartiennent au stable pourront être mis au vert.

On appelle *rouge de dégagement* la période de temps minimale qui s'écoule entre le passage au rouge d'un feu, et le passage au vert d'un feu antagoniste à ce dernier (AFNOR, 1991). Par abus de langage on appellera aussi rouge de dégagement la durée correspondante.

Un *plan de feux* d'un carrefour est un phasage muni de la durée de chaque phase qui la compose, et des rouges de dégagement correspondants. Le terme *cycle* peut tout aussi bien désigner la durée du plan de feu ou bien la séquence des phases (AFNOR, 1991). Dans la thèse, on désignera par cycle la *durée* du plan de feux. Pour un système basé sur l'utilisation d'un plan de feux, le cycle du système désignera le cycle du plan de feux utilisé.

1.1.3 Modélisations microscopique et macroscopique, mesure des grandeurs physiques associées à l'utilisation de capteurs

1.1.3.1 Modélisations microscopique et macroscopique

Il existe dans la littérature, deux grands types de modélisation du trafic routier qui sont la modélisation microscopique et la modélisation macroscopique. Pour une introduction approfondie, on se reportera à (Cohen, 1990).

Tandis que la modélisation microscopique s’attache à décrire le trafic en modélisant chaque véhicule individuellement, la modélisation macroscopique consiste à regrouper les véhicules de mêmes caractéristiques en une seule entité, qui, sous certaines conditions, est assimilable à un fluide, dont on modélise les propriétés dans l’espace et dans le temps.

Grandeurs microscopiques

Dans le cas d’une modélisation microscopique, on attribue à chaque véhicule un ensemble de variables telles que la position et la vitesse, qui vont évoluer en fonction du trafic et de certains paramètres du modèle, tel que le comportement du conducteur.

Grandeurs macroscopiques

Les grandeurs classiques liées à cette entité sont le débit, la concentration et la vitesse de l’écoulement, qui sont des fonctions du temps et de l’espace.

On appelle *débit*, en un point, le nombre de véhicules qui passent en ce point par unité de temps. Deux types de débits sont remarquables : le débit en entrée de tronçon, que l’on appellera *débit des arrivées* et le débit en sortie de tronçon, que l’on appellera *débit de sortie*. Une simplification commune à beaucoup de systèmes de régulation est de considérer que le débit de sortie est égal au *débit de saturation* (AFNOR, 1991), c’est-à-dire au débit maximal possible en sortie de tronçon en cas de présence d’une file d’attente infinie et feu au vert. En effet, le débit de sortie peut s’avérer complexe à calculer, car il dépend, en particulier, de gènes sur le carrefour, difficilement prévisibles. Les débits des arrivées pour un carrefour isolé, en revanche, peuvent être estimés à partir de prévisions statistiques plus fiables, par exemple en fonction de l’heure des arrivées.

Les notions de concentration et de vitesse de l’écoulement ne servant pas par la suite, nous n’en donnerons pas ici de définition. Le lecteur intéressé par ces notions pourra néanmoins se référer à (Cohen, 1990).

La *capacité* d’une voie (entrante) est le débit maximal que celle-ci est capable de laisser passer. En général, cette grandeur est utilisée lorsque la régulation se fait via un plan de feux. A un instant donné, elle vaut $\frac{\Delta T_{vert} \times DBS}{C}$, où ΔT_{vert} est la durée d’un signal vert sur la voie, DBS le débit de saturation de la voie, et C est le cycle du plan de feux. La *capacité d’un carrefour*, critère usuel pour la conception de carrefour, est la

somme des capacités de ses voies.

1.1.3.2 Capteurs

Pour mesurer les différentes grandeurs physiques liées à la situation du trafic, relativement à la modélisation choisie, il existe différents types de capteurs. En ce qui concerne les capteurs utilisés aux carrefours à feux, ils sont essentiellement de deux types :

- boucles électromagnétiques
- caméras vidéo

Les boucles électromagnétiques sont les capteurs les plus utilisés en France. Ils permettent en particulier de mesurer le débit à l'endroit où ils sont implantés. A l'aide de deux boucles successives, on peut également mesurer la vitesse des véhicules.

Les caméras vidéo, capteurs plus récents, permettent de fournir, grâce à des traitements d'images, les longueurs de files d'attente, mais aussi les débits aux endroits que l'on souhaite dans le champ de la camera, ainsi que la vitesse des véhicules.

1.1.4 Caractéristiques essentielles d'un système de régulation

Un système de régulation agit sur la couleur de feux tricolores pour minimiser un certain nombre de critères, sous certaines contraintes. On donne ici quelques caractéristiques fondamentales de tels systèmes.

1.1.4.1 Adaptabilité : perception et flexibilité

Une première caractéristique importante d'un système de régulation est sa capacité à prendre en compte de l'information sur le trafic de manière régulière et à pouvoir utiliser cette information sur ses prises de décision. En d'autres termes, il s'agit de sa capacité à s'adapter à la situation de trafic courante, qu'on appellera *adaptabilité*.

De par la complexité des systèmes mais aussi de la notion d'adaptabilité, il est difficile de proposer un classement des systèmes de régulation selon leur adaptabilité. Boillot et Braban (2003) proposent de distinguer les systèmes principalement par la fréquence de leur adaptation au trafic.

Selon nous, l'adaptabilité d'un système se compose des deux éléments suivants :

- la *perception* de l’environnement par le système, qui comprendra à la fois la diversité des informations considérées par le système et la fréquence de révision de ces informations.
- la *flexibilité* du système, qui sera sa capacité à exploiter l’ensemble des solutions admissibles, en terme de commutations des feux, par rapport au problème de la régulation du trafic aux carrefours à feux, posé en introduction de ce rapport.

Pour appartenir à la classe des systèmes appelés *adaptatifs*, il nous semble nécessaire qu’un système possède à la fois une bonne perception et une bonne flexibilité. La bonne perception est naturellement nécessaire puisque si l’on n’a pas ou peu d’information fournie en temps réel, on ne peut pas agir en conséquence. Mais la flexibilité aussi, peut-être dans une moindre mesure, semble être un élément important, car elle définit le champ des possibilités d’utilisation de l’information.

Selon cette perspective, on propose de distinguer trois classes de systèmes, décrits dans les paragraphes suivants.

Les systèmes non-adaptatifs

Les systèmes non-adaptatifs se caractérisent par :

- une mauvaise perception de l’environnement, qui repose au mieux sur des données statistiques d’évolution du trafic uniquement.
- une absence totale de flexibilité. Les solutions fournies en temps réel obéissent à un schéma défini en amont, avant la mise en route.

C’est le cas des systèmes qui utilisent des *plans de feux fixes*. Un plan de feu est généré “off-line”, c’est-à-dire avant la mise en service du système, puis il sera répété indéfiniment sans être modifié. Cependant, l’étude de tels systèmes n’est pas dénuée d’intérêt. En effet, on peut s’intéresser à la manière dont est généré le plan de feu initial. Dans certains cas, les méthodes utilisées pour bâtir le plan de feu sont assez élaborées.

Les systèmes semi-adaptatifs

Les systèmes semi-adaptatifs se caractérisent par :

- une bonne perception de l’environnement, qui repose sur des données fournies en temps réel, éventuellement couplées avec des données statistiques.
- une flexibilité moyenne. Les solutions fournies sont soumises à de fortes contraintes.

Parmi les contraintes classiques qui peuvent limiter la qualité des solutions fournies, sous l’hypothèse que ces solutions obéissent au schéma d’un plan de feu, on trouve essentiellement les limitations suivantes :

- Le cycle est fixé.
- Le phasage est fixé.
- Les durées de certaines phases sont fixées.
- Toute combinaison des trois contraintes précédentes.

Bien sûr, plus il y a de paramètres fixés, moins le système est à même de s’adapter à la situation présente, même s’il dispose d’informations très précises sur le trafic.

Ce type de système est assez fréquent, car limiter la flexibilité possède également des avantages. Cela permet, dans certains cas, et suivant la méthode d’optimisation utilisée, de gagner en temps de calcul.

Les systèmes adaptatifs

Les systèmes adaptatifs se caractérisent par :

- Une bonne perception, qui repose sur des données fournies en temps réel, éventuellement couplées avec des données statistiques.
- Une bonne flexibilité, ce qui leur permet de profiter pleinement des informations dont ils disposent.

Plusieurs systèmes proposent d’optimiser directement un ou plusieurs critères sans “recours aux plans de feux”. Autrement dit, ces systèmes sont conçus pour pouvoir prendre des décisions sur les états des feux indépendamment du concept de phases. Ainsi, l’utilisation des concepts de phasage et de cycle, tel qu’introduits dans la sous-section 1.1.2, disparaît également. On parle alors de systèmes acycliques, qui sont naturellement plus flexibles que les systèmes cycliques.

A noter que l’utilisation de systèmes acycliques, systèmes très flexibles, n’aurait pas beaucoup de sens sans une perception suffisamment bonne pour que de l’information soit disponible en temps réel afin de prendre des décisions aussi précises. Ceci explique qu’il n’y ait pas de classe qui correspondrait à des systèmes possédant une “mauvaise perception” et une “bonne flexibilité”.

1.1.4.2 Exactitude de la méthode d'optimisation

Il est bon de s'interroger également sur les prétentions exactes du système en terme de solution fournie. Les deux classes extrêmes sont les *méthodes exactes* et les *heuristiques*. Les méthodes exactes ont pour but de fournir une solution "optimale" (selon les critères utilisés et la façon dont ils sont agrégés). Elles sont souvent assez coûteuses en temps de calcul. Les heuristiques cherchent avant tout, en un temps très court, une solution réalisable, qui en général est un optimum local.

Il existe des classes intermédiaires, mais pas toujours faciles à apprécier, où un compromis est souvent recherché entre exactitude et rapidité. En effet, si une grande partie de cette caractéristique est classiquement déterminée lors du choix de la méthode d'optimisation, des techniques connues permettent parfois d'améliorer la rapidité d'exécution au détriment de l'"exactitude" de la solution. Par exemple, la méthode de l'horizon glissant en programmation dynamique permet cela.

1.1.4.3 Finesse de la modélisation

Une autre marge de manœuvre est celle de la modélisation. Elle peut être très détaillée pour représenter au mieux la réalité. Dans ce cas on a naturellement beaucoup de variables dans le système, d'où un temps de calcul possiblement grand.

La modélisation peut, au contraire, être très simplifiée, et dans ce cas elle donne lieu en général à un système beaucoup plus rapide. Un compromis est, là encore, nécessaire.

1.1.4.4 Vision locale ou globale, du point de vue du réseau routier

On peut s'interroger aussi sur l'étendue géographique pour laquelle a été dimensionné le système. Décide-t-on d'avoir une vision sur un seul carrefour (vision *locale*), sur un ensemble de carrefours (vision *zonale*), sur un réseau urbain en entier (vision *globale*) ?

Cependant, les systèmes à vision zonale ou globale, ne comportent pas forcément de méthode d'optimisation qui prend en compte tous les carrefours de la zone considérée en même temps. La plupart du temps, pour ces types de systèmes, l'optimisation est basée sur une collaboration heuristique des différents carrefours et non pas une optimisation globale. En général, les systèmes à vision globale comprennent un module de *gestion*

globale (qui ne fait pas d'optimisation), couplé à un module de gestion ou d'optimisation locale. Il existe aussi des systèmes qui font vraiment de l'optimisation globale.

1.1.4.5 Aspects multi-objectifs

Pour les systèmes qui sont destinés à prendre en compte plusieurs critères, il convient de regarder en premier lieu comment sont pris en compte ces critères. Quelle méthode est utilisée et quelle vision ont les auteurs de cet aspect fondamental de leur système ? Pour ce faire, nous introduisons quelques concepts fondamentaux de l'optimisation multi-objectif dans la section 1.2 suivante.

1.2 Concepts d'optimisation multi-objectifs

On introduit ici les notions nécessaires à la bonne compréhension des parties de la thèse où l'on discutera des méthodes d'optimisation multi-objectif, en se basant essentiellement sur (Vanderpooten, 1990), dans lequel on trouvera des compléments.

1.2.1 Formulation

Soit X un ensemble, appelé ensemble des actions potentielles et soit $f_1, \dots, f_p, p \geq 2$ des fonctions de X dans \mathbb{R} appelées fonctions objectif ou critères.

On appelle optimisation multi-objectif l'étude des problèmes (P) de la forme :

$$\begin{array}{ll} \min & f_1(x), \dots, f_p(x) \\ \text{s.c.} & x \in X \end{array}$$

Cette formulation ne signifie pas qu'il convient d'optimiser simultanément les p critères, ce qui n'est généralement pas possible. Elle peut s'interpréter de deux façons différentes :

- Déterminer l'ensemble de tous les compromis potentiels.
- Déterminer un compromis particulier.

Dans notre cas, nous nous intéresserons essentiellement à la détermination d'un compromis particulier.

X est également appelé espace de décision.

1.2.2 Dominance, efficacité

Soit $Z \subseteq \mathbb{R}^p$, l'espace des critères.

Soit z et z' deux éléments de Z . On note :

- $z' \leq z$ si et seulement si $\forall j \in \{1, \dots, p\}, z'_j \leq z_j$
- $z' < z$ si et seulement si $z' \leq z$ et $\exists j_0 \in \{1, \dots, p\}$ tq $z'_{j_0} < z_{j_0}$
- $z' \ll z$ si et seulement si $\forall j \in \{1, \dots, p\}, z'_j < z_j$

L'image de X dans Z est définie par :

$$Z_X = \{z \in Z : \forall j \in \{1, \dots, p\}, z_j = f_j(x), x \in X\}$$

Soit $z \in Z_X$, on dit que :

- z est *non dominé* si et seulement si il n'existe pas d'élément z' de Z_X tel que $z' < z$
- z est *faiblement non dominé* si et seulement si il n'existe pas d'élément z' de Z_X tel que $z' \ll z$

Soit $x \in X$, on dit que :

- x est *efficace* si et seulement si le vecteur critère associé à x est non dominé
- x est *faiblement efficace* si et seulement si le vecteur critère associé à x est faiblement non dominé

On appelle *frontière efficace* (resp. *faiblement efficace*) l'ensemble $E(X)$ (resp. $FE(X)$) des solutions efficaces (resp. faiblement efficaces). Pour leurs équivalents dans l'espace des critères, on parlera simplement de *l'ensemble* $ND(Z_X)$ (resp. $FND(Z_X)$) *des points non dominés* (resp. *faiblement non dominés*).

Il existe deux types de points non dominés :

- Les points non dominés *supportés* sont les points de $ND(Z_X)$ qui se situent sur l'enveloppe convexe de Z_X . Ils peuvent donc être trouvés à l'aide d'une agrégation linéaire des critères (Goeffrion, 1968).
- Les points non dominés *non supportés* sont les points de $ND(Z_X)$ qui ne sont pas supportés.

1.2.3 Point idéal, point nadir

Considérant le problème (P) , avec les mêmes notations, on appelle *point idéal* le vecteur $z^* = (z_1^*, \dots, z_p^*)$, où chaque z_j^* , $j \in \{1, \dots, p\}$ est l'optimum obtenu en résolvant le problème mono-objectif :

$$\begin{aligned} \min f_j(x) \\ \text{s.c. } x \in X \end{aligned}$$

On appelle "point nadir" le vecteur $n = (n_1, \dots, n_p)$ tel que chaque n_j , $j \in \{1, \dots, p\}$, correspond à l'optimum du problème suivant :

$$\begin{aligned} \max f_j(x) \\ \text{s.c. } x \in E(X) \end{aligned}$$

Ce problème est en général difficile. On peut obtenir une approximation du point nadir en construisant le tableau des coûts suivant.

z_1^*	...	\tilde{z}_j^1	...	\tilde{z}_p^1
...
\tilde{z}_1^j	...	z_j^*	...	\tilde{z}_p^j
...			...	
\tilde{z}_1^p	...	\tilde{z}_j^p		z_p^*

Où pour chaque $j \in \{1, \dots, p\}$, et $k \in \{1, \dots, p\}$, $k \neq j$, on a noté $\tilde{z}_j^k = f_j(\tilde{x}^k)$, avec \tilde{x}^k une solution du problème suivant :

$$\begin{aligned} \min f_k(x) \\ \text{s.c. } x \in X \end{aligned}$$

L'approximation z_* du point nadir est alors obtenue en considérant la valeur maximum de chaque colonne $j \in \{1, \dots, p\}$ du tableau des coûts : $(z_*)_j = \max_{k=1, \dots, p} \{\tilde{z}_j^k\}$. z_* est appelé *point anti-idéal*.

Il est bien connu que, généralement, les valeurs de z_* ne représentent pas les valeurs maximales potentiellement prises par chacun des critères correspondants. Pour une étude plus approfondie de cette notion, on pourra consulter (Isermann et Steuer, 1987).

Cependant, pour un point z_* donné, le pavé passant par les points z^* et z_* peut être vu comme une bonne approximation de l'emplacement et de l'étendue de la frontière efficace.

1.2.4 Paramètres préférentiels, fonction d'agrégation

Un problème d'optimisation multi-objectif s'accompagne, dans une grande partie des cas, d'un “décideur”, qui peut être le commanditaire de l'optimisation, et qui doit prendre un ensemble de décisions au regard du résultat obtenu.

1.2.4.1 Paramètres préférentiels

Pour prendre en compte les préférences du décideur, on introduit des *paramètres préférentiels* dont les trois principales formes sont :

- des paramètres intercritères : sous forme de “poids” accordés aux critères
- un point de référence : chacune de ses composantes représente une valeur souhaitée sur un critère (voir (Wierzbicki, 1999, 1980)).
- des valeurs maximales à ne pas dépasser pour chacun des critères (“*reservation levels*”)

1.2.4.2 Fonctions d'agrégation

Une fonction d'agrégation est une fonction $s : Z \rightarrow \mathbb{R}$, qui à un point de l'espace des critères donné associe une unique valeur réelle.

s peut éventuellement dépendre d'un paramètre préférentiel ω . On notera alors la fonction s_ω .

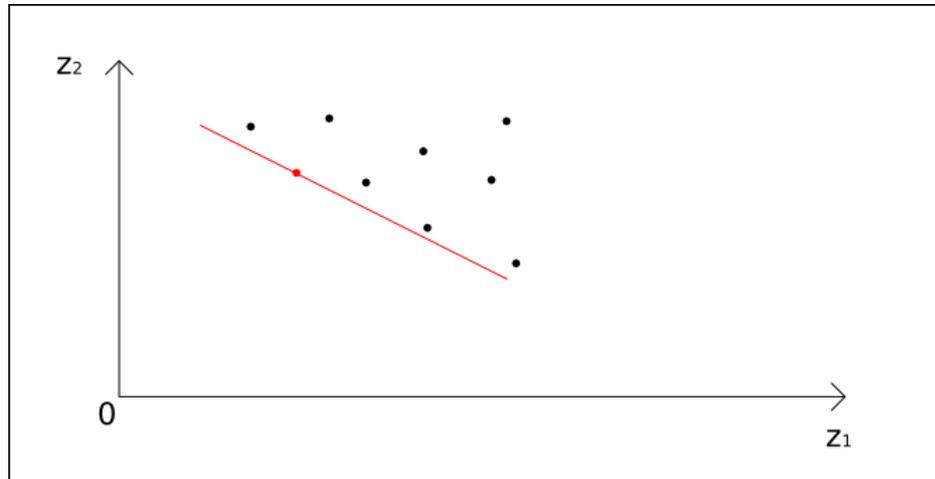
Une fonction d'agrégation s est dite :

- monotone croissante si : $\forall z, z' \in Y \subseteq Z, z > z' \Rightarrow s(z) \geq s(z')$
- strictement croissante si : $\forall z, z' \in Y \subseteq Z, z \gg z' \Rightarrow s(z) > s(z')$
- fortement croissante si : $\forall z, z' \in Y \subseteq Z, z > z' \Rightarrow s(z) > s(z')$

Les fonctions d'agrégation les plus classiques permettant de prendre en compte les paramètres préférentiels sont :

- La somme pondérée (figure 1.6) :

$$\sum_{j=1}^p \lambda_j z_j$$



- Points de Z_X
- Solution dans Z
- Représente la fonction à minimiser

FIGURE 1.6 – Optimisation avec la somme pondérée, avec $p = 2$

L'avantage de la somme pondérée est qu'elle ne génère que des points non dominés dès lors que l'on a $\lambda_j > 0$ pour $j \in \{1, \dots, p\}$ car cette fonction est fortement monotone. Cependant, les points non supportés ne pourront jamais être trouvés par cette méthode.

- Les fonctions à base de point de référence, dont le représentant le plus simple et le plus connu est :

$$g_{\bar{z}, \lambda}(z) = \max_{j=1..p} \{\lambda_j (z_j - \bar{z}_j)\}$$

où les poids $(\lambda_1, \dots, \lambda_p) = \lambda$ et le point de référence $\bar{z} = (\bar{z}_1, \dots, \bar{z}_p)$ sont les paramètres préférentiels.

Contrairement à la somme pondérée, cette fonction d'agrégation, qu'il convient de minimiser, permet d'atteindre tout point non dominé, en particulier les points non

supportés. En revanche, cette fonction n'est que strictement monotone, ce qui entraîne que les points qu'elle génère peuvent être faiblement non dominés.

- Pour remédier aux inconvénients de $g_{\bar{z},\lambda}$, on peut rajouter un terme supplémentaire à la fonction $g_{\bar{z},\lambda}$ afin d'obtenir la fonction suivante :

$$f_{\bar{z},\lambda}(z) = \max_{j=1..p} \{\lambda_j (z_j - \bar{z}_j)\} + \rho \sum_{j=1}^p \lambda_j (z_j - \bar{z}_j)$$

Cette fonction est appelée la forme *augmentée* de la précédente. $\rho > 0$ est une valeur très petite qui rend f fortement monotone, garantissant ainsi que tout point généré est non dominé. En revanche, si ρ n'est pas suffisamment petit, certains points non dominés peuvent ne pas être engendrés. Néanmoins, dans certains cas, ρ peut être fixé de manière à ce que chaque point non dominé reste atteignable. De plus, pour $\rho > 0$, chaque point généré par l'optimisation de la forme augmentée ne peut être faiblement dominé. Une illustration est donnée au niveau de la figure 1.7.

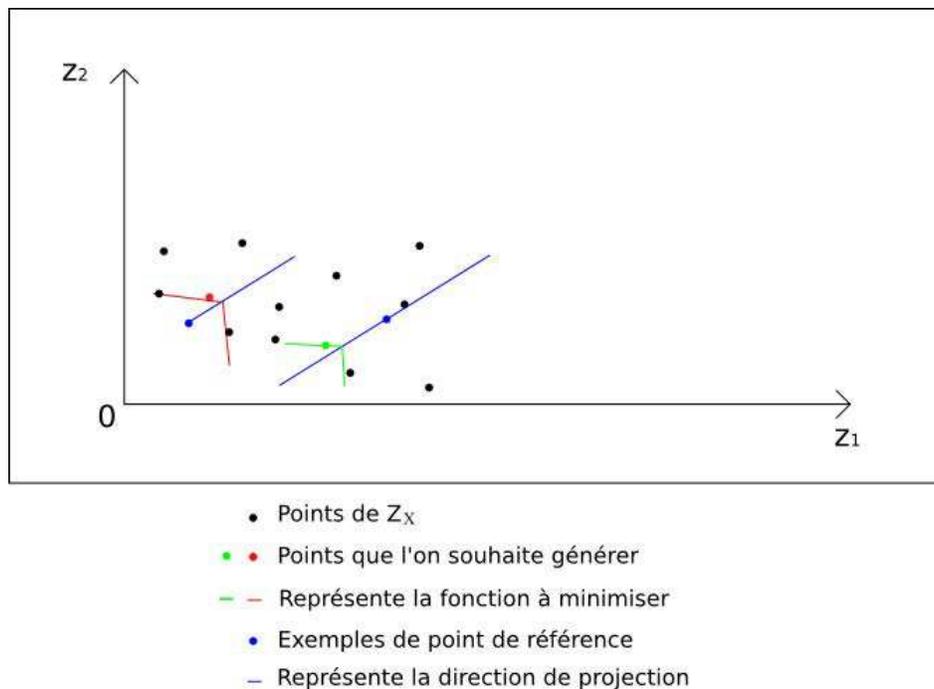


FIGURE 1.7 – Optimisation avec la forme augmentée de la méthode du point de référence. Dans le cas représenté en rouge, le point non dominé (rouge) n'est pas atteignable car ρ est trop élevé. Dans le cas représenté en vert, le point non dominé (vert), qui se trouve dans une situation similaire est atteignable en utilisant une valeur de ρ plus faible

Le calcul de ρ procurant les propriétés voulues pour les fonctions de type $f_{\bar{z},\lambda}$ peut être plus ou moins complexe selon le contexte. Le lecteur pourra consulter (Steuer et Choo, 1983), (Steuer, 1986) ou encore (Dächter *et al.*, 2012).

Dans notre contexte, où Z_X est discret, on obtient une valeur convenable de ρ assez facilement (voir Proposition 1).

Lemme 1. *Soit P un problème multi-objectif à $p \geq 2$ critères tel que défini en section 1.2.1 dont les critères prennent des valeurs entières non négatives. Soit m_j un minorant de la valeur prise sur chaque critère $j \in \{1, \dots, p\}$.*

Tout point non dominé $z' \in ND(Z_X)$ peut être généré comme solution unique de la fonction $f_{\bar{z},\lambda}$, en choisissant :

$$\rho < \frac{\min_{j \in \{1, \dots, p\}} \lambda_j}{\sum_{j \in \{1, \dots, p\} \setminus \{k\}} \lambda_j (z'_j - m_j) - \lambda_k}$$

où $k \in \arg \min_{j \in \{1, \dots, p\}} \{\lambda_j (z'_j - m_j)\}$.

Démonstration. Soit $z' \in ND(Z_X)$ un point non dominé quelconque. Posons $\bar{z} = z'$, on a alors $f_{\bar{z},\lambda}(z') = 0$. Pour que z' soit solution optimale unique, on doit avoir $f_{\bar{z},\lambda}(z'') > 0$ pour tout autre point $z'' \in Z_X$. Observons que compte tenu de l'intégrité de l'espace des critères il existe nécessairement un indice $j \in \{1, \dots, p\}$ tel que $z''_j \geq z'_j + 1$, sinon z'' dominerait z' . On a donc $\max_{j \in \{1, \dots, p\}} \{\lambda_j (z''_j - z'_j)\} \geq \min_{j \in \{1, \dots, p\}} \lambda_j$. Pour assurer que $f_{\bar{z},\lambda}(z'') > 0$, on doit donc choisir ρ de façon à ce que $\min_{j \in \{1, \dots, p\}} \lambda_j + \rho \sum_{j \in \{1, \dots, p\}} (z''_j - \bar{z}_j) > 0$, quel que soit le point $z'' \in Z_X$. Le pire cas intervient pour un éventuel point $z'' = (m_1, \dots, z'_k + 1, \dots, m_p)$ où k est tel que $\lambda_k (z'_k - m_k) = \min_{j \in \{1, \dots, p\}} \{\lambda_j (z'_j - m_j)\}$. \square

Proposition 1. *Soit P un problème multi-objectif à $p \geq 2$ critères tel que défini en section 1.2.1 dont les critères prennent des valeurs entières non négatives. Soit respectivement m_j et M_j un minorant (resp. majorant) de la valeur prise sur chaque critère $j \in \{1, \dots, p\}$.*

Tout point non dominé peut être généré comme solution unique de la fonction $f_{\bar{z},\lambda}$, en choisissant $\rho = \frac{\min_{j \in \{1, \dots, p\}} \lambda_j}{\sum_{j=1, \dots, p} (M_j - m_j)}$

Démonstration. Pour tout $z' \in ND(Z_X)$ on note $\bar{\rho}(z') = \frac{\min_{j \in \{1, \dots, p\}} \lambda_j}{\sum_{j \in \{1, \dots, p\} / \{k\}} \lambda_j (z'_j - m_j) - \lambda_k}$, où $k \in \arg \min_{j \in \{1, \dots, p\}} \{\lambda_j (z'_j - m_j)\}$. On a alors, pour chaque point non dominé $z' \in ND(Z_X)$, $\rho \leq \bar{\rho}(z')$. Alors, nous avons grâce au Lemme 1 que chaque point non dominé $z' \in ND(Z_X)$ est atteignable. \square

En pratique, il est plus simple d'utiliser $\rho = \frac{\min_{j \in \{1, \dots, p\}} \lambda_j}{\sum_{j=1, \dots, p} (M_j - m_j)}$, de la Proposition 1, qui ne fait pas référence aux points que l'on génère, que ρ du Lemme 1, qui lui est fonction de ces points. Si un "bon choix" de M est facile à déterminer (dans le sens où chaque M_j n'est pas trop grand), nous préférons ainsi fixer la valeur de ρ à celle fournie par la Proposition 1.

1.2.4.3 Utilisations usuelles des fonctions de type $f_{\bar{z}, \lambda}$

Plutôt que de faire varier l'ensemble des paramètres préférentiels, on peut choisir de faire varier uniquement une partie de ceux-ci.

Par exemple, dans la *méthode de Tchebychev*, on fixe \bar{z} , et on fait évoluer λ . En général, \bar{z} correspond à z^* ou à un point qui domine légèrement z^* . Dans un contexte où l'instance du problème peut évoluer dans le temps, z^* évolue également. \bar{z} doit alors être révisé régulièrement pour garantir que $\bar{z} \ll z^*$.

Dans la *méthode du point de référence*, en revanche, on fixe λ et on fait évoluer \bar{z} .

1.2.5 Procédures interactives

Supposons que l'on dispose d'un décideur. On appelle *procédure interactive* la procédure qui consiste à alterner les deux phases suivantes :

- On propose une solution au problème multi-objectif qui nous intéresse, en utilisant une fonction d'agrégation donnée, avec les paramètres préférentiels courants.
- Ces valeurs sont présentées au décideur, qui en retour, indique s'il est satisfait ou non. S'il n'est pas satisfait, il devra fournir des informations permettant de faire évoluer les paramètres préférentiels.

On itère le procédé jusqu'à ce que le décideur soit satisfait d'une des solutions obtenues. Une schématisation du procédé est proposée figure 1.8. Le lecteur pourra se référer à (Vanderpooten, 1989; Gardiner et Vanderpooten, 1997) pour plus de détails.

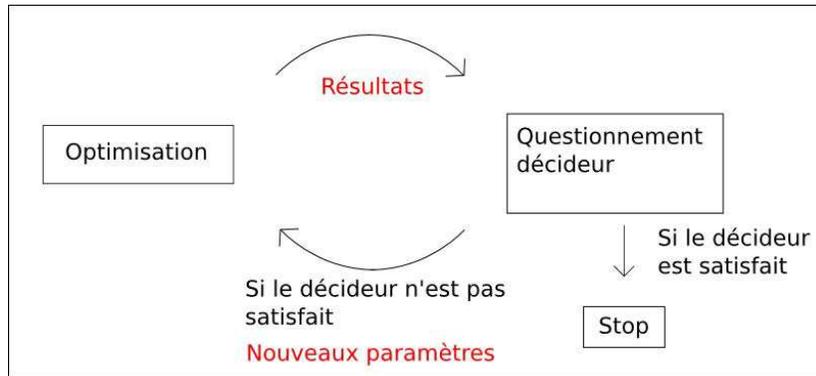


FIGURE 1.8 – Procédure interactive. En rouge, les informations transmises pour la nouvelle étape

Ce processus est usuellement mis en œuvre dans un contexte où l’instance du problème n’évolue pas dans le temps (voir, par exemple, (Perny et Vanderpooten, 1998)).

1.2.6 Procédures interactives temps réel

On entend par *procédure interactive temps réel* une procédure interactive utilisée en temps réel où l’instance du problème multi-objectif associé est susceptible d’évoluer (dans le temps). Les règles d’évolution de l’instance (figure 1.9), dépendent à la fois des solutions fournies aux itérations précédentes et du monde physique dans lequel est plongé le système, que l’on appellera *environnement*. De plus, si la procédure doit s’exécuter avec un temps de résolution limité, il se peut que l’on n’ait pas suffisamment de temps pour interroger une personne physique apte à fournir les informations nécessaires à l’évolution des paramètres. Le décideur peut alors être remplacé par un ensemble de *règles de décision*, établies avec le décideur, qui auront pour rôle de le remplacer, autant que faire se peut, dans la procédure. D’éventuelles conditions d’arrêt, à définir avec le décideur, peuvent également être ajoutées aux règles de décision.

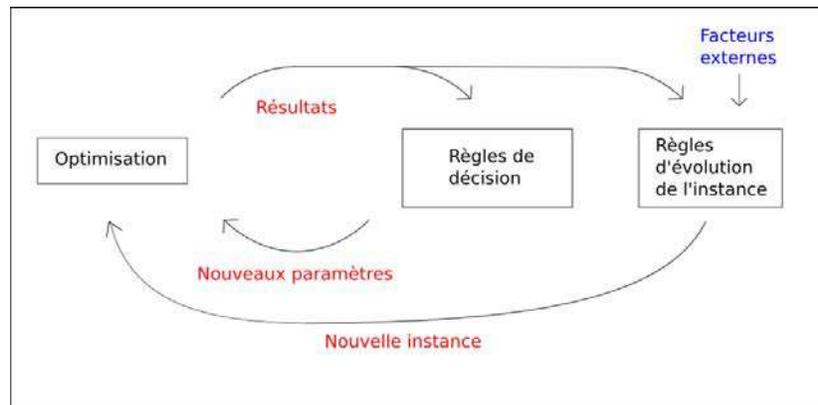


FIGURE 1.9 – Procédure interactive temps réel. En rouge, les informations transmises pour la nouvelle étape

L'évolution de l'instance dépend à la fois de l'application de la solution courante à l'environnement, et d'un certain nombre de "facteurs externes" au modèle. Ces facteurs externes sont les éléments de l'environnement qui peuvent influencer l'instance, mais qui ne sont pas pris en compte dans le modèle.

De plus, dans la mesure où l'on se place dans un contexte où l'environnement évolue en permanence, on n'indique pas de sortie du processus.

Comme l'instance évolue, l'ensemble des points non dominés peut évoluer.

En ce qui concerne les *règles de décision*, on a deux exigences principales :

- elles doivent prendre en compte les préférences d'un décideur.
- elles doivent prendre en compte l'évolution de l'environnement.

Mais on peut aussi avoir, dans le contexte de la thèse, les exigences supplémentaires suivantes :

- Les règles de décision doivent être suffisamment simples pour qu'un non spécialiste puisse exprimer ses préférences facilement.
- Les règles de décision peuvent facilement être changées une fois que le système est lancé, sans remettre en question le bon fonctionnement du système.

1.3 Problématique de la thèse

Comme nous l'avons évoqué en introduction, le problème de régulation du trafic, posé dans le cadre de cette thèse, peut être formulé comme un problème d'optimisation multi-objectif.

Il s'agit en effet de décider, en temps réel, de la couleur des feux d'un carrefour isolé, relativement à plusieurs enjeux, qui seront détaillés en sous-section 1.3.1, et qui donnent lieu à des critères bien précis à minimiser (sous-section 1.3.2). De plus il convient de respecter un certain nombre de contraintes liées à la sécurité des usagers du carrefour.

Dans un premier temps il est nécessaire de modéliser le mieux possible le carrefour et les usagers utilisant le carrefour. Plus notre représentation de la réalité est fidèle, plus on a de chances d'avoir une action efficace sur le monde réel. A partir de cette modélisation de l'environnement, on pourra construire des modèles (des programmes linéaires en ce qui nous concerne) qui vont permettre de poser le problème mathématiquement, et qui correspondent chacun respectivement à une approche différente.

La nature du problème (multi-objectif, temps réel) appelle la mise en place d'une procédure interactive temps réel décrite en sous-section 1.2.6. Chacun des modèles devra pouvoir résoudre en temps réel le problème relatif à l'instance de l'itération courante. Le fait que cette optimisation doive se faire dans un contexte temps réel implique que, si on veut espérer obtenir des résultats intéressants, pour une instance donnée, la résolution ou au moins une partie de celle-ci se fasse en temps raisonnable. Plusieurs approches seront proposées pour répondre à ce dernier point, qui est un point crucial.

Enfin, des descriptions précises de l'évolution de l'instance et des paramètres préférentiels, dans la procédure interactive temps réel, font partie intégrante de la problématique. On parlera de *politique de régulation* pour désigner l'ensemble des paramètres préférentiels relatifs à une itération donnée dans la procédure, et de *politique générale de régulation* pour désigner les règles générales qui régissent l'évolution de ces paramètres.

Les deux sous-sections suivantes décrivent les enjeux pris en compte dans la thèse et les critères qui en découlent.

1.3.1 Les enjeux

Les enjeux importants et pris en considération dans cette thèse sont : la *sécurité*, la *fluidité*, le *confort* et l'*environnement*.

- On entend par sécurité la sécurité des personnes qui doivent emprunter le carrefour : piétons, véhicules particuliers, transport en commun.
- La fluidité est la fluidité du trafic sur le carrefour, dont la mesure habituelle est le temps d'attente global des véhicules à la traversée de celui-ci.
- Le confort peut être vu comme le temps d'attente moyen, durant un horizon de temps, des véhicules particuliers sur le carrefour. Mais il peut également être traduit dans le nombre d'arrêts (voir (Cohen, 1990)).
- Le facteur environnemental, quant à lui, est relatif à l'émission des différents agents polluants par les véhicules lors de leur traversée du carrefour.

De plus, les transports en commun (bus) doivent être régulés “séparément”, de sorte qu'on puisse établir une priorité entre la fluidité des bus et des véhicules particuliers. Cet aspect peut être vu comme un enjeu à part ou bien comme l'expression indirecte de l'enjeu environnemental cité précédemment, puisque “favoriser” les transports en commun pourrait permettre, à terme, d'inciter les utilisateurs des réseaux routiers à les emprunter plutôt qu'un véhicule particulier, réduisant ainsi le nombre global de véhicules circulant sur les routes.

1.3.2 Les critères

A partir des différents enjeux cités précédemment, on peut construire des critères, tous à minimiser, reflétant globalement les enjeux.

On peut regrouper l'enjeu de fluidité et la partie temps d'attente de l'enjeu confort, en un seul critère, ne concernant que les véhicules particuliers, et appelé *temps total d'attente (TA)*. Ce critère correspond à la somme des temps d'attente de chaque véhicule sur un horizon de temps.

Le terme total signifie que l'on parle de la *somme des temps d'attente de chaque véhicule* présents sur le carrefour pendant l'horizon de temps. Cependant, s'il n'y a pas d'ambiguïté possible, on utilisera aussi le terme “temps d'attente”.

L'émission de polluants est liée à la fois au temps d'attente mais aussi au nombre d'arrêts effectués sur le carrefour, pendant un horizon de temps donné (voir (Midenet *et al.*, 2004)). Le temps d'attente fait déjà partie de nos critères. De plus, la relation entre ce temps d'attente et le nombre d'arrêts est complexe et dépend à la fois du polluant visé et de la nature des véhicules (modèle, année, entretien, etc.). Il serait donc périlleux et contestable de chercher à agréger a priori ces deux aspects pour bâtir un critère reflétant l'impact sur la pollution. Pour ces raisons, nous retiendrons que le *nombre d'arrêts (NA)* doit être considéré séparément du temps d'attente. De plus, il ne concernera que les véhicules particuliers, responsables de la majeure partie de l'émission de gaz polluants sur un horizon de temps. Ce critère peut également traduire un confort de conduite.

Pour traiter le cas des transports en commun, nous introduirons un troisième critère, qui sera explicité dans la modélisation (chapitre 3) : l'*erreur totale de consigne pour les bus* ou, s'il n'y pas d'ambiguïté possible, *erreur de consigne pour les bus (ECB)*. On part du principe qu'une ou plusieurs compagnies de bus fournissent une demande relative à des contraintes horaires pour chaque bus (consignes). Pour simplifier, on considère qu'il n'y a qu'une seule compagnie, mais cela ne change rien à la façon de traiter le problème.

1.4 Conclusion du chapitre

Dans ce chapitre, nous avons posé les bases des notions utiles pour la suite du rapport. En particulier, nous avons proposé une grille de lecture permettant de caractériser un système de régulation.

Dans le chapitre suivant, nous allons étudier un certain nombre de systèmes de régulation existants, en essayant de les analyser selon les caractéristiques fondamentales introduite en sous-section 1.1.4.

Chapitre 2

État de l'art des systèmes et des méthodes d'optimisation utilisées dans la régulation du trafic

Dans ce chapitre, on présente différents systèmes de régulation de trafic, classés suivant leur adaptabilité. Trois niveaux d'adaptabilité seront distingués - non adaptatif, semi-adaptatif et adaptatif - donnant lieu respectivement à trois sections. Les systèmes mono-objectifs et multi-objectifs sont traités séparément à l'intérieur de chaque section. Enfin, on essaiera, dans la mesure du possible, de décrire, pour chaque système, toutes les autres caractéristiques énoncées en sous-section 1.1.4, ainsi que la méthode d'optimisation employée. Les systèmes à vision zonale ou globale seront abordés essentiellement à travers leur aspect local, c'est-à-dire qu'on s'attardera plus sur les caractéristiques de la régulation au niveau d'un carrefour.

2.1 Systèmes non-adaptatifs

Bien qu'une multitude de systèmes existent pour assurer la gestion du trafic aux carrefours à feux, la méthode la plus fréquemment utilisée en pratique est celle utilisant les plans de feux fixes. Les systèmes utilisant ce genre d'approche consistent essentiellement en :

- l'élaboration d'un ou de plusieurs plan(s) de feux, avant application

- l’application du ou des plan(s) de feux sur le terrain

Le fait d’utiliser un plan de feux fixe, ou même de plusieurs, n’est pas anodin. En effet, les systèmes utilisant ce genre d’approche sont incapables de faire face à une évolution non prévue du trafic. Il ne sont adaptés qu’à une seule situation de trafic bien précise, parmi l’ensemble de toutes les situations possibles, qui peuvent être extrêmement nombreuses dans l’absolu. L’hypothèse principale qui sous-tend ces systèmes consiste à considérer qu’on n’a que quelques situations de trafic type dites nominales.

Nous distinguerons les systèmes ou méthodes dimensionnés pour faire de l’optimisation mono-objectif, des systèmes multi-objectifs.

2.1.1 Systèmes mono-objectifs

Plans de feux fixes déterminés analytiquement suivant une méthode empirique

Webster (1958) a montré qu’on pouvait calculer, sous certaines hypothèses, à partir d’une simple formule, la “formule de Webster”, le cycle “optimal” d’un plan de feux fixe, ainsi que la répartition des temps de verts adéquat. Ce cycle est “optimal” selon le critère du temps d’attente subi par les véhicules, tous véhicules confondus (à minimiser). Autrement dit, il calcule le plan de feux dans son intégralité de manière purement analytique. Ajoutons que la formule contient également des coefficients empiriques.

L’avantage qui apparaît immédiatement est sa simplicité. On peut rajouter aussi le fait que le temps de calcul demandé on-line est nul, puisque tout est calculé par avance.

Cependant, plusieurs objections simples peuvent être faites, portant sur les hypothèses posées, qui sont, de toute évidence, très fortes :

1. On n’envisage qu’un seul type de phasages.
2. On ne tient pas compte des variations du trafic autour de la valeur nominale ni d’une dérive au cours du temps de cette valeur (sur de grandes périodes de temps). En particulier, les “charges” calculées sont fixées et immuables. La charge d’un tronçon, à un instant t_0 , est le débit d’arrivée divisé par le débit de saturation à $t = t_0$.
3. L’optimisation ne porte que sur un seul objectif.

4. Il n’y a, en fait, pas de garantie sur l’optimalité de la solution trouvée. En effet, les paramètres nécessaires dans la formule sont obtenus *empiriquement* pour “satisfaisant” le plus “grand nombre” de cas possibles. Comme on ne peut pas être à l’optimum simultanément pour toutes les situations de trafic, il y a peu de chance que l’on soit à l’optimum pour l’une d’entre elles choisie au hasard.

Comme dans la pratique on utilise une bibliothèque de plans de feux adaptés à la charge moyenne sur une période donnée de la journée, de la saison ou de l’année, ou à un événement particulier (sortie d’un stade par exemple), on peut dire que la deuxième objection n’est que partiellement justifiée. Cependant, on ne peut pas encore parler de système semi-adaptatif, c’est-à-dire de système capable de s’adapter partiellement à la situation. Pour cela, il faudrait que la situation du trafic soit prise en compte en temps réel dans la modification du plan de feux.

Détermination de plans de feux fixes par programmation linéaire et par la théorie des graphes

Cantarella et Improta (1984) adoptent une approche basée sur la programmation linéaire mixte (utilisant des variables binaires et entières) permettant de s’affranchir partiellement du point 1 du paragraphe précédent. Les feux sont regroupés en “groupes de feux”, dont les membres sont mutuellement non antagonistes. Ce ne sont pas des phases, dans le sens où ces groupes ne correspondent pas forcément à des stables *maximaux* du graphe des antagonismes (mais correspondent à des stables tout de même, dans le graphe des antagonismes). Il peut alors exister des groupes compatibles et d’autres incompatibles, tout comme pour les feux considérés individuellement. Chaque groupe de feu reçoit le même signal en même temps. Les variables principales sont des variables binaires w_{ij} , chacune associée à une paire (i, j) de *groupes antagonistes*, c’est-à-dire tels que leurs feux respectifs ne peuvent être mis au vert en même temps, qui prennent la valeur 1 si et seulement si le vert associé au groupe j succède au vert associé au groupe i .

L’intérêt de considérer de tels groupes de feux réside principalement dans l’utilisation d’un plus petit nombre de variables que dans les deux cas extrêmes suivants :

- Chaque groupe de feux correspond à un seul feu. Le nombre de variables utilisées est alors nécessairement plus élevé, avec une matrice des antagonismes conséquente.

- Chaque groupe de feux correspond à un stable maximal du graphe. Le nombre de variables utilisées peut alors être exponentiellement élevé par rapport au nombre de feux.

La notion de phasage, dans la méthode, diffère légèrement de la notion introduite en sous-section 1.1.2, puisqu’il s’agit, pour les auteurs, de solliciter *exactement une fois* chaque feu durant un cycle (et non pas *au moins une fois*). Le calcul de sa durée, qui se trouve alors être une variable, ainsi que le calcul des temps de vert, reposent entièrement sur les calculs de Webster.

L’optimisation se fait off-line. Le plan de feu ainsi calculé, optimisé pour une certaine fonction objectif, et un certain état du trafic, sera répété indéfiniment lors de l’application réelle au carrefour.

Enfin, plusieurs types d’objectifs ont été testés (en mono-objectif), comme le temps d’attente total (à minimiser), la capacité (à maximiser) ou encore la durée du cycle (à minimiser).

Peu de temps après, les mêmes auteurs ont proposé, cette fois, une approche utilisant la théorie des graphes, dans le but de maximiser la capacité du carrefour ou de minimiser le cycle du plan de feux (Cantarella et Improta, 1988). La méthode d’optimisation consiste à obtenir une orientation adéquate d’un graphe initialement non-orienté, dont chaque sommet correspond à un groupe, et dont une arête relie deux sommets si et seulement si les groupes associés sont incompatibles, c’est-à-dire que les flux de véhicules correspondants ne peuvent circuler en même temps pour des raisons de sécurité. En reprenant les mêmes notations que dans l’article précédent, une orientation de i vers j dans ce graphe correspond à l’attribution de la valeur 1 à la variable $w_{i,j}$. La résolution se fait par un Branch & Bound. Des considérations de théorie des graphes leur permettent cependant d’éliminer certaines solutions a priori.

L’approche par regroupement de feux est fréquente pour les systèmes non-adaptatifs. Pour plus de détails sur la notion de groupes, voir (Gallivan et Heydecker, 1988; Dudgeon et Heydecker, 1993; Silcock, 1997), où il est fait état, en particulier, du gain de flexibilité procuré par l’approche par groupes plutôt que par phases.

Conception géométrique et détermination de plans de feux fixes par programmation linéaire

Wong et Wong (2003) proposent une approche par programmation linéaire mixte pour résoudre *simultanément* les problèmes de la régulation de trafic et de la “conception géométrique” de carrefour (isolé), ce qui le démarque de la grande majorité des systèmes traités dans ce chapitre. En effet, la conception géométrique de carrefour et l’optimisation des plans de feux sont, en général, traités indépendamment l’un de l’autre.

Dans la conception géométrique d’un carrefour on s’autorise à modifier physiquement le carrefour pour satisfaire certaines contraintes. Les variables de décision correspondantes peuvent être nombreuses et variées, mais dans le contexte de l’article, il s’agit des marquages au sol sur les voies (flèches directionnelles), qui donneront lieu à l’implantation des feux correspondants.

Les critères considérés (tous deux en mono-objectif) sont :

- la capacité (à maximiser).
- le cycle (à minimiser).

Le temps d’attente total n’est pas traité car non-linéaire. Cependant les auteurs évoquent l’idée d’une linéarisation par morceaux.

2.1.2 Systèmes multi-objectifs

Système TRANSYT

Un système très connu et possédant une vision globale, est le système TRANSYT (Robertson, 1969). La méthode utilise des bibliothèques de plans de feux disponibles pour chaque carrefour du réseau. De plus, plusieurs critères sont implicitement considérés, agrégés par somme pondérée. Le but est de trouver une solution optimale selon ce type d’agrégation, établie *a priori* (c’est-à-dire qu’on ne revient jamais sur la valeur des coefficients de la somme pondérée), qu’on appellera *super-critère*.

Pour construire ces plans de feux le système considère l’heuristique itérative suivante : à la première étape, toutes les caractéristiques des plans de feux de tous les carrefours sont fixées (phasage, cycle, durées des phases) sauf pour un carrefour, dont il modifie éventuellement le *décalage*, qui consiste à décaler toutes les dates de commutations futures de la même durée. Le décalage s’effectue selon les valeurs que peut prendre le super-critère en fonction des arrivées correspondantes sur ses tronçons. Puis il passe à un autre carrefour, et ainsi de suite jusqu’à la visite de tous les carrefours du réseau. Ensuite, la même procédure est appliquée, mais en modifiant cette fois, la

“*durée des verts*”, autrement dit la durée de chaque phase du plan de feux, pour chaque carrefour successivement. Ces deux visites du réseau correspondent à une itération. A chaque itération, les plans de feux calculés sont mémorisés pour les itérations suivantes.

Le critère à optimiser est une combinaison linéaire d’un ensemble de caractéristiques du réseau, à savoir les retards “totaux” (sur l’ensemble des tronçons du réseau) et le nombre “total” d’arrêts.

Il est intéressant de constater qu’un poids est attribué à chaque retard *sur chaque tronçon* ainsi qu’à chaque nombre d’arrêts *sur chaque tronçon*. Il est indiqué que chaque poids représente l’influence du tronçon correspondant sur le retard ou sur le nombre d’arrêt. Leur valeur ne dépend pas d’une politique de régulation mais de l’“importance intrinsèque” du tronçon, par rapport soit au retard, soit au nombre d’arrêts. On peut donc considérer que ces poids font partie intégrante de la méthode heuristique utilisée dans ce système pour résoudre le problème de régulation et ne correspondent pas à une politique de régulation. Ainsi, il n’existe que deux critères dont l’importance peut réellement refléter une politique de régulation : le retard total et le nombre d’arrêts, ce qui permet tout de même au système d’appartenir, selon nous, aux systèmes multi-objectifs.

Amélioration du système TRANSYT utilisant une méthode d’optimisation heuristique

Wong (1996) propose d’améliorer un peu le système TRANSYT en optimisant simultanément toutes les variables utilisées dans TRANSYT (cycle, durée des verts, décalages) via une formulation par programmation mathématique. Les phasages d’un petit nombre de carrefours peuvent être modifiés pour comparer les résultats obtenus, mais le changement de phasage n’est pas inclus dans la méthode d’optimisation. Ces carrefours sont les carrefours considérés comme “critiques” pour le réseau.

Les critères sont les mêmes que pour le système TRANSYT. La fonction objectif est une combinaison linéaire des critères.

La méthode utilisée pour résoudre ce programme non-linéaire et non-convexe, est une heuristique mêlant programme linéaire mixte et méthode de descente.

Résumons la méthode d’optimisation : dans la formulation du problème, les variables sont à la fois des variables entières et des variables continues, la fonction objectif n’est pas linéaire, et les contraintes sont toutes linéaires. La fonction objectif est remplacée

par le produit scalaire de son gradient au point courant et de la direction de descente cherchée, et les contraintes sont modifiées en conséquence. Il reste à résoudre le problème linéaire correspondant, qui constitue tout de même un problème difficile car les variables restent entières. Le pas de la descente est ensuite optimisé également via un programme linéaire mixte. La méthode est réitérée jusqu'à atteindre un optimum local. Ce genre d'approche itérative, qui consiste en l'approximation d'une fonction objectif non-linéaire par l'hyperplan tangent en un point courant, est bien connue dans le domaine de l'optimisation non-linéaire (méthode du gradient). On pourra consulter à ce sujet (Culioli, 1994), page 50.

Malgré la flexibilité significativement plus élevée que celle du système TRANSYT, le système est un système non-adaptatif, pour plusieurs raisons :

- La modélisation ne prend en compte aucune donnée provenant de capteur. Le système n'a donc aucune perception de l'environnement en temps réel.
- Les temps de calcul sont assez élevés, et comparables aux temps de calcul de TRANSYT (plusieurs minutes pour l'exemple donné dans l'article correspondant). Des travaux postérieurs (Wong, 1997) ont d'ailleurs porté sur l'amélioration du temps de calcul en parallélisant certaines tâches.

Système utilisant la programmation dynamique multi-objectif

Reljic (1996) modélise un réseau de transport par un graphe orienté, où chaque sommet représente une intersection et chaque arc (i, j) représente une voie orientée de l'intersection i à l'intersection j . Partant du principe que chaque intersection utilisera un plan de feux fixe dont tous les paramètres sont déjà déterminés sauf le décalage, Reljic propose d'optimiser le choix de ces *décalages* (*qui sont donc les seules variables*) en fonction de plusieurs critères à minimiser :

- Le temps d'attente total
- Le nombre total d'arrêts
- La consommation totale
- Le "coût total"

Les deux derniers critères sont des combinaisons linéaires des deux premiers. La méthode choisie est la programmation dynamique, qui portera, en réalité, non pas sur le réseau entier, mais sur les circuits simples du graphe (circuits dont un même arc ne figure pas deux fois), considérés séparément. Le nombre d'étapes de la méthode est la longueur du

circuit considéré. La motivation principale pour utiliser la programmation dynamique est que les critères sont des fonctions non-linéaires des variables de décision.

Pour résoudre ce problème multi-objectif, l'auteur n'utilise pas d'agrégation a priori. Il procède en deux phases.

Dans un premier temps, il obtient les solutions efficaces en utilisant la programmation dynamique généralisée à la programmation dynamique multi-objectif (Daellenbach et Kluyver, 1980; Karwan et Villarreal, 1982). Dans un second temps, il va sélectionner la "meilleure" solution, d'un certain point de vue, parmi toutes les solutions efficaces calculées explicitement. Trois méthodes sont évoquées :

- Une agrégation de type $\sum_{j=1,\dots,p} \lambda_j \frac{z_j - z_j^*}{z_j^*}$
- Une agrégation de type $\max_{j=1,\dots,p} \lambda_j \frac{z_j - z_j^*}{z_j^*}$
- La méthode de surclassement ELECTRE (Roy, 1968) - il s'agit de la méthode ELECTRE I.

Pour les deux premières méthodes, les notations sont celles introduites dans la section 1.2. On choisit la solution qui obtient la valeur la plus petite pour la fonction d'agrégation considérée.

2.2 Systèmes semi-adaptatifs

On rappelle qu'un système de régulation semi-adaptatif est un système qui ne s'adapte que partiellement à la situation de trafic. En général, ces systèmes s'appuient sur un plan de feux initial fixé mais modifiable, pour effectuer leur optimisation. Le plan de feux est modifié partiellement, en temps réel, en fonction de la situation du trafic.

Nous distinguerons les systèmes ou méthodes dimensionnés pour faire de l'optimisation mono-objectif, des systèmes multi-objectifs.

2.2.1 Approches mono-objectifs

Système SCATS

Le système australien SCATS (Lowrie, 1982), à *vision globale*, fonctionne sur deux niveaux. Le niveau stratégique vise à une gestion globale du réseau et fonctionne à partir de bibliothèques de cycles, de décalages et de durées de vert. Le niveau tactique

permet de modifier localement les *durées de vert* ou bien de faire de l'*escamotage* de phase (qui consiste à supprimer ou à rajouter une phase dans le plan de feux). La régulation tactique s'effectue à partir d'informations collectées en temps réel sur les distances inter-véhiculaires, c'est-à-dire les distances qui séparent chaque véhicule de celui qui le suit, sur chaque tronçon. Le critère au niveau tactique est donc du type *intervalle véhicule* sur chacun des tronçons.

Systeme utilisant une methode de gradient

Dotoli *et al.* (2006) proposent un systeme temps reel à base de plan de feux modifiables, avec une *vision globale*. Les variables de decision sont les *durées de vert*, les *décalages* et le *cycle*, le cycle étant commun à tous les carrefours du reseau.

Les auteurs formulent le probleme sous la forme d'un programme mathematique non-lineaire.

Le critere, à minimiser, est le nombre de vehicules presents sur les differents tronçons du reseau sur un horizon de temps. Ce critere est lineaire, dans la modelisation choisie. En revanche, les contraintes sont en grande partie non-lineaires et non-convexes. Les non-linearités sont de differentes natures (quadratiques, fractionnaires, expressions disjonctives, presence d'operateurs de type min/max, etc.), que les auteurs ne cherchent pas à lineariser.

Ces derniers ont au contraire opte pour une methode d'optimisation de problemes non-lineaires : la "methode de gradient reduit generalise". Cette methode est une methode de descente tres utilisee dans le domaine du transport et de la distribution d'energie électrique via un reseau. Le lecteur pourra consulter (Lasdon *et al.*, 1973) pour plus de details.

Les auteurs indiquent que la methode est suffisamment "rapide" (mais ils ne fournissent pas de resultats experimentaux) pour permettre une utilisation temps reel. Même si les auteurs ne le mentionnent pas, rappelons que, comme pour toute methode de descente appliquee à un probleme non-convexe, on n'obtient que des solutions locales. Pour ce type d'instance, il s'agit donc d'une *heuristique*.

Systeme à vision globale avec des intersections simples utilisant la programmation lineaire mixte

Wey (2000) propose une formulation par programmation linéaire mixte au problème de régulation de trafic sur un réseau. Il se situe dans le cadre d'une *vision globale*.

De plus, l'*optimisation* est globale. Pour ce faire, l'auteur utilise non pas une méthode de "branch and bound" qui appelle classiquement l'algorithme du simplexe à chaque nœud, mais utilise une "version réseau du simplexe" ("network simplex algorithm", (Ahuja et Orlin, 1992)), réduisant ainsi le nombre de pivots de la méthode standard du simplexe (Dantzig *et al.*, 1954) à un nombre plus raisonnable pour un problème possédant une structure de type réseau. Les résultats obtenus sur une utilisation zonale (5 intersections), en termes de temps de calcul (entre 10 et 20 secondes par optimisation), permettent alors d'envisager, selon l'auteur, une application temps réel du système.

Pour la formulation du problème, l'auteur discrétise le temps en pas de 5 secondes sur un horizon de temps. Ceci lui permet d'avoir une formulation linéaire du temps d'attente total, considéré comme critère. Pour prendre en compte les interactions entre les carrefours, le modèle de dispersion des pelotons de Robertson (Bretherton *et al.*, 1982), également utilisé dans les systèmes TRANSYT et SCOOT, est utilisé. Entre deux carrefours les véhicules sont typiquement regroupés en "paquets" : un peloton est un ensemble de véhicules en file l'un derrière l'autre. Modéliser la dispersion des pelotons revient à prendre en compte la répartition spatiale des véhicules sur une voie reliant deux intersections. Cette modélisation améliore la prédiction des arrivées des véhicules sur l'intersection aval.

Le critère, à minimiser, est le temps d'attente total sur le réseau, soit, dans sa formulation, la somme des files d'attente sur chaque pas de temps sur chaque tronçon.

La flexibilité est meilleure que dans certains autres systèmes ayant une vision globale car les cycles et les durées des phases sont gérées indépendamment pour chaque carrefour. Cependant, les carrefours sont tous des intersections simples (à deux phases). Cette hypothèse est une hypothèse encore plus forte que de présupposer un phasage pour un carrefour quelconque (il n'y a qu'un seul phasage possible pour les carrefours en croix). Ceci place le système dans la catégorie semi-adaptatif, du point de vue que nous nous étions fixé.

D'autres systèmes utilisant une formulation en programmation linéaire mixte, mono-objectifs et semi-adaptatifs existent (Chang *et al.*, 1986, 1988; Lin et Tsay, 1988; Kim,

1990), qui ont une vision plus “régionale” que globale, c’est-à-dire une vision réseau mais portant sur un nombre réduit de carrefours.

2.2.2 Approches multi-objectifs

Le but d’un système de régulation est de fournir une “bonne” solution lors d’un fonctionnement en temps réel. Aucune méthode de génération de l’ensemble des points non dominés n’apparaît donc dans les articles relatifs aux systèmes de régulation multi-objectifs de cette sous-section.

Étant donné le nombre de systèmes qui correspondent à cette sous-section, on fera une distinction supplémentaire, en identifiant les systèmes traitant explicitement des transports publics et ceux qui ne le font pas.

2.2.2.1 Systèmes sans considération relative aux transports publics

Système SCOOT

Le système SCOOT (Bretherton *et al.*, 1982), à *vision globale*, initialement mono-objectif, est semi-adaptatif. En effet, il récupère tout d’abord des plans de feux obtenus pour les carrefours du réseau par un autre système, le système TRANSYT, (Robertson, 1969). Puis il effectue des modifications sur les *durées de vert*, les *décalages*, et le *cycle* indépendamment sur chaque carrefour, sauf pour le cycle qui est commun. Les versions 2.4 et suivantes de SCOOT prennent en compte plusieurs objectifs, à savoir la somme des longueurs des files d’attente et la somme des nombres d’arrêts sur chaque ligne de feu (Bretherton et Browen, 1990).

Une caractéristique assez surprenante au premier abord, mais partagée par d’autres systèmes, est que le critère d’optimisation change en fonction du type de variables (durées de vert, décalages ou cycle) considéré. Le système n’utilise pas de méthode d’optimisation pour prendre ses décisions, mais procède à une série de tests, décrits ci-après, en calculant la valeur du critère courant à chaque fois puis en comparant les valeurs.

Le critère choisi pour optimiser les durées de vert est le *degré de saturation*, c’est-à-dire le rapport entre le débit moyen des véhicules et le débit maximum possible pour la traversée du carrefour en entier. Il est à maximiser. La méthode calcule le degré de saturation pour chacun des trois ajustements possibles suivants : garder la durée de

vert prévue initialement, l'augmenter de 4 secondes, ou la diminuer de 4 secondes. Le traitement est effectué avant chaque changement de phase sur chaque carrefour.

Le critère pour optimiser les décalages est appelé "*indice de performance*". Il est égal à la somme pondérée des files d'attente sur tous les tronçons et du nombre d'arrêt au niveau de chaque ligne de feu de chaque tronçon.

Le cycle est le même pour un certain nombre de groupes de carrefours du réseau, qui seront donc traités en même temps. Elle est réajustée en fonction du degré de saturation obtenu sur le carrefour le plus dense, c'est-à-dire possédant le degré de saturation maximum, et ce toutes 3 à 5 minutes. Les valeurs considérées pour effectuer l'optimisation sont des incréments ou décréments de 4 à 8 secondes par rapport à la valeur courante.

Une version plus récente prenant en compte les bus a été développée, et est décrite en section 2.2.2.2.

Système CARS

Le système CARS (Barcelo *et al.*, 1991) utilise une méthode de régulation qui s'apparente à celle de SCOOT. Il utilise la technique d'horizon glissant, ce qui lui permet, en théorie, de prendre davantage en compte les conséquences de ses décisions dans le temps. La taille de l'horizon vaut un cycle. Cependant, il ne considère pas de décalage, ni d'ajustement de cycle.

Approche "cellulaire"

Lin et Wang (2004) modélisent le problème de régulation du trafic à l'aide d'un programme linéaire mixte. Leur raisonnement est macroscopique, et ils travaillent sur un réseau (vision globale). Les carrefours ont un fonctionnement cyclique et il s'agit de déterminer, pour chaque intersection, la *durée des phases*, et le *cycle*. Le phasage est déterminé à l'avance, ce qui constitue un manque de flexibilité à noter.

Le temps et l'espace sont discrétisés. Pour la partie espace, il est discrétisé en différentes régions appelées cellules. Deux critères sont considérés : le temps d'attente cumulé et le nombre d'arrêts, tous deux à minimiser.

L'agrégation des critères s'effectue par somme pondérée, sans normalisation préalable. La méthode de résolution fait appel à un solveur de PLNE (CPLEX, LINDO, etc.). Le nombre de variables 0-1 nécessaires pour la modélisation, soit la longueur de

l'horizon multiplié par le nombre d'intersections, laisse sceptique quant à l'optimisation en temps réel. Les auteurs ne donnent d'ailleurs aucun résultat expérimental en terme de temps de résolution.

Le choix de discrétiser l'espace et d'utiliser la méthode sur un réseau entier semble rendre peu réaliste la résolution par un algorithme exact. De plus, bien que le nombre total de variables soit, selon les auteurs, de l'ordre du nombre de pas de temps sur l'horizon choisi multiplié par le nombre d'intersections, il est fait abstraction des carrefours plus "conséquents". On entend par carrefour conséquent un carrefour qui comporte plus de quatre tronçons et où la régulation ne peut se faire qu'en considérant plus de deux phases de trafic possibles. La complexité du problème (nombre de variables) se mesure alors par rapport au nombre de carrefours multiplié par le nombre de tronçons par carrefour, au lieu du nombre d'intersections (simples) sur un réseau.

Approche cellulaire robuste

Une approche "robuste" de la vision "*cellulaire*" est envisagée par Ukkusuri (2010). Les phases sont communes à toutes les intersection (simples).

Trois critères sont optimisés. Le premier est appelé le "temps total de traversée du réseau". Il s'agit de la somme des temps d'attente sur l'ensemble des cellules du réseau. Le second s'intitule "temps de traversée des intersections". Il s'agit de la somme des temps d'attente sur l'ensemble des cellules proches des intersections. Enfin le troisième est le nombre d'arrêts, mais sur les cellules situées à proximité des intersections uniquement.

Une des différences majeures avec l'article précédent est la "robustification" obtenue. La fluctuation des demandes au cours du temps est prise en compte dans le calcul des critères, en supposant qu'elle suit une certaine loi de probabilité. La nouvelle fonction objectif est une combinaison linéaire des espérances mathématiques des critères, et des écarts types correspondants.

A noter qu'il est également possible, si on le souhaite, de rendre un programme linéaire robuste sans faire d'hypothèses sur la distribution en probabilité, portant par exemple sur l'arrivée des véhicules sur un tronçon (Bertsimas et Sim, 2004). Cet article est d'ailleurs connu de Ukkusuri (2010), qui n'a pas opté pour cette technique.

Méthode COP

Head et Sen (1997) ont développé une méthode de régulation temps réel, appelée COP (Controlled Optimization of Phases), pour la régulation du trafic sur un carrefour isolé, qui prend en compte plusieurs objectifs : la longueur maximale des files d'attente, le temps d'attente total et le nombre d'arrêts des véhicules particuliers (seuls les véhicules particuliers sont modélisés).

La méthode d'optimisation utilisée est la programmation dynamique. Le temps est discrétisé mais une étape dans la méthode correspond à une phase et non à un pas de temps. Un horizon de temps est également pris en compte.

Sans être complètement microscopique, la modélisation choisie attribue tout de même, à chaque véhicule k , une estimation a_k de la date d'arrivée prévue. Les arrivées des véhicules sont alors connues à chaque pas de temps, ce qui permet de mettre à jour les files d'attentes. Les dates a_k rentrent explicitement dans le calcul du critère temps d'attente, permettant de prendre en compte implicitement la répartition des véhicules sur le tronçon, ce qui est plus difficile dans une modélisation macroscopique stricte.

Le principe est de se fixer initialement une séquence de phases fixe répétée un certain nombre de fois. L'optimisation ne porte que sur la durée des phases, pendant l'horizon de temps. Les séquences de phases peuvent être modifiées, puisque qu'une durée nulle peut être attribuée à n'importe quelle phase, ce qui revient à la supprimer de la séquence initiale. Ceci confère au système une flexibilité plus élevée que les systèmes n'agissant que sur la durée bornée de phases appartenant à un phasage fixé, dont l'ordre ne peut pas changer. Cependant, selon les auteurs, les solutions peuvent ainsi contenir toutes les séquences de phases possibles, sans autre précision. Ceci est, selon nous, assez discutable, surtout si le nombre de phases possibles est grand, même si le nombre d'étapes (ou de phases traitées) est limité.

En effet, pour pouvoir générer toutes les séquences possibles de longueur p , avec n phases possibles ($n \geq p$), il faut prendre une séquence initiale fixe assez conséquente. Si on prend, par exemple, comme les auteurs le suggèrent, une séquence initiale qui répète toutes les phases possibles, il faut qu'elle ait une longueur d'exactly $p \times n$ phases. A titre d'exemple, si $p = 4$ et $n = 6$ (six phases possibles A,B,C,D,E et F), on a une séquence initiale qui ressemblera à (A, B, C, D, E, F, A, B, C, D, E, F, A, B, C, D, E, F, A, B, C, D, E, F) qui permet effectivement d'obtenir n'importe

quel ordre de 4 phases parmi 6. Mais on aura alors 24 étapes dans la programmation dynamique. On peut éventuellement diminuer ce nombre d'étapes (considération non présentée dans l'article) en considérant, par exemple, la séquence initiale suivante : (A, B, C, D, E, F, A, B, C, D, E, A, B, C, D, A, B, C). La longueur correspondante est de $n + (n - 1) + \dots + (n - p) = pn - \frac{p(p-1)}{2} = O\left(p\left(n - \frac{p}{2}\right)\right)$ ce qui représente tout de même 18 étapes dans cet exemple. Si p peut être volontairement pris faible, correspondant à un horizon de temps faible, n peut être assez grand sur des carrefours complexes. Sur le carrefour pris comme exemple dans l'article, qui n'est pas spécialement complexe, il vaut déjà 9 (mais les auteurs se restreignent volontairement à 3 phases possibles).

Pour les expérimentations présentées, l'horizon de temps est de 30 secondes, le temps total alloué à la séquence des phases étant 10 pas de temps, d'où un pas de temps de 3 secondes. Avec ces hypothèses, ils obtiennent de bons temps de résolution.

En ce qui concerne les critères considérés, leur formulation est classique :

- La longueur de file maximale s'obtient facilement puisque les longueurs de files sont des variables du modèle.
- Le nombre d'arrêts est considéré comme suit. Pour les feux au rouge pendant la phase considérée, le nombre d'arrêts correspond au nombre d'arrivées des véhicules. Pour les feux au vert, le nombre d'arrêts vaut le nombre de véhicules initialement présents dans la file plus les véhicules arrivés pendant la phase, qui ne peuvent pas passer pendant la phase.
- Le temps d'attente est considéré comme suit. Pour les feux au rouge pendant la phase, il correspond à la longueur de file courante multipliée par la durée de la phase plus le temps que devra attendre chaque véhicule supplémentaire arrivé. Pour les feux au vert, la même considération est faite pour les véhicules qui ne pourront passer. Quant aux véhicules pouvant passer pendant la phase, un terme supplémentaire leur est associé.

Mais il faut noter aussi que la programmation dynamique, qui permet des formulations non-linéaires, autorise une finesse dans la modélisation plus difficilement atteignable par une formulation linéaire. Les critères par exemple, tiennent compte, dans ce système, de fonctions empiriques non forcément linéaires, qui sont :

- le nombre *maximum* de véhicules pouvant passer à un feu mis au vert, en fonction du temps de vert et du type de phases.
- le délai d'attente de véhicules étant autorisés à passer, pendant une phase, en

fonction du nombre de véhicules et du type de phases.

Ces fonctions sont, en général, approximées par des fonctions linéaires dans les autres systèmes.

Les auteurs ne font pas mention d'une agrégation précise des critères, mais laissent l'"utilisateur" choisir entre des utilisations mono-objectif et une "combinaison" des critères. De plus, une discussion est réservée à l'utilisation des critères en fonction de la situation de trafic, avec la réflexion suivante, basée vraisemblablement sur des éléments obtenus auprès d'un ingénieur du trafic américain (Arizona) : lorsque le carrefour n'est pas saturé, le nombre d'arrêts est privilégié, tandis que lorsque le carrefour "commence à être saturé", c'est plutôt le temps d'attente qui semble pertinent comme critère. Le "degré de saturation" d'un carrefour peut se mesurer en calculant, par exemple, le rapport entre le débit de véhicules arrivant sur le carrefour et la "capacité du carrefour", qui peut être vue comme un débit seuil des arrivées. D'après leur source, un "changement de critère" est souhaitable à partir d'une saturation d'environ 0.8.

En conclusion, la qualité notable de ce système, qui utilise une méthode exacte pour l'optimisation, est qu'il possède une bonne finesse dans la modélisation. Mais bien que le système ait une bonne perception et soit plus flexible que les systèmes classiques de régulation à base de phasage fixe, il n'est pas suffisamment flexible pour être qualifié de système adaptatif, selon nos critères, à cause de la limitation du nombre de phasages possibles utilisables en temps réel.

Système MOVA

Le système MOVA (Microprocessor Optimized Vehicule Actuation), de Vincent et Peirce (1988) est un système semi-adaptatif conçu pour l'utilisation sur des carrefours isolés. Il s'agit, chaque demi-seconde, de déterminer la durée de la phase en cours.

Deux particularités du système nous semblent originales. Premièrement, le système n'utilise pas les informations classiques sur le trafic, portant sur la quantité de véhicules arrivant sur les tronçons d'arrivée, mais considère les distances inter-véhiculaires (entre les véhicules successifs). Deuxièmement, les trois critères considérés, le temps d'attente, le nombre d'arrêts et la capacité, sont appliqués ou non en fonction de la situation du trafic. Ainsi, tant que le carrefour n'est pas saturé, seuls le temps d'attente et le nombre d'arrêts, agrégés sous forme de somme pondérée, sont appliqués (à minimiser), et lorsque le carrefour est saturé, seule la capacité (à maximiser) est appliquée.

2.2.2.2 Systèmes prenant en compte les transports publics

Prise en compte des transports en commun dans SCOOT

Récemment, SCOOT a été amélioré pour incorporer un système de priorité accordée aux bus (version SCOOT-MC3 de Bretherton (2007)). Cependant le critère global à optimiser est toujours le même, la priorité aux bus étant gérée indépendamment, par prolongement du vert sur la ligne concernée au moment où le bus arrive sur le carrefour, ou bien par restriction du temps de vert des lignes antagonistes.

Système UTOPIA

Le système UTOPIA (Mauro et di Taranto, 1990) a pour but de minimiser le temps total perdu par les véhicules particuliers lors de leur passage dans le réseau, avec des contraintes portant sur la priorité accordée aux transports en commun. Il comprend deux niveaux : un niveau “régional”, à l’échelle du réseau, et un niveau “local”, à l’échelle d’un carrefour. Le contrôleur régional déclenche un processus d’optimisation (heuristique) à chaque fois qu’un changement dans les conditions de trafic est détecté, et qui consiste à établir des plans de feux de référence aux contrôleurs locaux afin de minimiser le temps perdu par les véhicules particuliers. Les contrôleurs locaux effectuent ensuite des ajustements en fonction du trafic et des informations échangées avec les carrefours voisins.

En ce qui concerne la gestion des transports en commun, à chaque fois qu’un transport en commun est détecté par un contrôleur local, un message est envoyé au contrôleur régional qui modifie les plans de feux de tous les carrefours afin de lui attribuer une priorité.

Système MOTION

Dans le système MOTION (Bielefeldt et Busch, 1994) la priorité accordée aux bus est variable. Il s’agit d’un système à *vision globale*, utilisant une heuristique pour la détermination des variables de régulation (*cycle commun, répartitions des verts, décalages*) qui consiste dans un premier temps en la succession de phases d’“optimisation par niveau” :

- Une première optimisation est faite au “niveau local”, c’est-à-dire pour chaque carrefour indépendamment les uns des autres. L’optimisation consiste en la dé-

termination des durées de vert et d'un cycle minimal de fonctionnement. Les solutions sont transmises à un "niveau global".

- Une seconde optimisation s'effectue au "niveau global", c'est-à-dire au niveau du réseau, qui vise à calculer un cycle optimal, commun à tous les carrefours.
- A partir de cette durée, l'optimisation au niveau local est ré-effectuée.
- Enfin, les décalages sont calculés.

Cette méthode est appliquée toutes les 10 à 15 minutes.

De plus, une fois par cycle, le système peut adapter au niveau local la modification de la phase courante sous les contraintes fournies par le niveau global. On parle de micro-régulation.

L'optimisation au niveau du réseau passe par la construction d'"ondes vertes". Il s'agit d'une heuristique qui consiste à mettre successivement au vert des feux sur un même chemin, dans le réseau, suivant une direction déterminée. La micro-régulation et les décalages sont constitués en prenant en compte un indicateur de performance, qui indique les retards et les nombres d'arrêts courants sur les voies du réseau.

A ceci se rajoute la prise en compte des bus au niveau global. L'aspect intéressant est la présence de deux types de priorités. Une priorité *absolue* et une priorité *relative*. Si la priorité absolue est accordée aux bus, alors les optimisations au niveau global iront dans le sens de la minimisation des retards des bus uniquement. Tandis que si la priorité accordée est conditionnée par les retards subis par les autres véhicules, il s'agira de priorité relative.

Systeme RHODES

Le système RHODES (Real-time, Hierarchical, Optimized, Distributed, and Effective System, (Head et Mirchandani, 2001)), à *vision globale*, utilise la méthode de régulation temps réel COP (Controlled Optimization of Phases), déjà mentionnée en 2.2.2.1, pour l'aspect local de sa gestion. Pour chaque carrefour, des recommandations sont fournies par une gestion au niveau réseau, grâce à l'utilisation du système REALBAND (Dell'Olmo et Mirchandani, 1995). Ces recommandations portent directement sur la durée des phases du COP.

REALBAND détecte les pelotons sur le réseau. Il considère qu'il y a un "conflit" si deux pelotons arrivent au niveau de feux antagonistes. Un arbre de décision est généré, dans lequel un nœud correspond à un conflit, et une branche à une décision possible.

Les décisions portent sur le ou les pelotons à arrêter parmi tous ceux impliqués dans le conflit correspondant.

De plus, RHODES a intégré, dans le même temps, une approche pour prendre en compte les bus (Mirchandani *et al.*, 2001). La prise en compte est relativement simple : comme la méthode utilisée au niveau local prend les véhicules en compte individuellement pour le critère temps d'attente, c'est à dire qu'une somme sur les véhicules est nécessaire dans son calcul, les termes correspondant à des bus sont multipliés par un poids relatif à la priorité accordée au bus. Ce poids vaudra $n \times 1$, où n est le nombre de passagers, si le véhicule est à l'heure ou en avance (on n'"aide" donc jamais le bus à prendre du retard), et il vaudra $n \times (1 + K \times d)$, où K est une constante et $d > 0$ le retard qu'a pris le bus dans le parcours précédant son arrivée. Prendre en compte le nombre de passagers et le retard de chaque bus en temps réel suppose bien sûr qu'un mécanisme de transmission de ces informations existe.

Système SPPORT

Un système connu sous le nom de SPPORT (Signal Priority Procedure for Optimization in Real-Time), développé au Canada (Han et Yagar, 1994), puis repris et amélioré par divers instituts canadiens et américains (Conrad *et al.*, 1998; Dion et Hellinga, 2002), s'intéressent particulièrement aux interactions qu'il peut y avoir entre les véhicules de transport en commun et le reste du trafic, sur des intersections "simples" (à deux phases).

Il s'agit d'un système temps réel à base de règles. Comme il n'y a que deux phases possibles, la décision va porter sur la prochaine commutation (choisie parmi 3 dates possibles au maximum) dans l'horizon de temps de 30 à 90 secondes pour les expérimentations. En faisant glisser l'horizon de temps, le futur plan de feu complet est ainsi généré. Tous les véhicules, VP compris, sont considérés indépendamment. La modélisation est donc entièrement microscopique.

Citons quelques-unes des règles, fixées à l'avance, mais dont les priorités sont laissées à l'"utilisateur" du système :

- Si une file de n véhicules existe et n'est pas en train de s'écouler, commencer à l'écouler le plus tôt possible.
- Commuter l'état courant au vert si la file excède une certaine longueur seuil, définie par l'utilisateur.

- Maintenir le vert courant si la fin de file dépasse un certain emplacement seuil, défini par l'utilisateur.
- Si un peloton de n véhicules ou plus approche de l'intersection, commuter au vert à un instant qui permet au peloton de ne pas être impacté par d'éventuels véhicules stoppés.
- Si un véhicule de transport en commun s'approche d'une ligne de feu, commuter au vert pour permettre au véhicule de passer l'intersection sans avoir à s'arrêter.

L'aspect multi-mode, et par conséquent multi-objectif, est déjà pris en compte à ce niveau, puisque les règles portent aussi bien sur le trafic général que sur les véhicules de transport en commun. De plus, cet aspect multicritère intervient dans l'indice de performance, combinaison linéaire du délai d'attente et du nombre d'arrêts de chaque véhicule (particulier ou de transport en commun). L'indice de performance, qui est donc une agrégation des critères, prend en compte l'"importance" de chaque *type* de véhicules, sous forme d'un poids, qui pourra être plus élevé pour des véhicules de transport en commun par exemple. Mais ce poids peut aussi traduire, selon les auteurs, l'occupation spatiale d'un véhicule.

Le système fonctionne comme suit. Plusieurs listes de priorités, portant sur les règles, fournies hors ligne par l'utilisateur, sont stockées. Lors de chaque optimisation, le système construit plusieurs plans de feux, générés à partir des règles. Chaque plan de feux correspond à une liste de priorités. Pour choisir le "meilleur" plan de feux, le système considère l'indice de performance.

Régulation multi-objectif par essaim particulaire

Une approche heuristique intéressante *via* un algorithme d'essaim particulaire a récemment été proposée par Kachroudi et Bhouri (2009).

L'aspect multi-mode occupe une place importante dans le système : VP et TC sont pleinement pris en compte. L'optimisation se fait sur un réseau avec un cycle fixe et commun à tous les carrefours. *Les phases sont fixes et connues.*

Les variables de décision sont les répartitions des *temps de vert*, sur chaque carrefour et les *décalages* entre carrefour. Les deux critères considérés sont :

- le nombre de véhicules particuliers présents dans le réseau
- la somme des écarts quadratiques entre la position de chaque bus et une position idéale

La méthode d'optimisation utilisée est une méta-heuristique appelée "optimisation par essaim particulaire". Il s'agit d'une méthode d'optimisation stochastique, pour les fonctions non-linéaires, basée sur la reproduction d'un comportement social (Eberhart *et al.*, 2001).

La version multicritère est la suivante. Dans le cas où l'on doit choisir entre deux solutions pendant l'"évolution", plusieurs cas se posent :

1. Une des deux solutions domine l'autre, et dans ce cas, il n'y a pas de problème, a priori, pour choisir
2. Il n'y a pas de dominance. On fait alors un choix suivant une politique de décision :
 - soit on a à disposition une fonction d'agrégation
 - soit on choisit aléatoirement
 - soit on donne systématiquement la priorité à un critère

Régulation multi-objectif basée sur un algorithme génétique

Une étude proposée par Schöcker *et al.* (2008) a été réalisée en 2008 sur un carrefour isolé de Londres, mettant en jeu un algorithme génétique ayant pour but l'optimisation de plusieurs critères décrits ci-après.

Dans cette approche, le phasage et le cycle sont fixes. Seule la *durée des verts* pour chaque phase est recherchée, en temps réel. L'optimisation en temps réel repose sur des données, supposées disponibles en permanence, de longueurs de files des véhicules, particuliers ou non, sur l'axe principal du carrefour, et de temps moyens d'attente des piétons sur tout le carrefour.

Le principe est de pouvoir modifier la durée des phases, comme dans le cas de SCOOT, qui est pris comme référence pour comparaison, à ceci près que le cycle est fixé (il s'agit de la version SCOOT-FT, où FT signifie Fixed-Time).

Les différents objectifs à minimiser sont : le temps d'attente moyen des véhicules particuliers, le temps d'attente moyen des bus, le temps d'attente moyen des piétons, différentes files d'attentes à des endroits jugés stratégiques du carrefour.

La modélisation recourt à la logique floue et la méthode de résolution s'appuie sur un algorithme génétique. Pour la partie logique floue, retenons que dans ce modèle, il est impératif de pouvoir transformer les informations entrantes, comme certaines longueurs de files d'attente et temps d'attente moyen des piétons, en différents états qualitatifs

(court, moyen ou long par exemple). En effet, il est fait usage d'une base de *règles* établie par un expert. Les états jugés optimaux sont transformés en nombres, qui sont les durées des phases.

La fonction d'agrégation choisie, à maximiser, est du type :

$$\min_{i \in \{1, \dots, p\}} \{C_1(y_1(x)), \dots, C_p(y_p(x))\}$$

où X est l'espace de décision, et chaque C_i , $i \in \{1, \dots, p\}$ est un degré d'acceptabilité du critère y_i .

Chaque degré d'acceptabilité est défini grâce à des seuils, dont les valeurs dépendent de la politique de régulation choisie. Le fait d'utiliser une agrégation différente de la somme pondérée montre une vraie préoccupation des auteurs sur l'aspect multi-objectif de leur système. Cependant la fonction d'agrégation reste assez simpliste. En effet, il s'agit d'une agrégation *a priori*, c'est-à-dire que les paramètres qui la composent, comme par exemple les seuils qui déterminent le degré d'acceptabilité des critères, ne varient jamais. On cherche toujours le même type de solution quelle que soit la situation du trafic. De plus, l'hypothèse que certaines zones de l'espace des critères ne contiennent jamais de solution intéressante est implicitement utilisée, puisque seules les solutions dont les critères sont compris entre deux seuils d'acceptabilité sont prises en compte. Faire une telle hypothèse est très discutable, puisque qu'il se pourrait que, dans certains cas, on veuille justement accéder à des solutions "asymétriques" (jugées très bonnes sur un critère et sans exigence particulière sur d'autres).

2.3 Systèmes adaptatifs

2.3.1 Approches mono-objectifs

Système OPAC

Le système OPAC (Gartner, 1983) est l'un des premiers systèmes utilisant une approche acyclique. Conçu en 1982 aux États-Unis, il considère chaque feu indépendamment, contrairement à l'approche par phases, en lui attribuant une variable d'état (0 pour rouge et 1 pour vert), à chaque pas de temps considéré.

Le critère utilisé est le temps total d'attente sur le carrefour, à minimiser. Il est calculé sur un horizon de temps. La méthode utilisée est la programmation dynamique.

Les arrivées sont considérées comme constantes et les départs dépendent bien entendu des commutations. Comme on raisonne à pas de temps fixe, le temps d'attente total sur l'horizon de temps est proportionnel à la longueur des files sur cet horizon. La principale difficulté des approches par programmation dynamique étant le temps de calcul, d'autant plus prohibitif que l'horizon de temps est grand, les auteurs proposent également une version avec un horizon glissant (Boillot et Braban, 2003), un compromis classique entre "l'exactitude" de l'optimisation et le temps de calcul.

Système PRODYN

Le système PRODYN (Farges *et al.*, 1983) dont la conception remonte aux années 80, a été développé pour minimiser, le temps total perdu par les véhicules particuliers, initialement uniquement sur un carrefour isolé, sur un horizon de temps de 75 secondes découpé en unités de temps de 5 secondes chacune. Il a rapidement été adapté pour fonctionner sur un réseau. Cependant le système n'effectue pas d'optimisation globale, mais organise une coordination globale par échange d'informations, entre les carrefours. Au niveau d'un carrefour (considéré isolément) une optimisation rigoureuse est cependant effectuée, à l'aide de la programmation dynamique. Il existe un certain nombre de versions de ce système, commercialisé depuis un certain nombre d'années. Il existe ainsi une version qui prend en compte les transports en commun (bus), sans pour autant que cela représente un deuxième critère (Farges et Henry, 1994). Lorsqu'un bus entre en approche d'un carrefour, il communique sa position au système (connue grâce à l'utilisation supposée d'un odomètre couplé à un GPS). Le système estime la date d'arrivée du bus à la fin de la file courante sur le tronçon concerné, ainsi que la longueur de la file. En fonction de ces deux éléments et du poids relatif au bus, qui correspond à un "niveau de priorité" accordé au bus, il va adapter la durée du signal du feu correspondant.

Approche par programmation dynamique approchée

Pour pallier au bien connu "fléau de la dimension", évoqué par Bellman (1957) concernant la programmation dynamique, un certain nombre de stratégies heuristiques se sont

développées. Parmi elles, figure la *programmation dynamique approchée* (Bellman et Dreyfus, 1959).

Cai *et al.* (2009) proposent d'utiliser la programmation dynamique approchée pour résoudre le problème de régulation aux carrefours à feux. Le principe (heuristique) est de remplacer, dans le calcul récursif classique de la fonction d'évaluation J , la valeur de la fonction d'évaluation sur les états *futurs* par une approximation. La fonction d'évaluation approximée dépend de l'état du système et d'un *paramètre* $r \in \mathbb{R}^n$ et ne nécessite pas de calcul récursif. Ceci permet de diminuer significativement les temps de calcul.

Le paramètre r , sert à améliorer l'approximation faite sur J . La détermination de la fonction h qui permet de faire évoluer le paramètre r constitue le cœur de l'heuristique. La méthode utilisée est l'apprentissage par renforcement (Barto *et al.*, 1983; Barto et Sutton, 1998), qui consiste en l'"apprentissage" du système (appelé "agent") par interaction avec son environnement, dans lequel l'agent a un but précis, malgré la présence d'aléas. Pour ce faire il faut quatre éléments :

- une politique, qui détermine le comportement et la performance de l'agent
- une fonction de récompense ou de coût, qui détermine les caractéristiques du problème rencontré par l'agent
- une fonction d'évaluation, ici la fonction J , qui estime la récompense ou le coût à long terme
- un modèle d'environnement

L'état du système correspond aux longueurs des files d'attente, avec un vecteur aléa, les variables de décisions sont les états des feux.

Le critère choisi est le temps total d'attente des véhicules particuliers (fonction J). La perception du système est assez bonne, les informations étant mises à jour toutes les demi-secondes, correspondant dans le même temps à la discrétisation temporelle choisie. Un horizon glissant est utilisé, de taille temporelle de 10 à 20 secondes. Les arrivées sont estimées par la méthode de Monte-Carlo. Le système dispose d'une très bonne flexibilité, puisqu'aucune hypothèse n'est faite restreignant les solutions. La méthode d'optimisation n'est pas un algorithme d'approximation à garantie de performance, mais bien une *heuristique*. Cependant, les résultats montrent que, pour certains scénarios de trafic, les gains en terme de temps d'attente, par rapport à un système à plan de feux fixe (TRANSYT), peuvent être intéressants.

Système CRONOS

Le système CRONOS (Boillot *et al.*, 2006) est un système de régulation original développé initialement dans les années 90 à l'INRETS. Le critère à minimiser est le temps d'attente total au carrefour sur un horizon de temps, de l'ordre de la minute, en utilisant l'"algorithme de Box". Cet algorithme est une heuristique qui travaille dans l'espace des solutions avec des considérations géométriques. Le fait d'utiliser cet algorithme lui permet d'être très rapide à l'exécution, donc adapté à l'utilisation temps réel, et lui confère une originalité du point de vue de la méthode d'optimisation. Cette rapidité lui permet également de procéder à une optimisation globale sur plusieurs carrefours (vision zonale), tout en fournissant des solutions en moins d'une seconde. Une version multi-objectif est en cours d'élaboration.

Système BONSAÏ

Enfin, le système BONSAÏ (Boumghar, 2000), développé à l'INRETS en 2000, utilise la programmation linéaire mixte en nombres entiers pour minimiser le temps d'attente totale sur un carrefour isolé, sur un horizon de temps. Le temps est discrétisé, et les variables principales utilisées sont des variables 0-1 doublement indicées qui représentent l'état de chaque feu à chaque instant. La matrice des antagonismes est utilisée directement comme matrice des contraintes sur les antagonismes, auxquelles se rajoutent d'autres contraintes liées aux limites de temps de vert et de rouge.

2.3.2 Approches multi-objectifs

A notre connaissance, aucun système pleinement adaptatif utilisant plusieurs critères n'apparaît dans la littérature.

2.4 Conclusion du chapitre

Les approches récentes, en particulier les approches temps réel, non seulement utilisent des méthodes d'optimisation variées, incluant pour certaines les dernières avancées dans ce domaine, mais intègrent également, de plus en plus, plusieurs critères. Néanmoins cet aspect n'est pratiquement jamais au cœur de la démarche. Cela s'explique

par le fait que les systèmes ont d'abord été conçus pour résoudre des problèmes de régulation liés aux véhicules particuliers, centrés en particulier sur le temps d'attente.

Le critère le plus présent dans les systèmes de régulation actuels est donc naturellement le temps d'attente (à minimiser). On trouve également le nombre d'arrêts dans les systèmes temps réel (semi-adaptatifs ou adaptatifs). En revanche, en ce qui concerne les systèmes à plan de feux fixe, le critère le plus fréquemment utilisé est la capacité du carrefour (à maximiser).

Pour le cas particulier des transports en commun, la grande majorité des systèmes actuels n'ont pas intégré de critère précis relatif aux bus. Il s'agit, pour la plupart, d'accorder ponctuellement des priorités de passage, en modifiant directement l'état des feux, indépendamment de tout indicateur de situation de trafic, sans recourir à un critère.

L'agrégation des critères est souvent choisie *a priori*, et correspond le plus souvent à une somme pondérée, ce qui est très différent de l'approche choisie dans la thèse (approche interactive). De plus, même pour ce type d'agrégation simple, portant sur des critères qui sont, en général, incommensurables, il arrive fréquemment qu'il n'y ait pas de distinction faite entre l'importance accordée à chaque critère, et les coefficients permettant de normaliser les échelles de valeur que peuvent prendre les différents critères. De manière générale, les auteurs qui recourent à plusieurs critères ne développent que très peu la façon de les prendre en compte.

Chapitre 3

Modélisation du trafic, présentation du système et des modèles, évolution des paramètres préférentiels

Trois modélisations différentes ont été formulées pour répondre à la problématique de la thèse. Tous les modèles sont des programmes linéaires mixtes en nombres entiers, et s'intègrent dans un même système développé pour la thèse.

- le modèle dit “par états”, où les variables principales sont les états des feux
- le modèle dit “par dates”, où les variables principales sont les dates de commutation des feux
- le modèle appelé “hybride”, qui allie les deux types de modélisation précédents

De plus, le système développé est construit de manière à autoriser l'utilisation de méthodes interactives, dont le fonctionnement sera évoqué dans ce chapitre.

Le chapitre est divisé comme suit :

- Dans la section 3.1, on trouvera la description détaillée des choix de modélisation du trafic aux carrefours à feux.
- La section 3.2 comporte la présentation générale du système et les notations communes aux trois modèles.

- Les sections 3.3, 3.4, 3.5 et 3.6 sont consacrées à la description des modèles.
- Enfin, dans la section 3.7, deux types méthodes interactives sont proposées, toutes deux utilisables au sein du système.

3.1 Modélisation du trafic sur un carrefour

Dans cette section nous présentons les hypothèses de travail et les choix faits concernant la modélisation du trafic. On note H la durée de l’horizon de temps futur sur lequel on va travailler. L’intervalle de temps $[0, \dots, H]$ contient donc toutes les dates entre le moment présent $t = 0$ et la date $t = H$.

3.1.1 Hypothèses sur le carrefour et sur les données disponibles en temps réel

- On travaille sur un carrefour isolé. Cette hypothèse se traduit principalement par le fait qu’on ne considérera pas les arrivées des véhicules particuliers sur le carrefour comme des variables des programmes linéaires mixtes utilisés, mais comme des quantités qu’on fera évoluer dans le temps, en fonction des mesures.
- Des capteurs, installés sur le carrefour, nous fournissent en temps réel la longueur des files d’attente. Une file d’attente est constituée des véhicules à l’arrêt sur le tronçon concerné. Plus concrètement, le type de capteur particulièrement propice à cela est le capteur de type caméra vidéo, qui permet en particulier de fournir en temps réel des longueurs de files d’attente. On considérera alors que l’entrée d’un tronçon correspond à l’endroit à partir duquel l’ensemble des caméras installées sur le carrefour peuvent fournir l’information relative à la présence ou non de véhicules à l’arrêt.
- On connaît à chaque instant la position et la vitesse des bus susceptibles de traverser le carrefour dans l’horizon de temps, grâce à l’utilisation d’un système de géolocalisation.

3.1.2 Description du carrefour

Nous décrivons ici les différentes informations sur le carrefour dont nous avons besoin et les approximations associées que nous faisons.

3.1.2.1 Structure et antagonismes

La structure du carrefour peut être résumée par le graphe des antagonismes. Lorsque des groupes de trois feux ou plus sont mutuellement antagonistes, plusieurs ordres de commutation au vert sont possibles. La méthode d’optimisation permettra, comme nous le verrons, de prendre en compte implicitement tous les ordres possibles. Ceci est un aspect important du système, car il offre le maximum de flexibilité possible quant à la recherche de solutions réalisables. En effet, contrairement à beaucoup de systèmes, passés en revue au chapitre 2, notre méthode permet de ne pas se référer à un plan de feux ou bibliothèque de plans de feux, ce qui supprimerait de nombreuses solutions possibles et, potentiellement, les solutions les plus intéressantes pour nous.

3.1.2.2 État des feux

Comme la durée du jaune est une constante, on utilisera seulement deux états possibles, “rouge” et “vert”. La durée du jaune sera ensuite ajoutée à celle du vert.

3.1.3 La modélisation des usagers

Puisque le système traitera différentes catégories d’usagers, à savoir les transports en commun et véhicules particuliers, il est naturel de les prendre en compte de manière différente.

Dans la plupart des modèles de régulation du trafic, les véhicules particuliers sont modélisés par des variables macroscopiques, c’est-à-dire considérant des débits de véhicules plutôt que les véhicules individuellement. C’est également notre choix. Les avantages d’une modélisation macroscopique sont principalement, dans notre contexte :

- Une adéquation entre les informations fournies par les capteurs et la modélisation.
- Une simplification des modèles, qui peut réduire les temps de calcul par rapport à une modélisation microscopique.

En revanche, une modélisation macroscopique est, en général, plus approximative qu'une modélisation microscopique. Une répartition inégale des véhicules sur un tronçon, par exemple, est plus délicate à prendre en compte dans une modélisation macroscopique.

En gardant ces considérations à l'esprit, la modélisation microscopique semble pertinente pour modéliser les bus dans le trafic :

- Il est techniquement possible d'avoir la position et la vitesse en temps réel grâce aux systèmes récents de géolocalisation et de transmission.
- Le nombre de bus est en général très faible par rapport au nombre de véhicules particuliers. Ceci permet de penser que les temps de calcul ne seront pas trop affectés par la prise en compte de chaque bus, individuellement.
- On a besoin de pouvoir différencier les bus, si on veut prendre en compte, en particulier, les cas où plusieurs bus sont en approche du carrefour sur des tronçons différents pendant l'horizon de temps.

3.1.4 Modélisation des files d'attente (VP)

3.1.4.1 Calcul des longueurs de file, représentation des files

Pour un tronçon donné, on appelle longueur de file le nombre de véhicules qui y sont à l'arrêt. Le calcul de la longueur des files est utile pour les trois critères impliqués.

On considère ici un tronçon donné. Pour chaque instant t , on note $QA(t)$ le débit des arrivées des véhicules, en véhicules par seconde, et $QS(t)$ le débit des sorties, sur le tronçon.

On fait ici plusieurs approximations notables :

- Pendant un petit intervalle de temps $\Delta t \geq 0$, on considère que QA ne varie pas, c'est-à-dire qu'on a, pour chaque $t' \in [t, \Delta t[$:

$$QA(t') = QA(t)$$

- On néglige le temps de démarrage du véhicule en début de vert, c'est-à-dire que l'on considère que le débit de sortie vaut toujours le débit de saturation QS si le feu est au vert, et 0 sinon.

Si $l(t)$ est la longueur de la file, en nombre de véhicules, à l'instant $t \in [0, \dots, H]$ et $\Delta t \geq 0$ tel que l'état du feu soit le même à $t + \Delta t$, on a :

$$l(t + \Delta t) = l(t) + QA(t) \times \Delta t$$

durant le rouge. Et on a :

$$l(t + \Delta t) = \max \{l(t) + (QA(t) - QS) \times \Delta t, 0\}$$

durant le vert.

Cette façon de procéder est classique, elle est à la base de la plupart des modélisations macroscopiques, avec débits constants ou non, pour la régulation sur un carrefour.

3.1.4.2 Extension spatiale des files en considérant le procédé de calcul précédent acquis

Un des critères considérés, le nombre d'arrêts, dépend de l'extension spatiale des files, d'où l'importance d'une modélisation au niveau spatial. Une illustration de la position d'une file sur son tronçon est donnée sur la figure 3.1 (bloc vert), juste après le passage au vert du feu correspondant, puis après T unités de temps, en supposant qu'on soit toujours au vert.

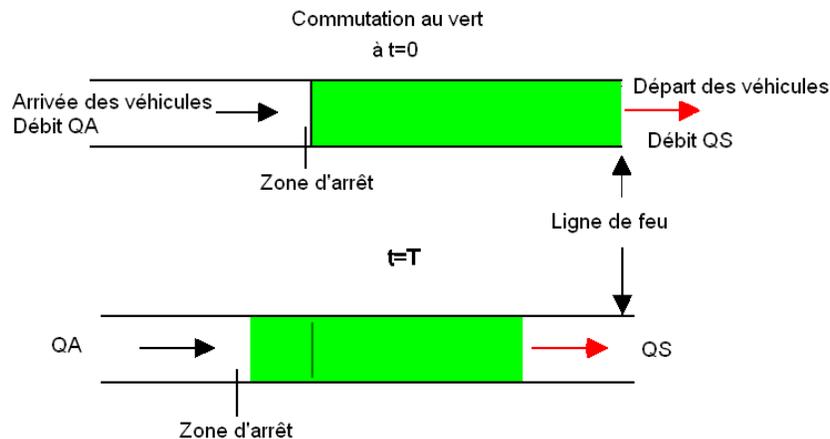


FIGURE 3.1 – Modélisation spatiale d'une file pendant un vert

Tous les véhicules qui arrivent sont alors arrêtés par la fin de la file courante (“Zone d’arrêt” sur la figure 3.1). Avec cette seule représentation, il est alors facile d’estimer le nombre d’arrêts, à un instant donné t , en fonction de la longueur de la file courante $l(t)$, lorsque le feu est au vert :

- si $l(t) = 0$, alors le nombre d’arrêts sur le tronçon est nul
- si $l(t) > 0$, alors le nombre d’arrêts vaut $QA(t)$

Lorsque le feu passe au rouge, on pourrait considérer simplement que le nombre d’arrêts vaut le nombre de véhicules qui arrivent pendant le rouge.

Cependant, si la commutation au rouge intervient avant que la file ne se soit écoulée entièrement, on peut éventuellement considérer que des *arrêts additionnels* apparaissent, à cause du fait que lors d’un vert, une *file se réduit essentiellement par l’aval*, et non par l’amont (figure 3.1).

En prenant en compte cette déformation de la file, il est clair que tous ces arrêts “supplémentaires” proviennent des véhicules de la file ne s’étant pas complètement écoulée pendant le vert (première partie de la figure 3.2), qui viendront s’arrêter progressivement, durant le rouge, au niveau de la ligne de feu (deuxième partie de la figure 3.2). Enfin, on fera l’approximation que, lorsque le feu repasse au vert, une seule file, homogène spatialement, subsiste. C’est-à-dire que l’on considère que les véhicules ont le temps de se réorganiser en file homogène pendant chaque rouge.

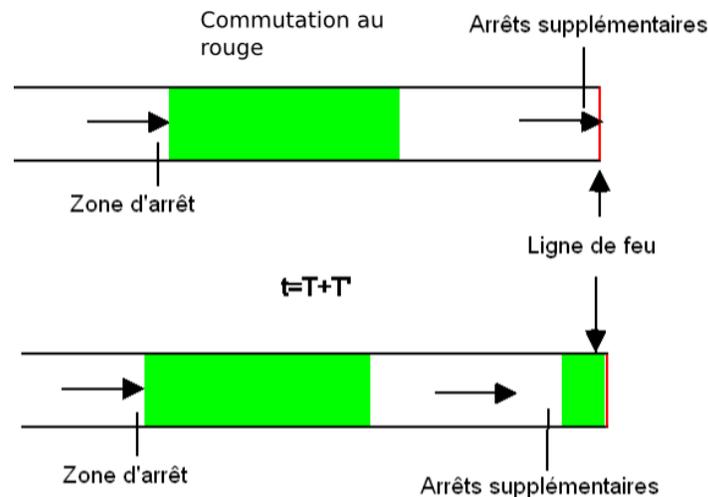


FIGURE 3.2 – Modélisation spatiale d’une file pendant un vert

En résumé, dans cette modélisation, on considère que pour un tronçon donné,

- Lorsque le feu est vert on a, pour chaque instant t pendant lequel le feu est vert :
 - si $l(t) = 0$, alors le nombre d’arrêts à l’instant t sur le tronçon est nul
 - si $l(t) > 0$, alors le nombre d’arrêts à l’instant t vaut $QA(t)$
- Lorsque le feu est rouge, le nombre d’arrêts pendant toute la durée du rouge vaut : $l(t_0) + N_{AR}$, où t_0 est la date de commutation au rouge, et N_{AR} le nombre de véhicules arrivés pendant le rouge. Le calcul de N_{AR} dépendra du modèle choisi pour l’optimisation.

3.1.5 Modélisation des bus

3.1.5.1 Demande de la compagnie de bus, temps d’attente des bus

Dans la modélisation qui suit, on peut tout aussi bien supposer que la compagnie de bus demande à ses bus de respecter un horaire fixe de passage à ses stations, ou bien qu’elle adapte elle-même le plan de circulation des bus au trafic, notamment en terme de fréquence de passage.

Ainsi, dans les deux cas, un bus ne respectant pas le plan pourra être signalé par la compagnie qui souhaiterait que l’avance ou le retard pris par le bus soit corrigé(e). Si la compagnie décide de ne pas le signaler, il pourra être considéré comme un véhicule particulier. La seule information utile, dans le premier cas, pour le système, est le temps d’arrêt du bus au feu souhaité par la compagnie. Un bus *communiqué par la compagnie* et qui atteindra le carrefour, c’est-à-dire pénétrant sur le tronçon correspondant, *dans l’horizon de temps*, sera pris en compte par le système uniquement sous ces conditions. On déclarera alors le bus “actif”. Si l’une des conditions n’est pas respectée, il sera dit “passif”.

Le temps d’arrêt souhaité pour un bus actif peut être non nul. En effet, ce sera le cas, par exemple, lorsqu’un bus sera considéré en avance par rapport au plan initial. Il se peut aussi que le planning des bus change, la compagnie préférant espacer ses bus plutôt que de minimiser le temps total d’attente des bus sur le réseau. Ce genre de contrainte de régularité arrive plutôt aux heures de pointe. On suppose que l’information peut être fournie en temps réel par la compagnie.

Un bus actif peut être stoppé soit directement par un feu au rouge soit par un autre véhicule, même au vert. Pour chaque bus actif b on appelle respectivement *temps d'attente du bus* (TAB_b) et *consigne d'attente* (TAB_b^0) le temps qu'il passe à l'arrêt sur le carrefour et le temps d'arrêt souhaité par la compagnie, pendant l'horizon de temps H . Et on appelle *erreur de consigne* la quantité $EC_b = |TAB_b - TAB_b^0|$.

Le critère “*erreur totale de consigne pour les bus*” (ECB), est alors défini comme suit :

$$ECB = \sum_{b \in B} EC_b = \sum_{b \in B} |TAB_b - TAB_b^0|$$

Où B désigne l'ensemble des bus actifs (on remarquera que le critère ne prend en compte que les bus actifs).

3.1.5.2 Calcul des TAB et interaction avec les véhicules particuliers

On veut calculer le temps d'attente TAB_b , du bus b , sur le carrefour.

On dit qu'un bus est “arrivé” au moment où il entre dans le tronçon qu'il doit emprunter pour traverser le carrefour. Ce tronçon est supposé connu à l'avance. Soit le bus passe toujours par le tronçon, soit la compagnie communique le nouveau trajet en même temps que le bus est signalé par la compagnie.

Soit b un bus actif. Deux cas peuvent se présenter, comme indiqué dans les deux sections suivantes.

Le bus n'est pas encore arrivé

Si le bus est prévu d'arriver pendant un rouge, le temps restant avant le prochain vert sera entièrement compté. Mais juste après la commutation, il se peut que le bus attende encore, à cause de la file devant lui, non encore résorbée. On inclura donc aussi dans TAB_b ce temps supplémentaire. Si le bus est prévu d'arriver pendant un vert, on ne comptera que le temps qu'il faudra pour résorber la file devant lui. Dans les deux cas, on note $T_b \in [0, \dots, H]$ la date à laquelle le bus arrivera sur le tronçon.

On suppose qu'une file ne peut dépasser, en longueur, la longueur de son tronçon. En supposant que, en général, le bus se déplace plus vite que la vitesse de résorption

d'une file, on fait l'approximation que la longueur de la file précédant le bus à T_b , au moment où il pénètre sur le tronçon, sera la même lorsque le bus arrivera "en contact" avec elle, c'est-à-dire au moment où il sera stoppé par le dernier véhicule constituant la file (première partie de la figure 3.3). L'effet de cette approximation sera une légère surestimation de TAB_b .

Le bus est déjà arrivé

Si le bus n'est pas encore à l'arrêt, on fait le même genre d'approximation que précédemment : on considère que la file courante précédant le bus, fournie par les capteurs, n'aura pas changé de longueur lorsque le bus y sera stoppé. S'il est déjà à l'arrêt, on suppose que les capteurs peuvent nous fournir directement la longueur de la file précédant celui-ci (la partie rouge sur la seconde partie de la figure 3.3).

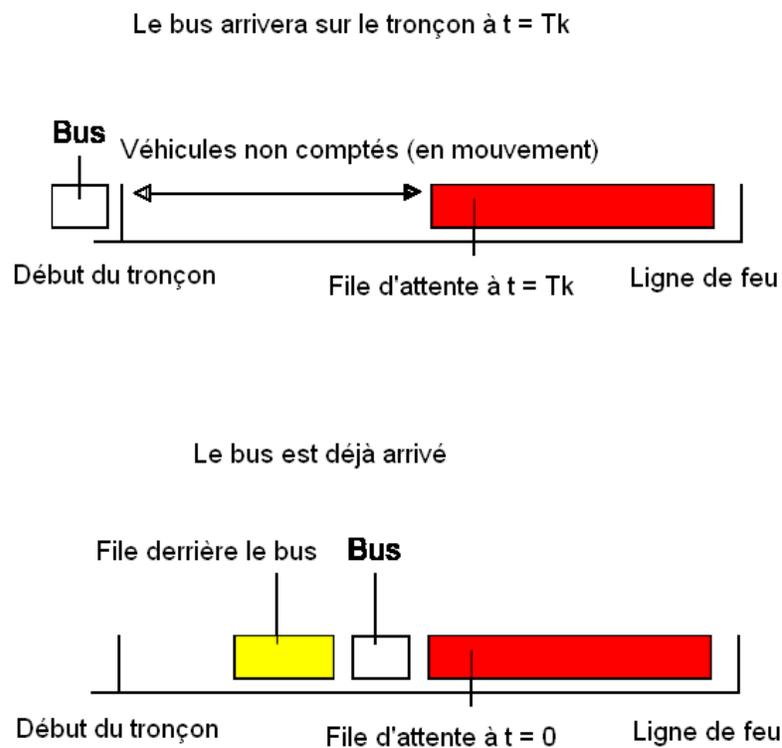


FIGURE 3.3 – Interaction bus-véhicules

Contraintes supplémentaires permettant d'éviter la présence du bus sur le tronçon pendant deux mêmes états consécutifs

Que le bus soit arrivé ou non, pour éviter sa présence sur le tronçon pendant deux mêmes états (du feu) consécutifs, nous ajoutons les contraintes suivantes :

1. Si un bus est arrivé ou est prévu d'arriver durant un rouge, toute la file présente au moment de la commutation au vert suivant ce rouge devra être résorbée durant le vert correspondant.
2. Si un bus est arrivé ou est prévu d'arriver durant un vert, la durée entre T_b et la prochaine commutation (au rouge), sera imposée supérieure ou égale au minimum des deux valeurs suivantes : le temps de résorption de la file précédant le bus et le temps nécessaire au bus pour traverser le tronçon entier sans obstacle.

On pourra éventuellement considérer des contraintes "souples", c'est-à-dire pouvant potentiellement être violées, mais entraînant un malus important reporté sur le critère bus lorsque l'une d'elle l'est. Cela permet d'éviter les cas où le solveur ne trouve aucune solution à cause du fait que la file devant le bus ne peut pas être écoulée en une seule fois (cas saturé).

3.2 Présentation générale du système

Le but de cette section est de présenter le système développé pour la thèse, dans lequel vont s'insérer les modèles d'optimisation.

On y trouve une description générale du système (sous-section 3.2.1), où apparaît, en outre, le schéma interactif permettant de prendre en compte une politique de régulation. En sous-section 3.2.2, la façon dont est modélisée l'évolution de l'instance est abordée.

La formulation générale de la fonction objectif, ainsi que les notations, qui sont toutes deux communes aux trois modèles y sont également indiquées (sous-sections 3.2.3 et 3.2.4).

Remarque : L'évolution précise des paramètres préférentiels utilisés dans la fonction objectif fera l'objet d'une section à part entière (section 3.7), et n'est donc pas abordée dans cette section.

3.2.1 Description générale du système

Comme nous l'avons évoqué précédemment, nous avons élaboré un système relativement flexible permettant de réguler de façon adaptative le trafic sur un carrefour à feux. Le fonctionnement général (figure 3.4) se fait itérativement, via une boucle reliant un bloc d'optimisation et l' "environnement". L' "environnement", dans notre contexte, est constitué du carrefour, du contrôleur et des capteurs. L'environnement fournit les informations suivantes à chaque itération :

- Les états courants des feux.
- Les dernières dates de commutation (passées).
- Les longueurs des files d'attente de chaque tronçon à l'instant présent, tous véhicules confondus, grâce aux capteurs.
- Les débits d'arrivées des véhicules particuliers, sur chaque tronçon, à l'instant présent (ces débits font partie de l' "État du trafic" de la figure 3.4) : ces derniers sont éventuellement stockés pendant un certain temps, afin de pouvoir procéder à une régression linéaire permettant de prévoir les débits sur l'horizon de temps futur H ("Débits" sur la figure 3.4) pour effectuer la prochaine optimisation.

Le bloc "Règles" comporte deux types de règles :

- Les règles permettant de calculer les débits futurs des véhicules particuliers, à partir des débits courants ou d'un historique des débits.
- Les règles permettant de calculer la politique de régulation pour la prochaine itération. La nouvelle politique est calculée en fonction de l'"État du trafic" courant.

L'état (ou situation) du trafic peut être appréhendé de différentes manières, et par conséquent être représenté par différents indicateurs qui seront développés en section 3.7. On peut en particulier considérer qu'à une itération donnée, la solution obtenue dans l'espace des critères représente aussi, d'un certain point de vue, la situation du trafic. Par conséquent, il se peut que l'on veuille utiliser ultérieurement les informations relatives aux solutions obtenues dans l'espace des critères pour orienter la politique de régulation. Ces solutions sont donc mémorisées au fur et à mesure des itérations.

A chaque itération, le bloc optimisation prend comme instance :

- la longueur de chaque file d'attente sur chaque tronçon
- les débits sur chaque tronçon

- la politique de régulation
- la demande provenant la compagnie de bus considérée, qui sera détaillée en sous-section 3.1.5

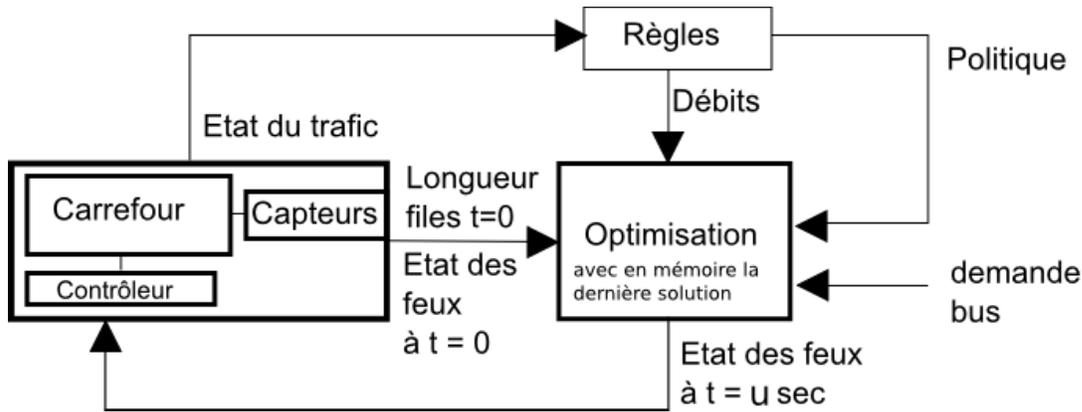


FIGURE 3.4 – Schéma général du système

Sur le schéma de la figure 3.4, $t = 0$ représente l'instant présent, et u est un intervalle de temps tel qu'on itère le procédé tous les u secondes. Ainsi, si l'on souhaite prendre en compte de l'information en provenance des capteurs le plus régulièrement possible, on essaiera de prendre u le plus petit possible. Le taux de révision des capteurs étant de 1 seconde, on aura cependant nécessairement $u \geq 1$.

3.2.2 Évolution de l'instance

Dans la procédure interactive temps réel telle que décrite en section 1.2.6, et comme on peut aussi le comprendre dans la section précédente (section 3.2.1), l'instance du problème d'optimisation à résoudre évolue au fil des itérations. On décrit ici de quoi est constituée exactement cette instance et comment cette instance va évoluer.

On distingue plusieurs types d'objets de cette instance : ceux qui n'évoluent jamais et ceux qui sont susceptibles d'évoluer.

3.2.2.1 Les objets "statiques"

Ces objets n'évoluent pas dans le temps. Autrement dit, ce sont des constantes aussi bien dans le cadre d'une optimisation à un instant donné que dans le cadre de la

méthode globale interactive.

Les objets suivants sont des objets statiques.

- La “structure” du carrefour représentée les relations d’antagonisme retenues entre les feux. L’élaboration de la matrice des antagonismes se fait par les exploitants et dépend principalement de la configuration géométrique du carrefour selon AFNOR (1991).
- Les durées minimales et maximales de rouge et de vert pour chaque feu : ces valeurs sont également fixées par l’exploitant, établies sur la base des normes françaises (AFNOR, 1991).
- Les rouges de dégagement.

3.2.2.2 Les objets “dynamiques”

Ces objets sont susceptibles d’évoluer dans le temps. Cependant, ils sont à distinguer des variables de décision du problème d’optimisation à résoudre à un instant donné.

Les objets suivants sont des objets dynamiques.

- La longueur des files d’attente des véhicules particuliers (VP).
- Les débits d’arrivée.
- Pour l’aspect “transport en commun” : la demande de la compagnie de bus prise en compte, qui porte sur des bus spécifiques, à un instant donné.

3.2.3 Formulation générale de la fonction objectif commune aux trois modèles

Les fonctions d’agrégation à *base de point de référence* (voir la section 1.2.4), et en particulier de forme augmentée, semblent adaptées à la problématique de la thèse, en regard des avantages qu’elles procurent, évoqués dans la sous-section 1.2.4.

Avec les critères TA , NA et ECB , la fonction d’agrégation $f_{\bar{z},\lambda}$ correspondante s’écrit :

$X \rightarrow \mathbb{R}$

$$x \mapsto \max \{ \lambda_{TA} (TA(x) - \overline{TA}), \lambda_{NA} (NA(x) - \overline{NA}), \lambda_{ECB} (ECB(x) - \overline{ECB}) \} + \rho [\lambda_{TA} (TA(x) - \overline{TA}) + \lambda_{NA} (NA(x) - \overline{NA}) + \lambda_{ECB} (ECB(x) - \overline{ECB})]$$

où ρ est fixé conformément à la section 1.2.4 et $\lambda = (\lambda_{TA}, \lambda_{NA}, \lambda_{ECB})$, $\bar{z} = (\overline{TA}, \overline{NA}, \overline{ECB})$ sont les paramètres préférentiels à adapter en fonction de la situation du trafic, et dont les valeurs représenteront la politique de régulation à un instant donné.

Dans les sections dédiées aux développements des modèles en eux-mêmes (sections 3.3 à 3.6), nous supposons que λ et \bar{z} sont fixes pour nous intéresser à la formulation des programmes, et en particulier des critères, en fonction des variables du programme. La manière dont vont évoluer les paramètres préférentiels en fonction de la situation du trafic sera abordée en section 3.7.

3.2.4 Notations utilisées pour les trois modèles

3.2.4.1 Ensembles

On rappelle que H désigne l'horizon de temps. On appelle S l'ensemble $\{1, \dots, H\}$ et on note $S' = \{1, \dots, H - 1\}$.

On note $F = \{1, \dots, N_F\}$ l'ensemble des feux du carrefour, où N_F représente le nombre de feux sur le carrefour. On identifiera dans la suite tronçons et feux, car on a précisé dans la section 1.1.1 qu'il y a, selon notre terminologie, un feu par tronçon.

Enfin B représente l'ensemble des bus actifs.

3.2.4.2 Constantes

Pour tout $f \in F$, on note :

- $QA_{s,f}$ le débit des arrivés des véhicules particuliers sur le tronçon f , en véhicules par seconde, à l'instant $s \in S$. On note également $QA_f = QA_{0,f}$.
- QS_f le débit de saturation, en véhicules par seconde.

- $l_{0,f}$ la longueur de la file, en nombre de véhicules, sur le tronçon f , à l’instant $t = 0$. Ces longueurs sont fournies chaque seconde par les capteurs.

A $t = 0$, on connaît les états de chaque feu. Cela nous permet de connaître l’état de chaque feu après chacune de ses commutations à venir.

Les durées minimum et maximum des états, verts et rouges, des feux sont considérés, pour simplifier, identiques pour tous les feux. On note respectivement D_V^{min} , D_V^{max} , D_R^{min} et D_R^{max} les durées minimum et maximum des états, verts et rouges, des feux.

On considère également que tous les rouges de dégagement sont identiques et on le note D_R .

En France et dans d’autres pays, il existe des carrefours où les durées minimum et maximum des états, verts et rouges sont différentes suivant les feux. Il serait très simple de modifier les différents programmes proposés pour prendre en compte cela, sans modifier en aucune façon le principe d’un des modèles. Il en va de même pour les rouges de dégagements qui pourraient différer d’une paire de feux à l’autre.

Enfin, pour chaque $b \in B$, on note T_b la date d’arrivée de chaque bus actif. On rappelle que TAB_b , TAB_b^0 et EC_b désignent respectivement le temps que le bus b passe sur le carrefour, sa consigne de temps d’attente, et l’erreur de consigne relative au bus b , pendant l’horizon de temps H .

3.3 Modèle “par états”

3.3.1 Présentation

Dans le modèle par états, nous considérons l’ensemble des états $e_{s,f}$ de chaque feu f à l’instant s comme variables principales. La figure 3.5 est un exemple de plan de feux ainsi obtenu sur un horizon H (chaque ligne correspond à un feu).

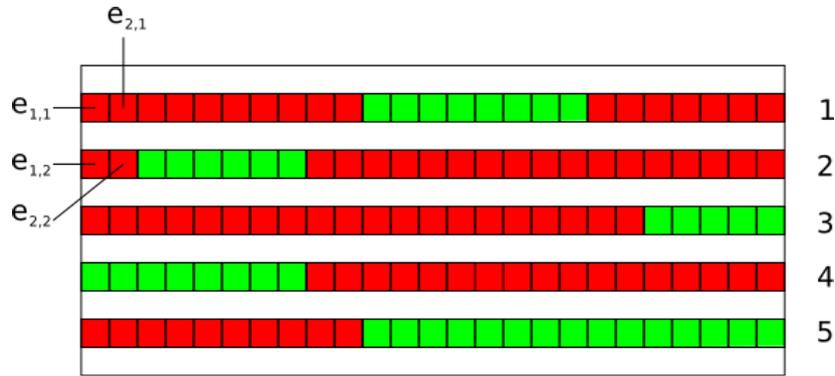


FIGURE 3.5 – Modèle par états

3.3.2 Variables

3.3.2.1 Variables principales

Pour chaque instant $s \in S$ et chaque feu $f \in F$, la variable $e_{s,f}$ représente l'état du feu f à l'instant s .

Ainsi, on a :

$$e_{s,f} = \begin{cases} 1 & \text{si le feu } f \text{ est au vert à l'instant } s \\ 0 & \text{sinon} \end{cases}$$

3.3.2.2 Variables représentant les longueurs de file

$l_{s,f}$ représente la longueur de la file, en nombre de véhicules, sur le tronçon $f \in F$, à l'instant $s \in S$.

3.3.3 Contraintes

3.3.3.1 Calcul des files d'attentes

Pour tout couple $(s, f) \in S \times F$, la file d'attente $l_{s,f}$ est définie par :

$$l_{s,f} = \max(l_{s-1,f} + QA_{s,f} - QS_f \times e_{s,f}, 0)$$

On linéarise avec les contraintes :

$$\begin{aligned} l_{s,f} &\geq l_{s-1,f} + QA_{s,f} - QS_f \times e_{s,f} \\ l_{s,f} &\geq 0 \end{aligned}$$

3.3.3.2 Bornes sur les durées des états des feux

Ces contraintes correspondent à des normes imposées : durées minimales et maximales de vert et de rouge.

Durées maximales de vert D_V^{max}

Les contraintes de durées maximales de verts peuvent se traduire de la manière suivante : pour chaque feu $f \in F$, chaque groupe de variables correspondant à $D_V^{max} + 1$ créneaux de temps successifs ne peut avoir au plus que D_V^{max} variables égales à 1 (ce qui correspond au vert pour f). D'où les contraintes suivantes :

$$\sum_{s=s_d}^{s_d+D_V^{max}} e_{s,f} \leq D_V^{max}, s_d \in \{1, \dots, H - D_V^{max}\}, f \in F$$

En pratique, il faut aussi prendre en compte les créneaux de temps déjà fixés, c'est-à-dire que s_d ne commence pas à 1 mais à $1 - D_V^{max}$.

Durées maximales de rouge D_R^{max}

Les contraintes de durées maximales de rouges peuvent se traduire de la manière suivante : pour chaque feu $f \in F$, chaque groupe de variables correspondant à $D_R^{max} + 1$ créneaux de temps successifs possède au moins une variable égale à 1 (ce qui correspond au vert pour f). D'où les contraintes suivantes :

$$\sum_{s=s_d}^{s_d+D_R^{max}} e_{s,f} \geq 1, s_d \in \{1, \dots, H - D_R^{max}\}, f \in F$$

En pratique, il faut aussi prendre en compte les créneaux de temps déjà fixés, c'est-à-dire que s_d ne commence pas à 1 mais à $1 - D_R^{max}$.

Durées minimales de vert D_V^{min}

Les contraintes de durées minimales de verts peuvent se traduire de la manière suivante : pour chaque feu $f \in F$, si deux variables d'état, correspondant à des créneaux de temps espacés dans le temps d'une durée D_V^{min} , sont à 0 (f est au rouge), alors il ne peut y avoir aucun vert dans l'intervalle de temps qui sépare les deux créneaux. D'où les contraintes suivantes :

$$e_{s,f} \leq e_{s_1,f} + e_{s_2,f}, (s_1, s_2) \in S^2, s_1 < s_2 \leq s_1 + D_V^{min}, f \in F$$

En pratique, il faut aussi prendre en compte les créneaux de temps déjà fixés, c'est-à-dire que s_1 ne commence pas à 1 mais à $1 - D_V^{min}$.

Durées minimales de rouge D_R^{min}

Par le même raisonnement que dans la sous-section précédente on obtient :

$$e_{s,f} \geq e_{s_1,f} + e_{s_2,f} - 1, (s_1, s_2) \in S^2, s_1 < s_2 \leq s_1 + D_R^{min}, f \in F$$

En pratique, il faut aussi prendre en compte les créneaux de temps déjà fixés, c'est-à-dire que s_1 ne commence pas à 1 mais à $1 - D_R^{min}$.

3.3.3.3 Antagonismes

Pour toute paire (f, f') de feux antagonistes et pour chaque seconde s , on souhaite que les deux feux ne puissent pas être au vert, c'est-à-dire que $e_{s,f}$ et $e_{s,f'}$ ne peuvent pas prendre la valeur 1 en même temps. D'où les contraintes :

$$e_{s,f} + e_{s,f'} \leq 1, f \in F, f' \in F$$

3.3.3.4 Rouges de dégagement

Pour chaque feu f et chaque feu f' antagoniste au feu f , f' doit rester au rouge pendant un temps R_D après un passage au rouge de f . f passe au rouge entre la seconde s et la seconde $s + 1$ si et seulement si $(1 - e_{s,f}) + e_{s+1,f} = 0$. Par ailleurs, l'expression $(1 - e_{s,f}) + e_{s+1,f}$ ne peut prendre que les valeurs 0, 1 ou 2. D'où les contraintes suivantes :

$$e_{s+k,f'} \leq (1 - e_{s,f}) + e_{s+1,f}, f \in F, f' \in A(f), s \in S, k \in \{1, \dots, R_D\}$$

où $A(f)$ désigne l'ensemble des feux antagonistes au feu $f \in F$.

3.3.4 Critères

3.3.4.1 Temps d'attente

En faisant l'approximation que la longueur d'une file d'attente sur un tronçon est constante entre deux créneaux de temps (soit une seconde), le temps d'attente cumulé TA_f pour un tronçon f sur l'horizon de commutation H est donné par :

$$TA_f = \sum_{s=1}^{H-1} l_{s,f} \times 1 = \sum_{s=1}^{H-1} l_{s,f}$$

Le temps d'attente cumulé sur le carrefour s'écrit alors :

$$TA = \sum_{f=1}^{N_F} TA_f = \sum_{f=1}^{N_F} \sum_{s=1}^{H-1} l_{s,f}$$

qui bien sûr est linéaire.

3.3.4.2 Nombre d'arrêts

A chaque instant $s \in S'$, et pour chaque feu $f \in F$, le nombre d'arrêts $NA_{s,f}$ est nul si et seulement si la file d'attente correspondante est vide. De plus, si la file est non vide, le nombre d'arrêts vaut $QA_{s,f}$.

Soit $xl_{s,f}$ les variables 0-1 telles que :

$$xl_{s,f} = \begin{cases} 0 & \text{si } l_{s,f} = 0 \\ 1 & \text{sinon} \end{cases}$$

Les contraintes suivantes permettent d'obtenir les propriétés souhaitées pour ces variables :

$$xl_{s,f} \geq \frac{l_{s,f}}{M}$$

En effet, les variables $xl_{s,f}$, qui interviendront positivement dans le critère NA , à minimiser, prendront naturellement la valeur 0 si elles ne sont soumises à aucune contrainte. Il suffit donc de les forcer à 1 lorsque $l_{s,f} = 1$.

On aurait simplement alors :

$$\forall (s, f) \in S' \times F, NA_{s,f} = QA_{s,f} \times xl_{s,f}$$

Cependant, la modélisation des files d'attentes proposée en 3.1.4 implique la prise en compte d'arrêts supplémentaires au passage au rouge, égaux à $NA_{s,f}^{suppl} = QA_{s,f} \times l_{s,f}$ pour chaque instant s où le feu f passe au rouge. D'où, en prenant en compte ces arrêts supplémentaires :

$$NA_{s,f} = QA_{s,f} \times xl_{s,f} + NA_{s,f}^{suppl}$$

Mais il faut rajouter les contraintes :

$$\begin{aligned} NA_{s,f}^{suppl} &\geq 0 \\ NA_{s,f}^{suppl} &\geq QA_{s,f} \times (l_{s,f} - M \times (e_{s+1} - e_{s,f} + 1)) \end{aligned}$$

Le terme $-M \times (e_{s+1} - e_{s,f} + 1)$ invalide la contrainte pour les instants qui ne correspondent pas à un passage au rouge sur la ligne f . M est un majorant de $l_{s,f}$, c'est-à-dire de la longueur d'une file en général.

On remarque que tout est linéaire. Cependant, il aura fallu rajouter $(H - 1) \times N_F$

variables 0-1.

3.3.4.3 Critère Bus

Enfin, l'erreur totale de consigne pour les bus s'écrit :

$$ECB = \sum_{b \in B} EC_b$$

Soit x_{s,f_b} , $b \in B$, $s \in \{1, \dots, H - T_b\}$ les variables booléennes telles que,

$$x_{s,f_b} = \begin{cases} 1 & \text{si } e_{s+T_b,f_b} = 0 \\ 0 & \text{sinon} \end{cases}$$

L'expression $\sum_{s=0}^{D_R^{max}} x_{s,f_b}$ représente alors la durée d'attente au rouge du bus b sur son tronçon f_b .

Les contraintes suivantes permettent d'obtenir les propriétés souhaitées pour les variables x_{s,f_b} . Pour chaque $s \in \{1, \dots, D_R^{max}\}$, et chaque bus $b \in B$:

$$\begin{aligned} x_{s,f_b} &\leq 1 - e_{s+T_b,f_b} \\ x_{s,f_b} &\leq x_{s-1,f_b} \\ x_{s,f_b} &\geq \frac{1}{2} [(1 - e_{s+T_b,f_b}) + x_{s-1,f_b}] \end{aligned}$$

En prenant en compte la durée $\frac{l_{T_b,f_b}}{QS_{f_b}}$ de l'écoulement de la file d'attente précédant le bus lors de son arrivée à $t = T_b$, on a alors, pour chaque bus $b \in B$ les contraintes suivantes :

$$\begin{aligned} EC_b &\geq \left[\sum_{s=0}^{D_R^{max}} x_{s,f_b} + \frac{l_{T_b,f_b}}{QS_{f_b}} \right] - TAB_b^0 \\ EC_b &\geq TAB_b^0 - \left[\sum_{s=0}^{D_R^{max}} x_{s,f_b} + \frac{l_{T_b,f_b}}{QS_{f_b}} \right] \\ x_{0,f_b} &= 1 - e_{T_b,f_b} \end{aligned}$$

Observons que les deux premières contraintes suffisent à modéliser la valeur absolue de la différence du fait que le terme EC_b intervient en minimisation dans le critère Bus.

Contraintes additionnelles pour les bus

Il s'agit de prendre en compte les contraintes liées à la présence des bus sur leur tronçon (sous-section 3.1.5.2).

On considère, dans cette sous-section, les variables y_{s,f_b} , $s \in S$, $b \in B$, similaires aux variables x_{s,f_b} , mais qui sont nulles à partir de la première commutation au rouge après T_b (au lieu de la première commutation au vert après T_b pour les variables x_{s,f_b}). En utilisant ces deux types de variables, l'expression de la durée du vert suivant T_b est $\sum_{s=T_b}^H e_{s,f_b} \times (1 - x_{s,f_b}) \times y_{s,f_b}$ (que le feu soit vert à $s = T_b$ ou non).

Les contraintes correspondant au point 1 du paragraphe relatif aux contraintes supplémentaires pour les bus (voir la sous-section 3.1.5.2) s'écrivent alors :

$$\sum_{s=T_b}^H e_{s,f_b} \times (1 - x_{s,f_b}) \times y_{s,f_b} \geq (1 - e_{T_b,f_b}) \frac{l_{T_b,f_b}}{QS_{f_b}}, b \in B$$

Et les contraintes correspondant au point 2 du même paragraphe s'écrivent :

$$\sum_{s=T_b}^H e_{s,f_b} \times (1 - x_{s,f_b}) \times y_{s,f_b} \geq e_{T_b,f_b} \times \max \left\{ \frac{l_{T_b,f_b}}{QS_{f_b}}, \Delta T_{travel}^b \right\}, b \in B$$

Ces contraintes sont aisément linéarisables.

3.4 Modèle “par dates”

3.4.1 Présentation

Le modèle par dates est un modèle dans lequel on considère comme variables les dates de commutation des feux sur l'horizon de temps plutôt que les états des feux à un instant donné de l'horizon de temps.

Dans le modèle par états, on faisait correspondre à chaque feu un ensemble de variables correspondant à l'état du feu à un instant donné. Il semblerait naturel de procéder de façon similaire dans le modèle par dates, en attribuant à chaque feu un ensemble de variables qui correspondraient aux différentes dates de commutation du feu sur l'horizon de temps. Cependant, cela poserait un problème majeur quant aux contraintes relatives aux antagonismes. En effet, le fait que deux feux soient antagonistes se traduirait par un grand nombre de contraintes *disjonctives*. Supposons que deux feux, le feu 1 et le feu 2 soient antagonistes. Alors pour tout couple de dates de commutation $(t_{1,v}, t_{1,r})$ tel que $t_{1,v}$ correspond à une commutation au vert du feu 1 et $t_{1,r}$ la commutation au rouge du feu 1 immédiatement suivante, et pour tout couple de commutation $(t_{2,v}, t_{2,r})$ tel que $t_{2,v}$ correspond à une commutation au vert du feu 2 et $t_{2,r}$ la commutation au rouge du feu 2 immédiatement suivante, on aurait la contrainte disjonctive suivante :

$$\begin{aligned} t_{2,v} &\geq t_{1,r} \\ &OU \\ t_{1,v} &\geq t_{2,r} \end{aligned}$$

On aurait alors un nombre de contraintes disjonctives de l'ordre de $\sum_{(f_1, f_2) \in A} C_{f_1} \times C_{f_2}$, où C_f est le nombre de commutations considérées pour le feu f , et A est l'ensemble des feux antagonistes. Les contraintes disjonctives ne peuvent pas être modélisées telles quelles dans un programme linéaire. Le traitement de telles contraintes est un problème difficile, bien connu en particulier dans le domaine de l'ordonnancement (voir, par exemple (Aloulou et Artigues, 2010)). Dans notre cas, leur linéarisation nécessiterait l'ajout de variables 0-1, dont le nombre est du même ordre que le nombre de contraintes, soit $\sum_{(f_1, f_2) \in A} C_{f_1} \times C_{f_2}$.

Pour éviter les contraintes disjonctives, dans notre contexte, il suffit de considérer l'ensemble des commutations possibles sur l'horizon de temps *tous feux confondus*.

Ainsi, nous considérerons les commutations successives des feux, indicées dans l'ordre chronologique de leur réalisation, comme variables principales. Si deux commutations se produisent simultanément, alors elles portent le même indice. La figure 3.6 est un exemple de plan de feux ainsi obtenu sur un horizon H (chaque ligne correspond à un

feu).

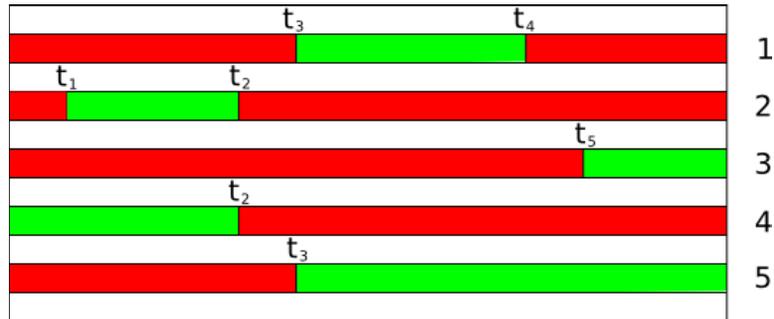


FIGURE 3.6 – Modèle par dates

3.4.2 Constantes

Soit N_C un majorant du nombre de commutations possibles, tous feux confondus, pendant l’horizon de temps. N_C peut être grossièrement calculé en considérant que, sur chaque feu, le nombre de commutations nécessaires pour atteindre l’horizon de temps si chaque durée, de vert ou de rouge, est à son minimum.

Cependant on peut aussi choisir N_C par expérimentations. En effet, plus la valeur de N_C est grande et plus le nombre de solutions admissibles est grand, et par conséquent plus la “qualité” de la solution a des chances d’être meilleure. D’un autre côté, plus N_C est faible et plus la résolution est rapide.

Les commutations sont numérotées dans l’ordre de leur apparition dans le temps. On note $C = \{1, \dots, N_C\}$ et $C' = \{1, \dots, N_C - 1\}$.

3.4.3 Variables

3.4.3.1 Variables principales

Pour chaque $c \in C$, la variable t_c est la date de la $c^{\text{ième}}$ commutation tous feux confondus, à partir de l’instant présent.

Ces variables sont non-négatives. Elles peuvent être entières ou non, suivant le type de contrôleurs utilisé sur le carrefour.

Pour représenter la couleur des feux, nous utiliserons les variables 0-1 $x_{c,f}$. $x_{c,f} = 1$ signifiera que le feu f est au vert entre t_c et t_{c+1} , et $x_{c,f} = 0$ signifiera que le feu f est au rouge entre t_c et t_{c+1} , $f \in F$, $c \in C'$.

3.4.3.2 Variables représentant les longueurs de file

$l_{c,f}$ représente la longueur de la file, en nombre de véhicules, sur le tronçon f , à la date t_c , $c \in C$, $f \in F$.

3.4.4 Contraintes

3.4.4.1 Contraintes sur les dates de commutation

On impose pour chaque $c \in C'$:

$$t_{c+1} - t_c \geq 0$$

Ces contraintes permettent d'ordonner les dates de commutation conformément à l'ordre naturel de l'ensemble C . Notons qu'on ne sépare volontairement pas une date et sa suivante d'un pas de temps, car on veut que le cas $t_c = t_{c+1}$ soit toujours possible. On souhaite que cette égalité reste toujours possible pour la raison suivante : on ne connaît pas à l'avance le nombre total de commutations à l'optimum sur l'horizon de temps. Le fait de pouvoir confondre n'importe quelle paire de commutations successives permet d'avoir, à l'optimum un nombre optimal de commutations réellement utilisées.

3.4.4.2 Dates minimale et maximale dans l'horizon de temps

On suppose que seules les u premières secondes de l'horizon sont déjà fixées. Toutes les dates de commutation sont donc supérieures ou égales à u . On a donc :

$$t_1 \geq u$$

La dernière date de commutation est égale à l'horizon de temps :

$$t_{N_C} = H$$

3.4.4.3 Contraintes sur les files d'attente

Malheureusement, dans ce modèle, on ne sait pas à l'avance à quel instant de l'horizon de temps correspondra la $c^{\text{ième}}$ date de commutation, $c \in C$. On suppose donc que les débits d'arrivées des feux $f \in F$ valent tous $QA_f = QA_{0,f}$.

Pour chaque feu $f \in F$, et pour chaque commutation $c \in C$, la longueur de file $l_{c,f}$ est définie comme suit :

$$l_{c,f} = \max(l_{c-1,f} + (QA_f - QS_f) \times (t_c - t_{c-1}), 0)$$

si f est au rouge entre t_c et t_{c+1} , et

$$l_{c,f} = \max(l_{c-1,f} + QA_f \times (t_c - t_{c-1}), 0)$$

si f est au vert entre t_c et t_{c+1} .

Les variables $l_{c,f}$ apparaîtront positivement dans les critères où elles interviennent (TA et NA). Ainsi, comme ces critères sont à minimiser, on peut remplacer les expressions précédentes par :

$$l_{c,f} \geq l_{c-1,f} + (QA_f - QS_f) \times (t_c - t_{c-1})$$

$$l_{c,f} \geq 0$$

si f est au rouge entre t_c et t_{c+1} , et

$$l_{c,f} \geq l_{c-1,f} + QA_f \times (t_c - t_{c-1})$$

$$l_{c,f} \geq 0$$

si f est au vert entre t_c et t_{c+1} .

Enfin, on utilise les variables $x_{c,f}$ pour traduire le fait que f est soit au rouge soit au vert entre t_c et t_{c+1} :

$$\begin{aligned} l_{c,f} &\geq l_{c-1,f} + (QA_f - x_{c,f} \times QS_f) \times (t_c - t_{c-1}) \\ l_{c,f} &\geq 0 \end{aligned}$$

Ces contraintes, qui impliquent des produits de variables 0-1 et de variables continues, sont facilement linéarisables en utilisant la linéarisation de Wu (Williams, 1985).

3.4.4.4 Bornes sur la durée des états

Ces contraintes sont assez délicates à écrire pour la modélisation choisie. Dans cette sous-section, on ne donnera pas l'écriture complète des contraintes, mais on donnera le raisonnement effectué sur un exemple.

Pour chaque feu $f \in F$, on souhaite que le temps écoulé entre une commutation au rouge et une commutation au vert soit compris entre les valeurs D_R^{min} et D_R^{max} . On souhaite, de façon similaire, que le temps écoulé entre une commutation au vert et une commutation au rouge soit compris entre les valeurs D_V^{min} et D_V^{max} .

La difficulté principale, pour écrire ces contraintes, tient au fait qu'on ne dispose pas des dates de commutation de chaque feu, mais seulement des dates de commutation tous feux confondus.

On sait néanmoins qu'entre deux commutations, il existe au moins un feu au rouge, ce qui nous permet d'écrire, pour chaque commutation $c \in C'$:

$$t_{c+1} - t_c \leq D_R^{max}$$

Mais ces contraintes sont les seules vraiment simples.

Prenons l'exemple de la durée minimum de vert. Soit $f \in F$. Pour chaque commutation $c \in C'$, on connaît la couleur du feu grâce à la variable booléenne $x_{c,f}$, qui vaut 1 si et seulement si f est au vert. Supposons que les conditions suivantes soient respectées :

- juste avant t_c , f est au rouge
- juste après t_{c+1} , f est également au rouge.

Alors il suffit d'écrire :

$$t_{c+1} - t_c \geq x_{c,f} \times D_V^{min}$$

Remarquons que si l'une des conditions n'est pas respectée, la contrainte est fausse.

Pour faire en sorte que cette contrainte ne s'applique que lorsque les conditions s'y prêtent, il suffit de prendre en compte l'état du feu avant et après l'intervalle $[t_c, t_{c+1}]$. Le cas le plus général est celui où $c \in \{2, \dots, N_C - 2\}$. On ne connaît alors l'état du feu f avant t_c que par l'intermédiaire de la variable $x_{c-1,f}$. De même, on ne connaît l'état du feu f après t_{c+1} que par l'intermédiaire de la variable $x_{c+1,f}$. Il suffit alors d'écrire :

$$t_{c+1} - t_c \geq (x_{c,f} - x_{c-1,f} - x_{c+1,f}) \times D_V^{min}$$

Ainsi, la contrainte $t_{c+1} - t_c \geq D_V^{min}$ ne s'applique que lorsqu'on est dans les conditions suivantes :

- $x_{c,f} = 1$: f est au vert durant $[t_c, t_{c+1}]$
- $x_{c-1,f} = 0$: f est au rouge juste avant t_c
- $x_{c+1,f} = 0$: f est au rouge juste après t_{c+1}

Dans le cas où le feu f est deux fois de suite au vert durant $[t_c, t_{c+2}]$, on pourra écrire, dans le cas général :

$$t_{c+2} - t_c \geq ((x_{c,f} + x_{c+1,f} - 1) - x_{c-1,f} - x_{c+2,f}) \times D_V^{min}$$

En effet, $(x_{c,f} + x_{c+1,f} - 1) = 1$ si et seulement si f est deux fois de suite au vert durant $[t_c, t_{c+2}]$. Sinon, cette quantité est négative ou nulle, et la contrainte est invalidée.

Pour une commutation c donnée, un feu ne peut être consécutivement au vert, à partir de t_c que $N_C - c$ fois. Ceci donne lieu à $N_C - c$ contraintes.

Toutes commutations et tous feux confondus, on peut ainsi couvrir tous les cas de minoration des durées de vert en $\Theta(N_F \times N_C^2)$ contraintes. Ceci est également le cas pour les autres bornes.

3.4.4.5 Antagonismes

Pour une commutation $c \in C'$ donnée, on veut que, pendant l'intervalle de temps $[t_c, t_{c+1}]$, deux feux antagonistes ne puissent pas être au vert en même temps. Ceci se traduit par les contraintes :

Pour chaque paire (f, f') de feux antagonistes, $x_{c,f} + x_{c,f'} \leq 1$.

De plus, on souhaite que lorsqu'un feu passe au rouge, ses feux antagonistes ne puissent pas passer immédiatement au vert, mais attendent une durée fixée minimale (le rouge de dégagement). Les contraintes correspondantes ne sont pas triviales. En effet, en supposant $c \in \{2, \dots, N_C - 1\}$ par exemple, on pourrait écrire que, pour tout feu f :

$$t_{c+1} - t_c \geq (x_{c-1,f} - x_{c,f}) D_R$$

Ainsi, on n'imposerait la durée minimale D_R que lorsque t_c est la date de commutation au rouge d'un feu f_0 . On a alors $x_{c-1,f_0} - x_{c,f_0} = 1$ (négatif ou nul sinon). Cependant, cela ne vaut que si aucune commutation n'a lieu entre le passage au rouge de f_0 et le passage au vert d'un de ses feux antagonistes. Sinon, il faut minorer une succession d'états rouges sur plusieurs commutations, et on est alors dans un cas similaire au cas présenté en sous-section 3.4.4.4 (durée minimum de vert).

3.4.5 Critères

3.4.5.1 Le temps d'attente pour les véhicules particuliers : TA

Pour chaque feu f , la file sur le tronçon correspondant, l_f , est définie par morceau :

$$\forall f \in F, l_f(t_0) = l_{0,f}$$

Et pour chaque $f \in F$, pour chaque $c \in C$, pour chaque $t \in [t_c, t_{c+1}]$,

$$\begin{aligned}
l_f(t) &= \max \{l_f(t_c) + q_{c,f}(t - t_c), 0\} \\
&= \max \{l_{c,f} + q_{c,f}(t - t_c), 0\}
\end{aligned}$$

$q_{c,f}$ vaut QA_f si le feu f est au rouge pendant $[t_c, t_{c+1}]$, et vaut $QA_f - QS_f$ si le feu f est au vert pendant $[t_c, t_{c+1}]$. Le temps d'attente cumulé TA_f , pour un tronçon f , vaut alors :

$$\begin{aligned}
TA_f &= \sum_{c=1}^{N_c-1} \int_{t_c}^{t_{c+1}} l_f(t) dt \\
&= \sum_{c=1}^{N_c-1} \frac{1}{2} \times (l_{c,f} + l_{c+1,f}) \times \min(\Delta t_c, \Delta t_{c,f}^0)
\end{aligned}$$

où $\Delta t_c = t_{c+1} - t_c$ et $\Delta t_{c,f}^0$ est le temps nécessaire pour écouler la file $l_{c,f}$, sur le tronçon f à partir de $t = t_c$. Ce temps vaut $-\frac{l_{c,f}}{q_{c,f}}$ dans le cas vert, et vaut $+\infty$ dans le cas rouge. Pour obtenir ce résultat, il suffit de considérer les deux trapèzes formés par l'axe des abscisses, la fonction (linéaire sur $[t_c, t_{c+1}]$), la droite $t = t_c$, et les deux droites $t = t_{c+1}$ et $t = t_c + \Delta t_{c,f}^0$ sur le graphique représentant la fonction l_f en fonction de t sur $[t_c, t_{c+1}]$.

Le temps d'attente cumulé vaut donc au total :

$$\begin{aligned}
TA &= \sum_{f=1}^{N_F} TA_f \\
&= \sum_{f=1}^{N_F} \sum_{c=1}^{N_c-1} \frac{1}{2} \times (l_{c,f} + l_{c+1,f}) \times \min(\Delta t_c, \Delta t_{c,f}^0)
\end{aligned}$$

Pour alléger le programme, on peut éventuellement considérer que :

$$\begin{aligned}
TA &= \sum_{f=1}^{N_F} \sum_{c=1}^{N_C-1} \frac{1}{2} \times (l_{c,f} + l_{c+1,f}) \times \Delta t_c \\
&= \sum_{c=1}^{N_C-1} \Delta t_c \times \sum_{f=1}^{N_F} \frac{1}{2} \times (l_{c,f} + l_{c+1,f})
\end{aligned}$$

Ce qui implique une surestimation de TA , dépendant des paramètres suivants : H , D_{max}^V , R_d et D_{min}^R . Cette approximation peut être utile dans le cas où l'on souhaite obtenir des solutions réalisables très rapidement. En effet, on passe d'un nombre de termes quadratiques à linéariser de l'ordre de $N_f \times N_c$, à un nombre de tels termes de l'ordre de N_c , en posant par exemple $Sl_c = \sum_{f=1}^{N_F} \frac{1}{2} \times (l_{c,f} + l_{c+1,f})$, pour chaque $c \in C'$, puis en linéarisant, pour chaque $c \in C'$ les termes $\Delta t_c \times Sl_c$.

Linéarisation

La forme quadratique associée à ce critère n'étant pas définie-positive (ni même semi-définie positive), à cause de la présence de produits " $v_1 \times v_2$ ", avec v_1 et v_2 des variables fractionnaires, la linéarisation est délicate.

Une méthode classique consiste à utiliser des contraintes de type "Special Ordered Set" de type 2 (SOS2) (Williams, 1985). Ce genre de contraintes sont gérées par les solveurs récents grâce à des fonctions dédiées. Si on impose v_1 et v_2 entiers, une autre méthode consiste à écrire l'expression binaire de v_1 et v_2 , et à linéariser leur produit à l'aide des variables 0-1 introduites pour la décomposition (Billionnet, 2007). Cependant, ces méthodes sont toutes gourmandes en temps de calcul.

Pour améliorer la vitesse de l'optimisation, nous avons essayé deux autres approches.

Première approche

La première approche consiste ne décomposer que le terme Sl_c du produit $\Delta t_c \times Sl_c$, en utilisant des contraintes SOS de type 1 (SOS1), considérablement moins pénalisantes, en temps de calcul, que SOS2. Ce type de décomposition, est néanmoins moins précise que

la décomposition en base 2, puisqu'il s'agira de décomposer Sl_c avec un pas supérieur à un pour gagner en temps de calcul. Ceci implique donc une approximation de Sl_c .

Nous avons choisi d'approximer Sl_c plutôt que Δt_c car les termes Sl_c sont déjà des estimations des futures sommes de longueurs de files d'attente, qui dépendent aussi d'un certain aléa dans les arrivées des véhicules. Ce n'est pas le cas des termes Δt_c , qui ne sont pas des estimations mais des intervalles de temps bien définis.

Il semble raisonnable, pour ce critère, d'utiliser seulement des "niveaux" de longueur de file plutôt qu'une valeur exacte alors qu'on a des arrivées aléatoires. Par exemple, on peut considérer des blocs de 5 véhicules plutôt que véhicule par véhicule.

Pour effectuer cela, on a ajouté les variables 0-1 $sl_{c,k}$ sous les contraintes SOS1 et les contraintes suivantes :

$$\forall c \in C', Sl_c = \sum_{v=1}^{\kappa} l_v \times sl_{c,v}$$

où l_1, \dots, l_K sont les constantes permettant d'approximer la somme des files d'attente.

On n'a plus qu'à linéariser le produit de variables 0-1 et de variables continues ($\Delta t_c \times sl_{c,k}$), ce qui peut être effectué avec la linéarisation de Wu déjà évoquée.

Seconde approche

La seconde approche consiste à décomposer simultanément les deux termes du produit $\Delta t_c \times Sl_c$:

- Sl_c est décomposé conformément à la première approche (SOS1)
- Δt_c est décomposé en base 2 (ce qui implique, contrairement à la première approche, d'imposer les variables t_c entières) :

$$\forall c \in C', \Delta t_c = \sum_{k'=0}^{K'} 2^{k'} \times d_{c,k'}$$

où les variables $d_{c,k'}$, $k' = 0, \dots, K'$ sont des variables 0-1 et représentent Δt_c en base 2.

Tous les produits de variables restant à effectuer étant des produits de deux variables 0-1 ($d_{c,k'} \times sl_{c,k}$), la linéarisation est classique.

Conclusion sur la linéarisation

Finalement, quelques expérimentations effectuées sur le carrefour C (voir figure 1.1) ont montré que la deuxième approche était plus performante pour le modèle par dates. Les longueurs de files d'attente sont alors naturellement discrètes, de pas égal à :

$$\min \{QA_f, (QS_f - QA_f)\}, f \in F$$

Dans le cadre de cette approche, préciser au solveur que les longueurs de files prennent des valeurs discrète (en précisant le pas) améliore la résolution en général. Si les débits évoluent dans le temps et qu'ils sont fournis au système avec une précision de ϵ véhicules par seconde, un *pas* naturel pour les longueurs de files est de ϵ véhicules.

3.4.5.2 Le nombre d'arrêts : NA

Selon le modèle établi en 3.1, la fonction objectif se calcule comme suit.

Pour chaque $c \in C'$, soit $NA_{c,f}$ le nombre d'arrêts durant l'intervalle de temps $[t_c, t_{c+1}]$ sur le tronçon $f \in F$. Durant un vert le nombre d'arrêts dépend du temps pendant lequel la file reste non nulle. Plus précisément, si $x_{c,f} = 1$ (vert), on a :

$$NA_{c,f} = QA_f \times \min(\Delta t_c, \Delta t_{c,f}^0)$$

où $\Delta t_{c,tl}^0$ est le temps nécessaire pour écouler la file présente sur la section f à t_c . En revanche, si $x_{c,f} = 0$ (rouge), on a :

$$NA_{c,f} = QA_f \times \Delta t_c + l_{dr(c,f),f}$$

où $dr(c, f) \in C'$ est tel que $t_{dr(c,f)}$ correspond à la dernière commutation au rouge du feu f avant t_c inclus.

Enfin,

$$NA = \sum_{f=1}^{N_F} \sum_{c=1}^{N_c-1} NA_{c,f}$$

L'expression NA n'est pas linéaire. Cependant, sa linéarisation ne nécessite pas l'ajout de beaucoup de nouvelles variables, ni de contraintes :

Chaque terme $\min(\Delta t_c, \Delta t_{c,f}^0)$ étant à minimiser, il peut être remplacé par $y_{c,f} \times \Delta t_c + (1 - y_{c,f}) \times \Delta t_{c,f}^0$ où $y_{c,f}$ est une variable binaire. On remplace également les deux termes dans le min par leur expression : $\Delta t_{c,f}^0 = \frac{l_{c,f}}{QS_f - QA_f}$ et $\Delta t_c = t_{c+1} - t_c$. Il ne reste alors plus qu'à linéariser en utilisant la linéarisation de Wu déjà évoquée. On aura donc ajouté au total $N_C \times N_F$ variables 0-1 et $O(N_C \times N_F)$ contraintes.

3.4.5.3 Erreur totale de consigne pour les bus : ECB

TAB_b est le temps d'attente réel du bus $b \in B$ sur le carrefour durant l'horizon de temps. On note f_b le tronçon qu'il empruntera pour pénétrer sur le carrefour.

Comme on n'est pas capable, en général, de savoir à l'avance entre quelles commutations sera situé T_b , la date d'arrivée du bus $b \in B$ sur le tronçon, on utilisera les variables xb_c^b , qui sont telles que $xb_c^b = 1$ si et seulement si $T_b \in [t_c, t_{c+1}]$.

L'erreur totale de consigne pour les bus, s'écrit comme suit :

$$\begin{aligned} ECB &= \sum_{b=1}^B |TAB_b - TAB_b^0| \\ &= \sum_{b=1}^B \left| \left[\frac{l_{f_b}(T_b)}{QS_{f_b}} + \sum_{c \in C'} (t_{pr(c, f_b)} - T_b) xb_c^b \right] - TAB_b^0 \right| \end{aligned}$$

où $l_{f_b}(T_b)$ est la longueur de la file précédant le bus b à $t = T_b$ et $\frac{l_{f_b}(T_b)}{QS_{f_b}}$ est le temps qu'il lui faudra pour être résorbée (sous l'hypothèse faite dans la modélisation qu'on peut écouler la file en une seule fois).

L'expression $(t_{pr(c, f_b)} - T_b)$ est le temps de rouge restant si le bus b est arrivé au rouge où $pr(c, f_b) \in C'$ est tel que $t_{pr(c, f_b)}$ correspond à la première commutation au rouge du feu f_b après t_{c+1} inclus.

De plus, précisons que s'il y a des voies dédiées pour les bus, le terme $\frac{l_{f_b}(T_b)}{QS_{f_b}}$ disparaît pour les bus b concernés.

Les valeurs absolues sont facilement linéarisables. Pour chaque bus b , on considère l'erreur de consigne EC_b relative à b et on ajoute les contraintes suivantes :

$$EC_b \geq \left[\frac{l_{f_b}(T_b)}{QS_{f_b}} + \sum_{c \in C'} (t_{pr(c, f_b)} - T_b) xb_c^b \right] - TAB_b^0$$

$$EC_b \geq TAB_b^0 - \left[\frac{l_{f_b}(T_b)}{QS_{f_b}} + \sum_{c \in C'} (t_{pr(c, f_b)} - T_b) xb_c^b \right]$$

Les linéarisations restantes peuvent se faire avec la méthode de Wu, déjà évoquée. Enfin, on écrit

$$ECB = \sum_{b \in B} EC_b$$

Si le bus est déjà arrivé, $l_{f_b}(T_b)$ est remplacé par $l_{f_b}^0$, la longueur de la file précédant le bus (fournie par les capteurs), T_b est remplacé par 0, et xb_c^b par 0 si le feu concerné est vert, et par 1 s'il est rouge.

Le paragraphe suivant est dédié aux contraintes permettant de calculer les variables mises en jeu dans ce critère ECB , ainsi que les contraintes relatives à la présence des bus sur le carrefour (voir la sous-section 3.1.5.2).

Contraintes relatives au calcul de xb_c^b et de $l_{f_b}(T_b)$

Pour chaque bus $b \in B$, et chaque commutation $c \in C'$, on souhaite que $xb_c^b = 1$ si et seulement si $T_b \in [t_c, t_{c+1}[$. Ceci est garanti par les contraintes suivantes :

$$\forall b \in B,$$

$$\sum_{c \in C} xb_c^b = 1$$

$$\forall c \in C, xb_c^b \leq 1 - \frac{T_b - t_{c+1}}{M}$$

$$xb_c^b \leq 1 - \frac{t_c - T_b}{M}$$

$$xb_c^b \in \{0, 1\}$$

où M est un majorant de $\{|T_b - t_c| / b \in B, c \in C\}$.

Le calcul de $l_{f_b}(T_b)$, c'est-à-dire de la file d'attente devant le bus, est donné par les contraintes suivantes :

$$\forall b \in B,$$

$$\begin{aligned} l_{f_b}(T_b) &\geq \sum_{c \in C} [l_{c,f_b} + QA_{f_b} \times (T_b - t_c) + (x_{c,f_b} - 1) \times M] x b_c^b \\ l_{f_b}(T_b) &\geq \sum_{c \in C} [l_{c,f_b} + (QA_{f_b} - QS_{f_b}) \times (T_b - t_c) + (-x_{c,f_b}) \times M] x b_c^b \\ l_{f_b}(T_b) &\geq 0 \end{aligned}$$

où M est un majorant de la longueur des files d'attentes sur le tronçon f_b pendant l'horizon de temps H .

Pour le cas général, non détaillé ici, on utilise le même raisonnement qu'en 3.4.4.4.

Contraintes additionnelles pour les bus

Il s'agit de prendre en compte les contraintes liées à la présence des bus sur leur tronçon (voir sous-section 3.1.5.2). Ces contraintes étant relativement techniques à écrire, nous avons reporté leur formulation en annexe A (4.3).

3.5 Comparaison des modèles “par états” et “par dates”

Dans cette section, on compare, de façon qualitative les deux modèles, en considérant plusieurs caractéristiques.

3.5.1 Hypothèses de travail, représentation de l'environnement

Comme nous l'avons vu dans les premières sections de ce chapitre, les deux modèles sont prévus pour s'insérer dans un même système, qui fonctionne à partir d'une modélisation de l'environnement bien défini et commun aux deux modèles. Les hypothèses

faites au niveau de la modélisation du trafic, du fonctionnement du carrefour et des données disponibles en temps réel sont les mêmes, mis à part les débits d'arrivées, qui potentiellement peuvent être plus finement pris en compte dans le modèle par état. Si l'on fixe les débits d'arrivées sur H pour le modèle par état, on peut dire que les deux modèles ont exactement la même représentation de l'environnement, et les mêmes instances à traiter. Seuls les objets et la manière de procéder pour y parvenir diffèrent.

3.5.2 Estimation du nombre de variables et de contraintes pour le carrefour C

La nature des variables principales est la différence fondamentale entre les deux modèles. Leurs noms parlent d'eux-mêmes : dans un cas il s'agit de déterminer des dates de commutation, dans l'autre de déterminer les différents états des feux, seconde par seconde. Il semble que le modèle par dates soit plus "économe" en variables et en contraintes.

Pour le modèle par états, le nombre de variables principales est de l'ordre de $H \times N_F$ variables 0-1 (les variables $e_{s,f}$).

En ce qui concerne le carrefour C , le nombre total de variables est de l'ordre de 1200 dont environ 900 sont binaires, et le nombre de contraintes est de l'ordre de 7000.

Pour le modèle par dates, le nombre de variables principales est de l'ordre de :

- $N_c \times N_F$ variables 0-1 (les variables $x_{c,f}$)
- N_c variables entières non 0-1 (les variables t_c)

Pour le cas particulier du carrefour C , le nombre total de variables est de l'ordre de 800, dont environ 600 sont binaires, et le nombre de contraintes est de l'ordre de 2500.

Le grand nombre de variables par rapport au nombre de variables principales est en particulier dû à la linéarisation du critère TA .

3.5.3 Formulation des critères

Les critères de nombre d'arrêts et d'erreur totale de consigne pour le bus, représentent exactement la même chose dans un modèle comme dans l'autre. Cependant la linéarisation du critère TA dans le modèle par dates n'est pas une linéarisation exacte :

on approxime TA . Plus exactement, dans le modèle par dates, on le surestime. Les critères TA des deux modèles ne sont donc pas exactement les mêmes.

Enfin, le tableau 3.1 présente la nature des différents critères dans leurs modélisations respectives.

	TA	NA	ECB
Modèle par états	linéaire	linéaire -	linéaire -
Modèle par dates	quadratique	linéaire -	quadratique +

TABLE 3.1 – Nature des différents critères dans leur modélisations respectives

“Quadratique +” indique que le critère est quadratique mais que la linéarisation n’a pas posé de difficulté pour obtenir une linéarisation exacte.

“Linéaire -” indique que le critère est originellement presque linéaire, et qu’il a fallu ajouter un nombre raisonnable de variables pour obtenir une expression complètement linéaire.

“Linéaire -” indique que le critère est originellement presque linéaire, mais qu’il a fallu tout de même rajouter un nombre significatif de variables pour obtenir une expression complètement linéaire.

3.5.4 Extension sur une vision régionale (quelques carrefours)

Le modèle par états peut théoriquement être “étendu” à quelques carrefours. On entend par là qu’une modélisation du problème de régulation sur un ensemble de carrefours plutôt que sur un seul (vision régionale) est fortement envisageable en s’appuyant sur le modèle par états, tout en gardant la même problématique. En effet, la différence principale avec un carrefour isolé est que les arrivées véhicules peuvent être des variables *du temps*, et non pas des “constantes évolutives”, pour pouvoir correspondre à des départs d’autres carrefours proches.

Dans le modèle par état, les arrivées sont prises en compte à chaque pas de temps s , pour un feu f , principalement au niveau de la file d’attente $l_{s,f}$, sous la forme :

$$\begin{aligned}
l_{s,f} &\geq l_{s-1,f} + QA_{s,f} - QS_{s,f} \times e_{s,f} \\
l_{s,f} &\geq 0
\end{aligned}$$

Dans ce modèle, $QA_{s,f}$ est une constante qui représente le débit des arrivées, mais peut aussi être vu comme représentant le nombre de véhicules qui arrivent à l'instant s sur le tronçon f . On comprend bien qu'il n'y a qu'un pas à faire pour prendre en compte le fait que ces arrivées dépendent éventuellement d'un autre carrefour. En effet, en remplaçant $QA_{s,f}$ par une variable $a_{s,f}$, qui dépendrait des départs de l'autre carrefour (on peut utiliser le modèle de dispersion de Robertson (1969) par exemple), on ne change pas la nature linéaire des contraintes.

Cependant, le terme $QA_{s,f}$ intervient aussi multiplicativement dans ce modèle (voir sous-section 3.3.4.2). Comme il s'agit d'une multiplication par une variable 0-1, il est techniquement possible d'obtenir une linéarisation exacte, mais au prix d'ajouts d'un nombre significatif de variables et de contraintes.

Pour le modèle par date, une telle vision régionale semble plus difficile. Tout le problème vient du fait que, dans ce modèle, on n'a pas une représentation utilisant des "pas de temps", chaque pas de temps pouvant être partagé par plusieurs carrefours, mais une représentation utilisant des numérotations de dates de commutations propres à chaque carrefour. En effet, il est difficile d'écrire le lien qui existe entre les quantités mises en jeu pour tel *numéro* de commutation d'un carrefour, et tel autre *numéro* de commutation d'un carrefour adjacent : on ne sait même pas quel numéro des deux correspondra à la première commutation dans l'horizon de temps qu'on s'est donné.

De plus, dans la modélisation par dates, les arrivées véhicules sont représentées par les débits d'arrivées QA_f , $f \in F$, intervenant multiplicativement dans les expressions des files d'attentes (sous-section 3.4.4.3). Malheureusement il s'agit de multiplier QA_f à des variables continues, ce qui, techniquement, est difficile à prendre en compte.

Bien sûr, il est toujours possible de réaliser successivement des optimisations locales sur chaque carrefour en prenant en compte les résultats obtenus par les autres. Ce genre de procédé est évidemment moins attractif que la réalisation d'optimisations globales.

3.6 Modèle hybride

Les considérations précédentes conduisent à envisager un modèle, dit modèle hybride, utilisant simultanément les deux types de formulation. Intuitivement, pour deux programmes linéaires qui modélisent le même problème, celui ne comportant que des expressions simples (nombre de variables et de contraintes impliquées) est susceptible d'être plus performant qu'un programme linéaire possédant des expressions plus complexes. C'est à partir de cette idée que le modèle hybride est né, dans lequel, pour modéliser une même *contrainte* ou un même *critère*, seul l'expression la plus simple parmi les deux modèles est retenue. Des tests effectués sur quelques instances ont par ailleurs permis de confirmer le choix de chaque expression.

Ce procédé n'est possible qu'en autorisant l'utilisation conjointe des deux types de variables que sont les états des feux par unité de temps et les dates de commutation des feux. Il est alors nécessaire de lier ces variables principales pour que les états des feux, contrôlés par les variables $e_{s,f}$ correspondent bien aux dates de commutations, contrôlées par les variables t_c .

3.6.1 Modification des variables par dates

En ce qui concerne les contraintes d'antagonisme, la formulation par états semble particulièrement attrayante par rapport à la formulation par dates. Il n'est alors plus nécessaire de raisonner sur l'ensemble des commutations possibles sans distinction de feu, puisqu'on n'a plus le problème des contraintes disjonctives (voir le modèle par dates, section 3.4.1). On peut alors considérer les variables $t_{c,f}$, correspondant à la $c^{\text{ième}}$ commutation *du feu* f , $f \in F$, $c \in C_f$, où C_f représente l'ensemble des commutations possibles sur l'horizon de temps H pour le feu f , en lieu et place des variables t_c utilisées pour le modèle par date seul. Ce remplacement permet d'éviter le recours aux variables $x_{c,f}$ introduites pour le modèle par dates seul.

Ainsi, les contraintes relatives aux durées minimales et maximales de vert et de rouge s'expriment simplement, pour chaque $f \in F$:

Si $c \in C_f$ correspond à une commutation au vert,

$$\begin{aligned}
t_{c+1,f} - t_{c,f} &\geq D_V^{min} \\
t_{c+1,f} - t_{c,f} &\leq D_V^{max}
\end{aligned}$$

Si $c \in C_f$ correspond à une commutation au rouge,

$$\begin{aligned}
t_{c+1,f} - t_{c,f} &\geq D_R^{min} \\
t_{c+1,f} - t_{c,f} &\leq D_R^{max}
\end{aligned}$$

En réalité, il faut aussi prendre en compte le fait que pour un feu f , sur l'ensemble des commutations possibles C_f , toutes ne vont peut-être pas être utilisées sur l'horizon de temps H , car on ne peut pas prévoir combien de commutations il y aura pour un feu donné dans l'horizon de temps.

Contrairement au cas du modèle par dates, on ne peut plus utiliser le fait que deux commutations peuvent avoir lieu à la même date afin de réguler naturellement le nombre de commutation. Des variables 0-1 supplémentaires attachées aux dates de commutation doivent donc être ajoutées pour pallier cela. Ces variables valent 1 si et seulement si la date $t_{c,f}$ est réellement utilisée. Tout le programme linéaire représentant le modèle par dates doit être modifié en conséquence pour ne considérer $t_{c,f}$ que si la commutation est réellement utilisée sur H . En particulier, si une date n'est pas utilisée, les contraintes minimum de rouge ou de vert sont alors invalidées, même si la date de commutation immédiatement précédente est utilisée.

Toutes ces considérations, qu'on ne développera pas ici par souci de simplicité, font que les contraintes ci-dessus sont en réalité un peu plus complexes à écrire.

3.6.2 Liaisons entre les dates de commutation des feux et les états des feux.

On considère ici pour simplifier que toutes les commutations sont utilisées pendant l'horizon de temps H . Introduisons les variables de liaison $vl_{s,c,f}$ qui, pour chaque feu $f \in F$, valent 1 si $s \leq t_{c,f}$ et valent 0 sinon, $s \in S$, $c \in C_f$.

Pour cela, on écrit les contraintes suivantes.

$$\begin{aligned} vl_{s,c,f} &\leq 1 + \frac{1}{H} (t_{c,f} - s) \\ vl_{s,c,f} &\geq \frac{1}{H} (s - t_{c,f} + 1) \end{aligned}$$

De plus, il faut lier ces variables avec les variables d'état $e_{s,f}$. Or, on a, pour deux commutations successives c et $c + 1$, et ce par définition de $vl_{s,c,f}$:

$$\begin{aligned} vl_{s,c+1,f} - vl_{s,c,f} &= 1, \text{ si } t_{c,f} \leq s < t_{c+1,f} \\ vl_{s,c+1,f} - vl_{s,c,f} &= 0, \text{ sinon} \end{aligned}$$

Donc il suffit de poser, en contraintes :

$$e_{s,f} = \sum_{c \in C_f^V} (vl_{s,c+1,f} - vl_{s,c,f}), \quad f \in F$$

où $C_f^V \subseteq C_f$ représente l'ensemble des commutations au vert de C_f .

3.6.3 Choix de formulations

Dans cette sous-section nous évoquons les choix de formulation effectués en fonction de la simplicité de l'écriture du programme linéaire correspondant au modèle hybride.

3.6.3.1 Contraintes

Comme il a déjà été évoqué dans la section 3.6.1, avec les nouvelles variables de dates de commutation $t_{c,f}$, les contraintes relatives aux durées sont beaucoup plus simples que celles utilisées dans le modèle par états. Nous avons donc opté pour ces formulations.

De plus, les contraintes relatives aux antagonismes sont également plus simples dans le modèle par états que dans le modèle par dates et s'imposent naturellement comme formulation adéquate. Avec les nouvelles variables de dates de commutation $t_{c,f}$, il serait, tout comme avec les variables précédentes t_c , très coûteux d'essayer de prendre en compte les antagonismes. En effet, contrairement aux contraintes de type "durée", on ne peut écrire de relation linéaire entre deux variables $t_{c,f}$ et $t_{c',f'}$ concernant deux

feux antagonistes f et f' que si l'on connaît déjà l'ordre (temporel) entre les deux variables. Autrement dit, avec des antagonismes gérés *via* les variables $t_{c,f}$, on aurait le même problème qu'avec le modèle par dates seul (contraintes disjonctives). Alors que la formulation *via* le modèle par états ne pose aucune difficulté.

Enfin, certaines des expressions relatives au modèle par états tout comme certaines expressions relatives au modèle par dates nécessitent respectivement l'utilisation des variables $l_{s,f}$ et $l_{c,f}$ qui sont respectivement les variables représentant longueurs de files pour le modèle par états et pour le modèle par dates. Nous conservons donc les deux formulations permettant de les calculer, qui sont dans les deux cas des contraintes très simples.

3.6.3.2 Critères

Pour les choix de formulation de critère, nous pouvons nous référer au tableau 3.1 (section 3.5.3) qui donne la nature ainsi qu'une idée de la complexité de chaque critère en fonction de la formulation possible (par états ou par dates).

En ce qui concerne le critère TA , la formulation par états étant linéaire et la formulation par dates quadratique, la formulation la plus avantageuse est clairement la formulation par états.

Pour le critère NA , on a le choix entre une formulation faisant intervenir peu de variables (par dates) et une formulation faisant intervenir beaucoup de variables (par états). Le choix, ici encore, est simple : la formulation par dates est indiquée.

Enfin, pour le critère ECB , le choix semble plus discutable. Après différents tests réalisés sur un jeu d'instances aléatoires, le choix de la formulation par dates a été retenu.

3.6.4 Comparaison en terme de nombre de variables et de contraintes, avec les autres modèles

3.6.4.1 Variables

Le nombre de variables principales du modèle hybride est naturellement plus grand que pour les autres modèles puisqu'on utilise simultanément les deux types de variables principales possibles.

Variables principales Le modèle par états possède $N_F \times H$ variables principales, toutes binaires. Le modèle par dates en possède quant à lui $N_F \times N_C$ continue et $N_F \times N_C$ binaires, soit $2 \times N_F \times N_C$ variables.

Les variables principales pour le modèle hybride sont les variables $e_{s,f}$ et $t_{c,f}$, $f \in F$, $s \in S$ et $c \in C_f$, où C_f est le nombre maximum de commutations pour le feu f . On a donc $N_F \times H + \sum_{f \in F} N_{C_f}$ variables. De plus, le modèle hybride utilise des variables de liaisons $vl_{s,c,f}$ entre ses variables principales, dont le nombre s'élève à : $H \times \sum_{f \in F} N_{C_f}$, que l'on compte ici comme variables principales également.

Variables secondaires Le choix de formulations très simples pour la plupart des contraintes du modèle hybride a pour effet de diminuer significativement le nombre de variables secondaires, qui peuvent être utilisées en très grand nombre dans des contraintes complexes, par rapport aux modèles initiaux.

Conclusion sur les variables Finalement, le faible nombre de variables secondaires pour le modèle hybride compense le nombre élevé de variables principales (dont la cause est principalement le grand nombre de variables $vl_{s,c,f}$), ce qui est confirmé par le tableau 3.2 : le nombre total de variables du modèle hybride est comparable au nombre total de variables du modèle par dates, qui est le plus faible en taille des deux modèles initiaux. De plus, on note un nombre de variables binaires plus faible.

3.6.4.2 Contraintes

Pour chaque type de contraintes présentes dans les modèles initiaux, la formulation la moins coûteuse en nombre de contraintes est choisie, ce qui implique qu'elles sont moins importantes en nombre.

Cependant, le nombre de contraintes de liaisons, tout comme le nombre de variables de liaisons, est assez élevé, puisqu'il est de l'ordre de $2 \times H \times \sum_{f \in F} N_{C_f}$.

3.6.4.3 Estimations sur le carrefour C

Pour le carrefour C , on a les estimations suivantes (tableau 3.2), avec $H = 60$, $N_C = 8$ et $N_{C_f} = 4$, $f \in F$.

	Variables binaires	Variables (nombre total)	Contraintes
Modèle par états	900	1200	7000
Modèle par dates	600	800	2500
Modèle hybride	500	800	3750

TABLE 3.2 – Estimation du nombre de variables et de contraintes sur le carrefour C

Pour un nombre de commutations par feu raisonnable, la taille du modèle hybride se rapproche de la taille du modèle par dates, qui est le plus petit des deux modèles initiaux, et est sensiblement plus faible que pour le modèle par états.

3.7 Formulation de deux types de méthodes interactives pour la régulation de trafic

Dans cette section, on aborde la manière de faire évoluer les paramètres préférentiels introduits en section 3.2.3.

Les critères utilisés, TA , NA et ECB sont des critères qui ont tous une signification facilement appréciable par chacun. Il est donc plus naturel de considérer des méthodes interactives basées sur la *méthode du point de référence* (voir la sous-section 1.2.4.3) dans ce contexte. Cependant, on ne s'interdit pas de pouvoir éventuellement faire varier aussi le paramètre λ de la fonction d'agrégation correspondante (ce qui sera effectivement le cas dans la sous-section 3.7.2).

Deux types de méthodes interactives sont proposées. Le premier type de méthode (sous-section 3.7.1) peut s'appliquer au cas où le décideur n'a pas d'expertise particulière sur les valeurs que peuvent prendre les critères. Le second type de méthode (sous-section 3.7.2) s'appuie sur les connaissances d'un décideur qui serait en mesure de formuler des règles générales portant sur l'évolution de la solution courante dans l'espace des critères.

3.7.1 Cas où le décideur n'a pas d'expertise particulière sur les valeurs des critères, mais sur d'autres indicateurs

3.7.1.1 Problématique

Dans la majorité des cas, l'expertise des exploitants repose essentiellement sur les deux indicateurs suivants, pour évaluer la situation de trafic :

- les débits d’arrivées
- les remontées de files

Pour un tronçon donné, on dit qu’il y a une remontée de file, à un instant donné, si la longueur de la file d’attente à cet instant est supérieure ou égale à la longueur du tronçon. Pour éviter tout phénomène d’oscillation brutale, on considère en réalité qu’il y a remontée de file seulement si la longueur de file est supérieure ou égale à la longueur du tronçon pendant un certain temps $D_{remontee}$.

On peut écrire quantitativement ces indicateurs, par exemple, de la manière suivante :

- un “taux d’arrivées” τ_A qui évolue linéairement en fonction de la somme des débits d’arrivées, qui vaut 0 si tous les débits sont à leur minimum, et 1 s’ils sont tous à leur maximum
- le nombre de remontées de files à un instant donné

On suppose que l’on connaît, pour chaque tronçon f , un débit maximum dba_f^{max} des arrivées. Le taux d’arrivées, à l’instant t s’écrit alors :

$$\tau_A(t) = \frac{\sum_{f \in F} dba_f(t)}{\sum_{f \in F} dba_f^{max}}$$

où pour tout $f \in F$, $dba_f(t)$ est le débit d’arrivées sur le tronçon f à l’instant t .

En pratique, les débits maximum possibles sur un tronçon peuvent être difficiles à déterminer. Pour garantir que $\tau_A(t) \in [0, 1]$ à tout instant, on transforme l’expression du taux des arrivées en $\tau_A(t) = \min \left\{ \frac{\sum_{f \in F} dba_f(t)}{\sum_{f \in F} dba_f^{max}}, 1 \right\}$.

Le but de cette section est de proposer un mécanisme de régulation basé sur les indicateurs que sont le taux d’arrivées et les remontées de files, en essayant de respecter les exigences fournies en sous-section 3.7.1.2.

3.7.1.2 Exigences sur le mécanisme de régulation

La politique *générale* de régulation consiste à “donner une priorité” à un ou plusieurs critères en fonction de la situation du trafic. Cette politique peut se construire sur les principes suivants :

1. Lorsque la situation de trafic est dense ou avec des remontées de files (situation S_{TA}), la minimisation du critère TA est prépondérante, en particulier pour ne pas laisser la situation aboutir à un blocage du carrefour, ce qui pourrait influencer négativement sur l'ensemble des critères.
2. Lorsque la situation de trafic est fluide sans remontée de file (situation $\overline{S_{TA}}$), on privilégie la minimisation du critère qui reflète plutôt une préoccupation environnementale NA (nombre d'arrêts) ou du critère "transports en commun" ECB .

3.7.1.3 Représentation de la politique générale de régulation souhaitée par un profil évolutif, en fonction de la situation du trafic

Pour satisfaire les exigences 1 et 2 du mécanisme de régulation, la politique de régulation à une itération donnée va consister à favoriser la minimisation du critère TA dans la situation S_{TA} , et à favoriser la minimisation des critères NA et ECB dans la situation $\overline{S_{TA}}$.

Pour ce faire, nous allons considérer un profil évolutif z_{profil} dans l'espace des critères, qui représentera la politique de régulation à une itération donnée.

Dans un premier temps, identifions des profils qui correspondraient aux situations extrêmes S_{TA} et $\overline{S_{TA}}$:

- Dans le premier cas, on souhaiterait que z soit "le plus proche possible" de $z_{S_{TA}} = (z_{TA}^*, (z_*)_{NA}, (z_*)_{ECB})$, z^* et z_* étant respectivement les points idéal et anti-idéal.
- Dans le deuxième cas, on souhaiterait que z soit "le plus proche possible" de $z_{\overline{S_{TA}}} = ((z_*)_{TA}, z_{NA}^*, z_{ECB}^*)$.

La politique de régulation, dans les deux cas, peut donc être représentée par un profil z_{profil} de l'espace des critères, égal à $z_{S_{TA}}$ ou à $z_{\overline{S_{TA}}}$. Rappelons que, dans notre contexte, l'instance évolue dans le temps. Le point idéal et le point anti-idéal évoluent donc également dans le temps. Par la suite, nous nous intéressons à une approche où l'on considère non pas les points idéal et anti-idéal évolutifs, mais deux points m et M correspondant respectivement au point idéal dans le cas où l'instance est la plus favorable possible (le trafic est très fluide, il n'y a pas de bus prévu, les tronçons sont vides) et au point anti-idéal dans le cas où l'instance est la plus défavorable possible (le trafic est très dense, il y a plusieurs bus prévus, le carrefour est saturé) . Ainsi $z_{S_{TA}} = (m_{TA}, M_{NA}, M_{ECB})$ et $z_{\overline{S_{TA}}} = (M_{TA}, m_{NA}, m_{ECB})$. Ces considérations ont deux avantages : on n'est pas obligé

de réviser les profils extrêmes au cours de la procédure, et nous verrons que les exigences sur le mécanisme de régulation (sous-section 3.7.1.2) sont alors respectées relativement naturellement.

Nous proposons de traiter les cas intermédiaires de la manière suivante : s'il n'y a pas de remontée de file, $z_{profil} = \tau_A \times z_{STA} + (1 - \tau_A) z_{\overline{STA}}$, où τ_A est le taux des arrivées à l'instant où l'on souhaite réviser z_{profil} , qui représente le profil souhaité à cet instant. Pour "lisser" un peu la politique, τ_A peut éventuellement être une moyenne de $\tau_A(t)$, sur $t \in \{t_0 - \Delta t, \dots, t_0\}$, avec t_0 l'instant présent, et Δt la longueur de la période (passée) sur laquelle on veut lisser.

De plus, s'il existe au moins une remontée de file, alors $z_{profil} = z_{STA}$. Ce cas revient à forcer τ_A à 1, on pourra donc toujours considérer que z_{profil} peut s'exprimer en fonction de τ_A .

Étant donné que nous travaillons à partir d'une fonction d'agrégation à base de point de référence, il est naturel de considérer $\bar{z} = z_{profil}$ (méthode du point de référence) pour représenter la politique de régulation par des paramètres préférentiels, à partir du profil z_{profil} .

λ est choisi constant, tel que $\lambda_j = \frac{1}{M_j - m_j}$, $j \in \{TA, NA, ECB\}$, ce qui jouera un rôle dans le respect des exigences sur le mécanisme de régulation (sous-section 3.7.1.2). Nous montrerons en effet dans la sous-section 3.7.1.4 que la procédure respecte ces exigences. Une illustration de la méthode se trouve en figure 3.7.

Autres variantes possibles

La méthode proposée ne doit pas être considérée comme un élément fixé des modèles. Il existe une multitude de règles possibles, qui peuvent éventuellement être adaptées à la vision qu'à le décideur de la politique de régulation en général. On pourrait par exemple vouloir prendre en compte le *nombre* de remontées de files, plutôt qu'un indicateur de présence de remontée de file. Le terme correspondant au nombre de remontées de files dans le positionnement du profil n'est cependant pas simple à établir.

On pourrait également vouloir prendre en compte la solution courante \hat{z} , pour définir les profils extrêmes et par conséquent le profil courant, en révisant les profils extrêmes :

$$- z_{STA} = (m_{TA}, \hat{z}_{NA}, \hat{z}_{ECB})$$

$$- \overline{z_{STA}} = (\hat{z}_{TA}, m_{NA}, m_{ECB})$$

Dans le cas où le carrefour admet des instances qui modifient très fortement et régulièrement la position des points idéal et anti-idéal, m et M deviennent rapidement inadaptés. On pourrait alors souhaiter prendre en compte z^* et z_* , au niveau du profil courant et/ou au niveau du paramètre λ de la fonction d'agrégation : pour ce faire, il suffirait de réviser régulièrement $z_{STA} = (z_{TA}^*, (z_*)_{NA}, (z_*)_{ECB})$ et $\overline{z_{STA}} = ((z_*)_{TA}, z_{NA}^*, z_{ECB}^*)$ et/ou λ en imposant $\lambda_j = \frac{1}{z_j^* - (z_*)_j}$, $j \in \{TA, NA, ECB\}$ à chaque révision, ce qui suppose un calcul préalable de z^* et de z_* à chaque fois, ce qui impose trois optimisations supplémentaires. Néanmoins on peut bien sûr choisir un taux de révision inférieur à une révision par itération. Si, de plus, on souhaite aussi prendre en compte la solution courante, on peut alors calculer :

$$\begin{aligned} - z_{STA} &= (z_{TA}^*, \hat{z}_{NA}, \hat{z}_{ECB}) \\ - \overline{z_{STA}} &= (\hat{z}_{TA}, z_{NA}^*, z_{ECB}^*) \end{aligned}$$

Enfin, il est également possible de fixer $\bar{z} \simeq z^*$ et de prendre en compte le profil au niveau de la direction de projection en choisissant par exemple $\lambda_j = \frac{1}{(z_{profil})_j - \bar{z}_j}$, $j \in \{TA, NA, ECB\}$, ce qui correspondrait à une procédure de type Tchebychev (voir sous-section 1.2.4.3).

Conclusion sur les variantes

Une multitude de procédures sont donc possibles à partir de l'idée de profil défini dans cette section. Cependant, toutes les variantes énoncées ici ne présentent pas les mêmes garanties en ce qui concerne le respect des exigences du mécanisme de régulation, ce qui les rend moins attrayantes. Dans le cas où λ est quelconque (mais fixe), une étude générale du respect de ces exigences, en considérant une hypothèse supplémentaire, est néanmoins proposée en annexe B (4.3).

Dans la sous-section suivante, nous nous intéressons à la variante initiale, pour montrer qu'elle respecte les exigences.

3.7.1.4 Cohérence de la méthode par rapport aux exigences du mécanisme de régulation

Traduction des exigences sur le mécanisme de régulation (sous-section 3.7.1.2) en terme de valeurs prises sur les différents critères de la solution courante, d'une itération à une autre :

- A partir des définitions de profils données en sous-section 3.7.1.3, nous désirons que :
- une modification du profil courant, d'une itération à la suivante, vers un profil plus proche de S_{TA} implique que la nouvelle solution fournie soit au moins aussi bonne que la solution courante sur le critère TA .
 - réciproquement, une modification du profil courant, d'une itération à la suivante, vers un profil plus proche de $\overline{S_{TA}}$ implique que la nouvelle solution fournie soit au moins aussi bonne sur un des critères NA ou ECB .

Dans le paragraphe suivant, nous considérons le cas général à p critères, où le critère 1 joue le rôle du critère TA et les $p - 1$ autres critères jouent un rôle identique aux critères NA et ECB . Le profil correspondant à S_{TA} est alors représenté par le point $z_{S_1} = (m_1, M_2, M_3, \dots, M_p)$ et le profil correspondant à $\overline{S_{TA}}$ est représenté par le point $z_{\overline{S_1}} = (M_1, m_2, m_3, \dots, m_p)$.

Cadre et notations :

Dans cette sous-section, on se place dans le cadre d'un problème d'optimisation multi-objectif (P) (défini en sous-section 1.2.1) dont l'instance est fixée, avec p critères tous à *minimiser* et qui prennent uniquement des valeurs entières. On utilisera en particulier les notations relatives à l'optimisation multi-objectif introduites en section 1.2.

On rappelle que X désigne l'ensemble des solutions admissibles, $Z = \mathbb{R}^p$ et $Z_X \subseteq \mathbb{Z}^p$ est l'image de X dans Z .

On considère les deux fonctions d'agrégation à base de point de référence suivantes, introduites en sous-section 1.2.4.2 :

$$g_{\bar{z}, \lambda} : Z_X \rightarrow \mathbb{R}$$

$$z \mapsto \max_{j=1, \dots, p} \lambda_j (z_j - \bar{z}_j)$$

Et,

$$f_{\bar{z},\lambda} : Z_X \rightarrow \mathbb{R}$$

$$z \mapsto \max_{j=1,\dots,p} \lambda_j (z_j - \bar{z}_j) + \rho \sum_{j=1}^p \lambda_j (z_j - \bar{z}_j)$$

où $\bar{z} \in Z$, $\lambda \in (\mathbb{R}_+^*)^p$ et ρ sont les paramètres définis en sous-section 1.2.4.2 (ρ est fixé).

La fonction d'agrégation $h_{\bar{z},\lambda}$ représentera indifféremment l'une des deux fonctions $f_{\bar{z},\lambda}$ ou $g_{\bar{z},\lambda}$.

Le but de cette section est de montrer que, dans le cadre de l'utilisation de la fonction $h_{\bar{z},\lambda}$, si le point de référence \bar{z} correspond à un profil tel que défini en 3.7.1.3, c'est-à-dire si $\bar{z} = \bar{z}(\tau) = \tau \times z_{S_1} + (1 - \tau) z_{\bar{S}_1}$ avec $z_{S_1} = (m_1, M_2, M_3, \dots, M_p)$, $z_{\bar{S}_1} = (M_1, m_2, m_3, \dots, m_p)$, et $\tau \in [0, 1]$, et que λ est tel que $\lambda_j = \frac{1}{M_j - m_j}$, $j \in \{1, \dots, p\}$, alors pour deux paramètres τ et τ' :

- si $\tau' \geq \tau$, c'est-à-dire que $\bar{z}(\tau')$ est “plus proche” du profil z_{S_1} que ne l'est $\bar{z}(\tau)$, la solution obtenue avec le point de référence $\bar{z}(\tau')$ sera meilleure que celle obtenue avec $\bar{z}(\tau)$ sur le critère 1. Cette propriété correspond au premier point de l'exigence du mécanisme de régulation (3.7.1.4),
- si $\tau \geq \tau'$, c'est-à-dire que $\bar{z}(\tau')$ est “plus proche” du profil $z_{\bar{S}_1}$ que ne l'est $\bar{z}(\tau)$, la solution obtenue avec le point de référence $\bar{z}(\tau')$ sera meilleure que celle obtenue avec $\bar{z}(\tau)$ sur au moins un des critères $j \geq 2$. Cette propriété correspond au deuxième point de l'exigence du mécanisme de régulation (3.7.1.4).

Un des critères (ici le critère 1) joue donc un rôle particulier. Il s'agit du critère dont on veut pouvoir “contrôler” la valeur en priorité, en fonction du paramètre τ .

Les deux points correspondent au cas général des exigences sur le mécanisme de régulation de la sous-section 3.7.1.2.

La suite de la section a pour but de montrer que ces exigences sont respectées, ce qui fera l'objet de la Proposition 2.

Pour \bar{z} et λ donné, la génération d'un point non dominé en utilisant la fonction $f_{\bar{z},\lambda}$ peut être vue comme l'opération de projection du point \bar{z} sur l'ensemble $ND(Z_X)$ relativement à la droite $\mathbf{u}\mathbb{R}$ avec $\mathbf{u} = \left(\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_p}\right)$. Par abus de langage on utilisera le terme projection pour une fonction de type $h_{\bar{z},\lambda}$ en général.

Définition 1. *Cône de projection pour une fonction d'agrégation de type $h_{\bar{z},\lambda}$*

Soit \hat{z} une solution du problème multi-objectif (P) (avec Z_X discret), générée en utilisant une fonction d'agrégation de type $h_{\lambda,\bar{z}}$. On appelle *cône de projection* et on note $C_{\bar{z},\lambda}$ l'ensemble $\{z \in Z, h_{\bar{z},\lambda}(z) = h_{\bar{z},\lambda}(\hat{z})\}$.

Remarque 1. Bien qu'il soit commode d'utiliser \hat{z} pour définir $C_{\bar{z},\lambda}$, à partir du moment où l'on considère que $ND(Z_X)$ est fixé, $C_{\bar{z},\lambda}$ ne dépend en réalité que de \bar{z} et de λ . La notation $C_{\bar{z},\lambda}$ est donc cohérente.

Définition 2. Soit $C_{\bar{z},\lambda}$ un cône de projection donné, on appelle *sommet* de ce cône et on note $s = (s_1, \dots, s_p)$ le point de $C_{\bar{z},\lambda}$ tel que $\arg \max_{j \in \{1, \dots, p\}} \lambda_j (s_j - \hat{z}_j) = \{1, \dots, p\}$.

Définition 3. *Droite de projection pour une fonction d'agrégation de type $h_{\lambda,\bar{z}}$*

Soit $C_{\bar{z},\lambda}$ un cône de projection donné et s son sommet, on appelle *droite de projection* et on note $D_{\bar{z},\lambda}$ la droite (\bar{z}, s) . En remarquant que $D_{\bar{z},\lambda}$ est également dirigée par le vecteur \mathbf{u} tel que $u_j = \frac{1}{\lambda_j}$, $j = 1, \dots, p$, on notera également cette droite $D_{\mathbf{u}}$.

Lemme 2. Soit $\bar{z} = (\bar{z}_1, \dots, \bar{z}_p)$ et $\bar{z}' = (\bar{z}'_1, \dots, \bar{z}'_p)$ deux points de Z et $\lambda \in (\mathbb{R}_+^*)^p$ tels que :

$$\begin{aligned} \bar{z}'_1 &< \bar{z}_1 \\ \forall j \in \{2, \dots, p\}, \bar{z}'_j &= \bar{z}_j \end{aligned}$$

Soit respectivement \hat{z} et \hat{z}' des solutions du problème multi-objectif (P) en utilisant respectivement comme fonction d'agrégation : $h_{\bar{z},\lambda}$ et $h_{\bar{z}',\lambda}$.

On a $\hat{z}'_1 \leq \hat{z}_1$.

Réciproquement : si

$$\begin{aligned} \bar{z}'_1 &> \bar{z}_1 \\ \forall j \in \{2, \dots, p\}, \bar{z}'_j &= \bar{z}_j \end{aligned}$$

Alors il existe $j \in \{2, \dots, p\}$ tel que $\hat{z}'_j \leq \hat{z}_j$.

Démonstration. On suppose dans un premier temps que $h_{\bar{z},\lambda} = g_{\bar{z},\lambda}$ et $h_{\bar{z}',\lambda} = g_{\bar{z}',\lambda}$. Dans l'espace des critères que l'on considère comme un espace affine, \bar{z}' est l'image de \bar{z} par la translation de vecteur $\mathbf{t} = (\bar{z}'_1 - \bar{z}_1, 0, \dots, 0)$. La droite de projection $D_{\bar{z},\lambda}$ passant par le point s , $D_{\bar{z}',\lambda}$ passe donc par le point $s'' = s - \mathbf{t} = (s_1 - (\bar{z}'_1 - \bar{z}_1), s_2, \dots, s_p)$.

s'' appartient clairement à $C_{\bar{z},\lambda}$. Donc $s'' \leq s'$ car sinon s' dominerait s et alors on aurait \hat{z}' domine s ce qui est impossible. On a donc $s'_j \geq s_j$, $j = 2, \dots, p$. D'où $s'_1 < s_1$ car sinon c'est cette fois s qui dominerait s' ce qui est également impossible pour des raisons similaires.

Alors, tout point z de $C_{\bar{z},\lambda}$ tel que $z_1 < s'_1$ domine s' . Donc $\hat{z}_1 \geq s'_1$ car sinon \hat{z} dominerait s' . Comme $s'_1 \geq \hat{z}'_1$, par transitivité on a $\hat{z}'_1 \leq \hat{z}_1$.

Supposons maintenant que $h_{\bar{z},\lambda} = f_{\bar{z},\lambda}$ et $h_{\bar{z}',\lambda} = f_{\bar{z}',\lambda}$. Comme \hat{z} et \hat{z}' sont aussi solutions avec $g_{\bar{z},\lambda}$ et $g_{\bar{z}',\lambda}$, on se ramène au cas précédent et on a $\hat{z}'_1 \leq \hat{z}_1$.

La réciproque est évidente : si $\bar{z}'_1 > \bar{z}_1$ alors on se ramène au cas précédent pour avoir $\hat{z}_1 \leq \hat{z}'_1$. Donc on ne peut pas avoir $\hat{z}_j < \hat{z}'_j$, $j = 2, \dots, p$ car \hat{z} dominerait \hat{z}' . Donc il existe $j \in \{2, \dots, p\}$ tel que $\hat{z}_j \geq \hat{z}'_j$. \square

Proposition 2. *On suppose que le point de référence correspond à un profil tel que défini en 3.7.1.3.*

Soit $\tau, \tau' \in [0, 1]$, $z_{S_1} = (m_1, M_2, M_3, \dots, M_p)$, $z_{\bar{S}_1} = (M_1, m_2, m_3, \dots, m_p)$, $\bar{z}(\tau) = \tau \times z_{S_1} + (1 - \tau) z_{\bar{S}_1}$ et $\bar{z}(\tau') = \tau' \times z_{S_1} + (1 - \tau') z_{\bar{S}_1}$. Enfin soit λ tel que $\lambda_j = \frac{1}{M_j - m_j}$, $j = 1, \dots, p$.

Soit \hat{z} et \hat{z}' des solutions du problème multi-objectif (P) en utilisant respectivement comme fonction d'agrégation : $h_{\bar{z}(\tau),\lambda}$ et $h_{\bar{z}(\tau'),\lambda}$. Les deux assertions suivantes sont vraies :

- si $\tau' > \tau$, alors $\hat{z}'_1 \leq \hat{z}_1$,
- si $\tau' < \tau$ alors $\exists j \in \{2, \dots, p\}$, $\hat{z}'_j \leq \hat{z}_j$.

Démonstration. Étant donné la définition de λ , chaque droite de projection est dirigée par le vecteur $\mathbf{u} = (M - m)$.

On considère l'application linéaire bijective :

$$\begin{aligned} [0, 1] &\rightarrow Z \\ \tau &\mapsto \bar{z}^{\mathbf{u}}(\tau) = \bar{z}(\tau) - \tau \mathbf{u} \end{aligned}$$

Pour chaque $\tau \in [0, 1]$, $\bar{z}(\tau)$ et $\bar{z}^{\mathbf{u}}(\tau)$ ont la même droite de projection $D_{\bar{z}(\tau), \lambda} = \mathbb{R}\mathbf{u}$.
 Pour $\tau = 0$ nous avons

$$\begin{aligned}\bar{z}^{\mathbf{u}}(0) &= \bar{z}(0) \\ &= z_{\bar{S}_1} \\ &= (M_1, m_2, \dots, m_p)\end{aligned}$$

et pour $\tau = 1$, nous avons

$$\begin{aligned}\bar{z}^{\mathbf{u}}(1) &= \bar{z}(1) - \mathbf{u} \\ &= z_{S_1} - \mathbf{u} \\ &= (2m_1 - M_1, m_2, \dots, m_p)\end{aligned}$$

Le segment $[z_{S_1}, z_{\bar{S}_1}]$ est donc en bijection avec le segment $[m', z_{\bar{S}_1}]$ où $m' = (2m_1 - M_1, m_2, \dots, m_p)$. A chaque point de référence $\bar{z}(\tau)$ appartenant à $[z_{S_1}, z_{\bar{S}_1}]$, il existe donc un unique point de référence $\bar{z}^{\mathbf{u}}(\tau)$ appartenant à $[m', z_{\bar{S}_1}]$, qui de plus a la propriété d'appartenir à la même droite de projection $D_{\bar{z}(\tau), \lambda}$ de $\bar{z}(\tau)$, donc de générer exactement les mêmes solutions que $\bar{z}(\tau)$ pour le problème (P).

On en déduit que l'on peut ramener l'étude au cas où le point de référence se déplace sur le segment $[m', z_{\bar{S}_1}]$ qui a la particularité d'appartenir à une droite dirigée par un vecteur de coordonnées nulles pour $j \geq 2$. De plus, étant donné que $\bar{z}^{\mathbf{u}}(0) = z_{\bar{S}_1}$ et que $\bar{z}^{\mathbf{u}}(1) = m'$, nous avons que $(\tau > \tau') \Leftrightarrow (z^{\mathbf{u}}(\tau))_1 < (z^{\mathbf{u}}(\tau'))_1$ et $(\tau < \tau') \Leftrightarrow (z^{\mathbf{u}}(\tau))_1 > (z^{\mathbf{u}}(\tau'))_1$. On peut donc se ramener au cas du Lemme 2. Une illustration est proposée figure 3.7. □

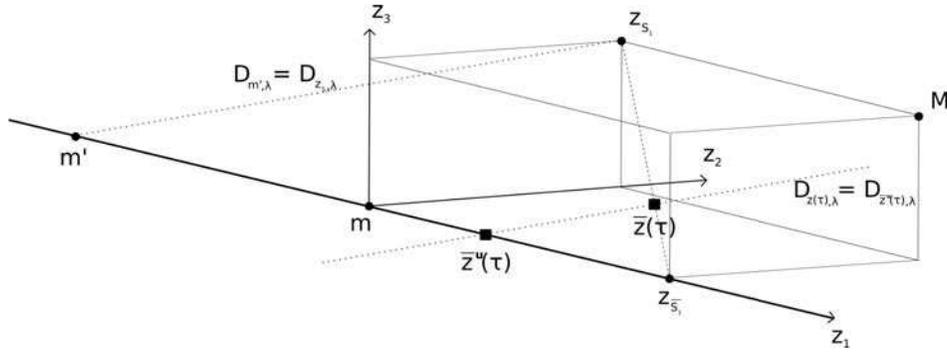


FIGURE 3.7 – Illustration de la méthode interactive proposée dans le cas où le décideur n’a pas d’expertise. Le point de référence $\bar{z}(\tau)$ se déplace sur le segment $[z_{S_1}, z_{S_1}^-]$ en fonction de la situation du trafic, ce qui revient, par le choix de λ , à déplacer un point de référence “équivalent” $\bar{z}^u(\tau)$ (qui génère les mêmes solutions) sur le segment $[m', z_{S_1}^-]$, permettant ainsi à la méthode d’avoir les propriétés souhaitées (voir la sous-section 3.7.1.2).

3.7.2 Cas où le décideur dispose de connaissances sur les valeurs des critères

3.7.2.1 Problématique

On se place ici dans le cas où le décideur dispose d’un historique du carrefour concerné, lui permettant d’appréhender la situation de trafic à *travers les trois critères TA, NA et ECB*. D’autres indicateurs peuvent également être utilisés, comme par exemple une remontée de file importante sur un seul tronçon, une gêne sur le carrefour, etc.

Il semble que ce genre d’expertise, sur un carrefour, soit assez peu disponible en pratique pour le moment. Ceci est sans doute lié au fait que les exploitants n’ont pas encore l’habitude de travailler avec des critères précis d’optimisation. Cependant, on ne peut pas exclure, qu’un jour, une telle expertise soit disponible.

Une approche connue et due à Nakayama et Sawaragi (1983) est construite en considérant que le décideur peut être interrogé directement à chaque étape sur la valeur des critères de la solution courante. Pour une étape donnée, le décideur prend connaissance de la solution courante. Il classe alors les valeurs des critères en trois catégories : la

première est celle des valeurs à améliorer, la seconde celle des valeurs qu’il juge acceptable de relaxer, et la troisième celle des valeurs qu’il veut garder telles quelles. Le point de référence est alors révisé afin de générer une nouvelle solution se rapprochant des préférences du décideur. La direction de projection est soit fixée, soit dépendante du point de référence.

Dans notre cas, on ne peut pas interroger le décideur à chaque itération. On peut cependant adapter l’approche à une procédure interactive automatisée en ne considérant pas la solution courante elle-même, mais en considérant une *zone* de l’espace des critères dans laquelle se trouve celle-ci. Si le nombre de zones n’est pas trop important, cela permettra d’interroger le décideur en amont, puis de se baser sur les règles d’ “évolution” de zone ainsi construites pour procéder à la régulation du trafic. Ces règles joueront ainsi le rôle de politique générale de régulation.

Une façon simple de découper l’espace des critères en différentes zones est de fixer $q \in \mathbb{N}$ hyperplans distincts par critère, orthogonaux à l’axe du critère considéré, qui définiront ainsi naturellement $(q + 1)^p$ pavés. Ainsi, chaque hyperplan peut être défini simplement par un seul paramètre, qui correspondra à un seuil de préférence fixé par le décideur.

Hypothèses sur le nombre de critères à améliorer et à relaxer

Les trois catégories de critères définies par Nakayama et Sawaragi (1983) (critères à améliorer, critères à relaxer, critères à ne pas modifier) et qui concernent un point non dominé courant, sont facilement transposables au cas où l’on travaille sur des *zones* et non des points, dès lors que l’on peut comparer deux zones par rapport à un critère donné (dans l’exemple qui suivra, on aura un ordre implicite de type *mauvais < moyen < bon*). En effet on peut alors parler, pour une zone donnée, de critère à améliorer, à ne pas modifier, ou à relaxer.

Nous considérons alors que, pour une zone donnée, *l’ensemble des critères à améliorer sera toujours réduit à un critère*. En ce qui concerne l’ensemble des critères dont la valeur peut être relaxée, deux cas, dits “de relaxation”, seront envisagés :

- l’ensemble des critères à relaxer est vide (aucun critère à relaxer)
- l’ensemble des critères à relaxer est un singleton (un seul critère à relaxer)

Les autres critères sont toujours considérés comme à garder tels quels.

Pour *chaque zone* où peut se trouver la solution courante, le travail du décideur lors

des questionnements se ramène donc à déterminer *le critère à améliorer*.

Notons que le premier cas de relaxation est possible parce que nous raisonnons sur des *zones* et non pas sur des points non dominés (sinon il n'y aurait jamais de nouvelle solution trouvée).

Exemple

Précisons la démarche sur un exemple à 9 zones avec 2 critères. Pour chaque critère $j \in \{1, 2\}$, on considère deux *seuils de préférence*, $s_{1j}, s_{2j} \in \mathbb{Z}$, avec $s_{1j} < s_{2j}$. Pour $z \in Z_X$ donné, la valeur z_j , $j \in \{1, 2\}$ est alors considérée comme :

- bonne si $z_j \leq s_{1j}$
- moyenne si $s_{1j} < z_j \leq s_{2j}$
- mauvaise si $z_j > s_{2j}$

Pour déterminer ces seuils, le décideur doit être capable d'apprécier les valeurs des différents critères, ce qui est effectivement le cas dans le cadre de cette sous-section.

On associe alors à ces seuils les zones suivantes :

- zone (bon, bon) : $\{z \in Z / z_1 \leq s_{11}, z_2 \leq s_{12}\}$
- zone (bon, moyen) : $\{z \in Z / z_1 \leq s_{11}, s_{12} < z_2 \leq s_{22}\}$
- zone (bon, mauvais) : $\{z \in Z / z_1 \leq s_{11}, z_2 > s_{22}\}$
- zone (moyen, bon) : $\{z \in Z / s_{12} < z_1 \leq s_{21}, z_2 \leq s_{12}\}$
- zone (moyen, moyen) : $\{z \in Z / s_{12} < z_1 \leq s_{21}, s_{12} < z_2 \leq s_{22}\}$
- zone (moyen, mauvais) : $\{z \in Z / s_{12} < z_1 \leq s_{21}, z_2 > s_{22}\}$
- zone (mauvais, bon) : $\{z \in Z / z_1 > s_{21}, z_2 \leq s_{12}\}$
- zone (mauvais, moyen) : $\{z \in Z / z_1 > s_{21}, s_{12} < z_2 \leq s_{22}\}$
- zone (mauvais, mauvais) : $\{z \in Z / z_1 > s_{21}, z_2 > s_{22}\}$

En ce qui concerne le nombre de zones ainsi construites, si l'on a p critères, avec deux seuils par critères, on a alors 3^p zones de l'espace des critères, qui feront toutes l'objet d'au moins un questionnement posé au décideur : quel critère améliorer si la solution courante se trouve dans la zone en question. Dans le cadre de la thèse, cela fait 27 cas à envisager, ce qui reste encore relativement raisonnable. En revanche, si on inclut un quatrième critère, concernant par exemple les piétons, on arrive déjà à 81 cas.

On considère dans cet exemple que zone courante est la zone (moyen, moyen) et où le critère à améliorer est le critère 1. Tout comme dans l'approche de Nakayama et Sawaragi (1983), la détermination d'une nouvelle solution s'effectue en révisant uniquement le point de référence, la direction de projection étant fixée. Une illustration est donnée figure 3.8.

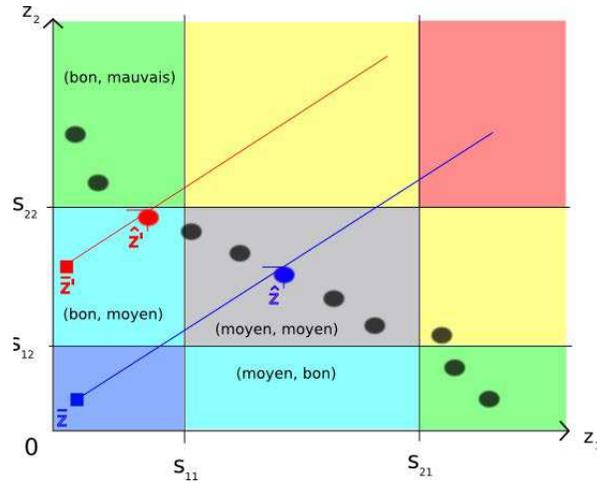


FIGURE 3.8 – Illustration de la méthode interactive avec repositionnement du point de référence, λ fixé, et $p = 2$ critères. La couleur bleue est utilisée pour les éléments de l'itération courante, tandis que la couleur rouge est utilisée pour l'itération suivante. Dans les deux cas, le point de référence est représenté par un carré. Les autres points sont les points non dominés de l'instance.

Description de l'illustration et formulation de la problématique 3.8

Pour chaque critère $j \in \{1, 2\}$, s_{1j} et s_{2j} sont les seuils de préférence. Les différentes zones colorées correspondent donc aux différents cas sur lesquels le décideur a été interrogé.

Le point \bar{z} est le point de référence courant et correspond ici au point idéal. Le point \hat{z} est la solution courante, et se trouve dans la zone (moyen, moyen).

Le décideur, dans ce cas, a indiqué qu'il préférerait améliorer le critère 1 sans dégrader le critère 2, ce qui revient à chercher une solution dans la zone (bon, moyen), dite zone "souhaitée", ou dans toute autre zone qui serait au moins aussi bonne sur chacun des deux critères. Un nouveau point de référence \bar{z}' est choisi afin de générer une nouvelle solution \hat{z}' .

Dans cet exemple, \hat{z}' se trouve bien dans la zone (bon, moyen). Cependant, il ne suffit pas de choisir \bar{z}' dans la zone souhaitée pour que \hat{z}' le soit aussi. Sur la figure 3.8, on peut facilement trouver des positions de \bar{z}' dans la zone souhaitée qui mèneraient soit de nouveau à la zone (moyen, moyen), soit à la zone (bon, mauvais), alors qu'il existe une solution dans la zone (bon, moyen).

La problématique sous-jacente, pour ce cas précis, est de trouver comment déterminer les paramètres préférentiels assurant la génération d'un points non dominé, s'il existe, dans une zone souhaitée ((bon, moyen) dans notre exemple), correspondant à une amélioration de la zone courante sur un critère, ou dans toute autre zone au moins aussi bonne sur chacun des critères. Dans ce contexte, il peut alors s'avérer judicieux de se laisser la liberté d'agir à la fois sur la position du point de référence et sur la direction de projection, afin de pouvoir explorer une zone précise de l'espace des critères.

Ainsi, le but de cette section est de déterminer une procédure qui permette de faire évoluer la zone "courante" (où se trouve la solution courante), en fonction des souhaits du décideur, en s'aidant des paramètres préférentiels de la fonction d'agrégation utilisée (de type $f_{\bar{z},\lambda}$). Le mécanisme de régulation ainsi obtenu devra respecter, dans la mesure du possible, certaines exigences, précisées en sous-section 3.7.2.3. Nous travaillerons dans le cas général de $p \geq 2$ critères et $q \geq 2$ seuils de préférences.

3.7.2.2 Définitions et notations

Définition 4. Soit $q \in \mathbb{N}^*$. Pour chaque critère $j \in \{1, \dots, p\}$, soit $s_{1,j}, s_{2,j}, \dots, s_{q-1,j} \in \mathbb{N}^*$, tels que $s_{i+1,j} \geq s_{i,j} + 1$, $i \in \{1, \dots, q-1\}$, avec $s_{q,j}$ un majorant des valeurs susceptibles d'être prises par z_j , $z \in Z_X$. On appelle ces nombres seuils de préférence. De plus, on pose $s_{0,j} = 0$, $j \in \{1, \dots, p\}$.

Définition 5. Pour chaque $i_1, i_2, \dots, i_p \in \{0, \dots, q-1\}$, soit Z_{i_1, i_2, \dots, i_p} la région de l'espace des critères Z telle que pour chaque point $z \in Z_{i_1, i_2, \dots, i_p}$:

$$\forall j \in \{1, \dots, p\}, s_{i_j, j} < z_j \leq s_{i_j+1, j}$$

Z_{i_1, i_2, \dots, i_p} est appelée la *zone* d'indices i_1, \dots, i_p .

Définition 6. Deux zones qui diffèrent d'une unité sur un seul indice sont dites zones adjacentes.

Définition 7. Soit Z et Z' deux zones d'indices respectifs i_1, \dots, i_p et i'_1, \dots, i'_p .

Si $\forall j \in \{1, \dots, p\}$, $i'_j \leq i_j$, on note $Z' \leq Z$.

Si $\forall j \in \{1, \dots, p\}$, $i'_j \leq i_j$ et $\exists j_0 \in \{1, \dots, p\}$, $i'_{j_0} < i_{j_0}$, on note $Z' < Z$ et on dit que Z' domine Z .

Comme nous avons pu le constater sur une illustration (figure 3.8), obtenir une solution dans une zone bien déterminée semble difficile en ne faisant pas évoluer la direction de projection. Par la suite, on considère donc deux points évolutifs : un point de référence \bar{z} et un point \tilde{z} fixant la direction de projection $\mathbf{u} = \left(\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_p}\right) = (\tilde{z} - \bar{z})$.

Définition 8. Pour tout \bar{z} et \tilde{z} de Z , on désigne par $l_{\bar{z}, \tilde{z}}$ la fonction d'agrégation suivante :

$$\begin{aligned} Z_X &\rightarrow \mathbb{R} \\ z &\rightarrow \max_{j=1, \dots, p} \frac{z_j - \bar{z}_j}{\tilde{z}_j - \bar{z}_j} + \rho \sum_{j=1}^p \frac{z_j - \bar{z}_j}{\tilde{z}_j - \bar{z}_j} \end{aligned}$$

Remarque 2. On a $l_{\bar{z}, \tilde{z}} = f_{\bar{z}, \tilde{\lambda}}$ avec $\tilde{\lambda}$ tel que $\tilde{\lambda}_j = \frac{1}{\tilde{z}_j - \bar{z}_j}$, $j \in \{1, \dots, p\}$.

Remarque 3. Le paramètre ρ de la fonction $l_{\bar{z}, \tilde{z}} = f_{\bar{z}, \tilde{\lambda}}$ est fonction de $\tilde{\lambda}$ (voir la sous-section 1.2.4.2), qui dépend du choix de \bar{z} et \tilde{z} (voir la remarque 2 précédente). Nous supposons que pour \bar{z} et \tilde{z} fixés, ρ est choisi conformément à la Proposition 1, où λ sera pris égal à $\tilde{\lambda}$. Ceci permet en effet de garantir que tout point de $ND(Z_X)$ peut être généré en utilisant la fonction $l_{\bar{z}, \tilde{z}}$, où ρ est choisi de manière adaptée.

3.7.2.3 Exigences sur le mécanisme de régulation

Étant donnée la problématique, on peut avoir les exigences simples suivantes :

- Tout point généré lors d'une optimisation qui vise à l'exploration d'une zone doit se trouver soit dans la zone à explorer (ou dans une zone qui la domine), soit dans la zone courante (ou dans une zone qui la domine).

- Dans le cas où le point nouvellement généré se trouve dans la zone courante, il doit être au moins aussi bon, sur le critère à améliorer, que la solution courante.

3.7.2.4 Mécanisme de régulation

Deux cas de relaxation considérés

Les deux cas de relaxation définis dans la sous-section 3.7.2.1 peuvent être reformulés ainsi dans le cas général :

1. Pour une zone courante donnée (contenant la solution courante), la nouvelle zone souhaitée est une zone adjacente “améliorant” un indice donné (le nouvel indice est alors plus faible). En d’autres termes, le décideur ne souhaite dégrader la position de la zone courante sur aucun indice mais l’améliorer sur un indice.
2. Pour une zone courante donnée, la nouvelle zone souhaitée peut être “moins bonne” sur un indice (d’indice plus élevé). En contrepartie elle sera meilleure sur un indice.

Chacun des deux cas possède ses avantages et ses inconvénients. L’avantage principal du premier cas est qu’il permet de respecter totalement les exigences sur le mécanisme de régulation (sous-section 3.7.2.3), comme il sera montré dans le paragraphe correspondant à ce premier cas de relaxation (3.7.2.4). L’avantage du deuxième cas est sa plus grande souplesse (par rapport au premier cas). Il permet en particulier d’éviter les cas où aucun point non dominé ne se trouve dans une zone adjacente, ce qui aurait pour effet de bloquer momentanément l’évolution de zone. En revanche, les exigences sur le mécanisme de régulation sont mises à mal dans le deuxième cas, comme il sera expliqué dans le paragraphe correspondant (3.7.2.4).

Mécanisme

Le mécanisme consiste à faire évoluer les points \bar{z} et \tilde{z} dans le temps, de manière à ce qu’à une itération donnée, leur position dans l’espace des critères permette la recherche d’une nouvelle solution dans une zone précise, en fonction du cas de relaxation considéré, en essayant de respecter les exigences sur le mécanisme de régulation. La détermination exacte de ces points étant fonction du cas de relaxation, elle dépendra de chaque cas.

Cas 1 : le décideur ne souhaite dégrader la position de la zone courante sur aucun indice

Ce cas est celui dans lequel, à une itération donnée, le décideur n'accepte jamais de dégrader la zone courante sur un de ses indices. L'amélioration de la zone courante va alors consister en l'amélioration sur un seul indice si cela est possible, de la zone courante, les autres restant constants.

La Proposition 3 donne un paramétrage possible permettant de satisfaire ce type de règles tout en respectant les exigences sur le mécanisme de régulation données en sous-section 3.7.2.3. On suppose que l'indice à améliorer est le premier indice, correspondant à une amélioration souhaitée sur le critère 1.

Proposition 3. *Soit \hat{z} une solution du problème multi-objectif (P) avec Z_X discret (voir sous-section 1.2.1).*

Soit \hat{Z} la zone où se trouve \hat{z} , d'indices $\hat{i}_1, \dots, \hat{i}_p$, où $\hat{i}_1 \in \{1, \dots, q-1\}$ et $\hat{i}_j \in \{0, \dots, q-1\}$ pour chaque $j \in \{2, \dots, p\}$, et soit Z' la zone d'indices $\hat{i}_1 - 1, \dots, \hat{i}_p$ où l'on souhaite générer une nouvelle solution. On souhaite donc améliorer le critère 1 en priorité.

Alors, il existe $\bar{z}', \tilde{z}' \in Z$ tels que, si on appelle \hat{z}' une solution de $\min_{z \in Z_X} l_{\bar{z}', \tilde{z}'}(z)$, nous ayons :

$$\hat{z}' \in Z'_D \cup \hat{Z}_D$$

où $Z'_D \leq Z'$ et $\hat{Z}_D \leq \hat{Z}$.

De plus, si $\hat{z}' \in \hat{Z}_D$ alors $\hat{z}'_1 \leq \hat{z}_1$ (ce qui est évidemment également vrai si $\hat{z}' \in Z'_D$).

En effet, les points $\bar{z}', \tilde{z}' \in Z$ respectant les exigences suivantes conviennent :

$$\begin{aligned} \tilde{z}'_1 &= s_{\hat{i}_1, 1} \\ \bar{z}'_1 &= s_{\hat{i}_1 - 1, 1} \end{aligned}$$

et pour tout $j \in \{2, \dots, p\}$:

$$\begin{aligned} \tilde{z}'_j &= s_{\hat{i}_j + 1, j} \\ \bar{z}'_j &< s_{\hat{i}_j + 1, j} \\ \bar{z}'_j &> s_{\hat{i}_j + 1, j} - \frac{s_{\hat{i}_1, 1} - s_{\hat{i}_1 - 1, 1}}{s_{\hat{i}_1 + 1, 1} - s_{\hat{i}_1, 1}} \end{aligned}$$

Démonstration. Soit s' le sommet du cône de projection $C_{\bar{z}, \tilde{\lambda}}$ où $\tilde{\lambda}$ est le vecteur donné

en remarque 2 et $D_{\mathbf{u}}$ la droite de projection dirigée par $\mathbf{u} = \left(\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_p}\right) = \tilde{z}' - \bar{z}'$. Comme $\tilde{z} \in D_{\mathbf{u}}$, qu'il existe un point $\hat{z} \in ND(Z_X)$ dans la zone \hat{Z} , et que, par le choix de \bar{z}' et \tilde{z}' , $\mathbf{u} \in \mathbb{R}_+^p$, alors on a :

- 1) soit $s' \ll \bar{z}$ (cas où s' appartient à la demi-droite $]-\infty \times u, \bar{z}'[$)
- 2) soit $s' \in Z'$ (cas où s' appartient au segment $[\bar{z}', \tilde{z}']$)
- 3) soit $s' > \tilde{z}'$, où \hat{Z}' est la zone d'indices $\hat{i}_1, \hat{i}_2 + 1, \dots, \hat{i}_p + 1$ (cas où s' appartient à la demi-droite $]\tilde{z}', +\infty \times u[$)

Dans le troisième cas, on a nécessairement $s'_1 \leq s_{i_1+1,1}$ car il existe une solution dans la zone \hat{Z} . De plus, il existe $t > 0$ tel que pour chaque $j \in \{1, \dots, p\}$ on ait $s'_j = \tilde{z}'_j + tu_j$. En particulier $s'_1 = \tilde{z}'_1 + tu_1 = s_{i_1,1} + t(\tilde{z}'_1 - \bar{z}'_1)$.

Donc,

$$\begin{aligned} t &= \frac{s'_1 - s_{i_1,1}}{\tilde{z}'_1 - \bar{z}'_1} \\ &\leq \frac{s_{i_1+1,1} - s_{i_1,1}}{\tilde{z}'_1 - \bar{z}'_1} \\ &\leq \frac{s_{i_1+1,1} - s_{i_1,1}}{s_{i_1,1} - s_{i_1-1,1}} \end{aligned}$$

Pour chaque $j \in \{2, \dots, p\}$ on a donc :

$$\begin{aligned} s'_j &= \tilde{z}'_j + tu_j \\ s'_j &< \tilde{z}'_j + \frac{s_{i_1+1,1} - s_{i_1,1}}{s_{i_1,1} - s_{i_1-1,1}} \times \frac{s_{i_1,1} - s_{i_1-1,1}}{s_{i_1+1,1} - s_{i_1,1}} \\ &< s_{i_j+1,j} + 1 \end{aligned}$$

Par conséquent, dans ce troisième cas, la solution générée ne peut appartenir qu'à \hat{Z} . En effet, pour chaque $j \in \{2, \dots, p\}$, $\hat{z}'_j \leq s'_j < s_{i_j+1,j} + 1 \leq s_{i_j+1,j}$ car $\hat{z}' \in \mathbb{Z}^p$ (et rappelons que tous les seuils sont des entiers).

De plus, pour chaque $j \in \{2, \dots, p\}$, $s'_j > s_{i_j+1,j}$, donc $\arg \max_{j \in \{1, \dots, p\}} \frac{\hat{z}'_j - \bar{z}_j}{\tilde{z}'_j - \bar{z}_j} = 1$. Donc $\hat{z}'_1 \leq \hat{z}_1$ car sinon on aurait $\hat{z}_1 < s'_1$ ce qui est impossible.

Enfin, les cas évidents 1 et 2 amenant à une solution dans $Z'_D \leq Z'$, la proposition est vérifiée. \square

La figure 3.9 est une illustration du procédé dans le cas où $p = 2$ et $q = 2$.

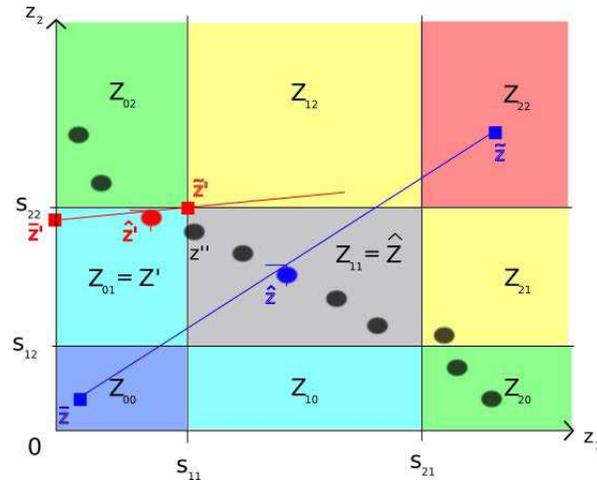


FIGURE 3.9 – Les carrés correspondent aux paramètres préférentiels \bar{z} et \tilde{z} . On souhaite obtenir un point non dominé, s’il existe, dans la zone Z' . Sur l’exemple, un tel point existe (\hat{z}'). La zone courante est \hat{Z} . Si z' n’existait pas, le point généré serait le point z'' , qui appartient à \hat{Z} .

Cas 2 : le décideur accepte de dégrader la position de la zone courante sur un indice

Ce cas est celui dans lequel, à une itération donnée, le décideur accepte de dégrader éventuellement un indice de la zone courante, d’une unité au maximum, pour pouvoir améliorer un autre indice. Les indices restants sont conservés. Il va de soit que si un point non dominé se trouve dans une zone qui améliore l’indice voulu sans dégrader les autres, cette zone sera préférée à la zone dégradée.

Nous sommes donc dans une situation un peu similaire à la précédente, où pour satisfaire les exigences de la sous-section 3.7.2.3 nous avons à explorer deux zones dans un certain ordre (d’abord la zone adjacente souhaitée puis la zone courante).

Soit \hat{Z}_d la zone courante, d’indices $\hat{i}_1, \dots, \hat{i}_p$, où $\hat{i}_1 \in \{1, \dots, q-1\}$ et $\hat{i}_j \in \{0, \dots, q-1\}$ pour chaque $j \in \{2, \dots, p\}$, soit Z'_d la zone d’indices $\hat{i}_1 - 1, \dots, \hat{i}_p$ où l’on souhaite générer une nouvelle solution, et soit Z''_d d’indices $\hat{i}_1 - 1, \hat{i}_2 + 1, \dots, \hat{i}_p$ la zone où l’on souhaite générer une nouvelle solution si Z'_d est vide. On souhaite donc améliorer le critère 1 en priorité, en acceptant de dégrader le critère 2 si cela est nécessaire. On applique la Proposition 3 en considérant que $\hat{Z} = Z''_d$ et $Z' = Z'_d$ mais avec pour objectif une amélioration sur le deuxième indice. Le placement correspondant de \bar{z}' et \tilde{z}' permet alors de garantir que si un point non dominé existe dans Z'_d ou toute zone qui la domine, elle

sera générée en priorité. Si aucun point non dominé n'y figure, alors, si un point non dominé existe dans Z''_d ou toute zone qui la domine, elle sera générée. Une illustration en dimension deux est proposée figure 3.10.

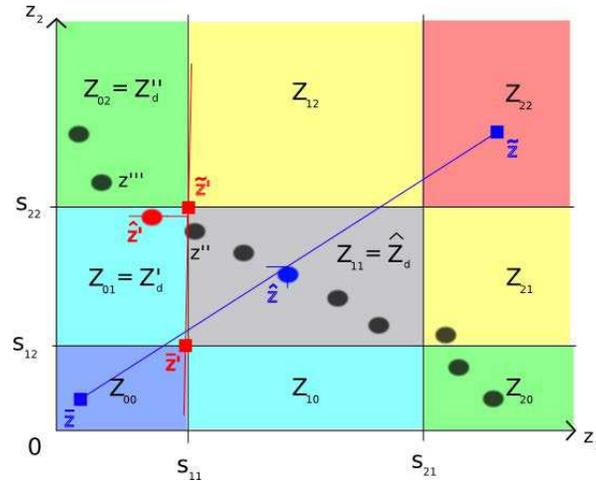


FIGURE 3.10 – On souhaite en priorité obtenir un point non dominé, s’il existe, dans la zone Z'_d . Sur l’exemple, un tel point existe (\hat{z}'). Si \hat{z}' n’existait pas, le point généré serait le point z''' , qui appartient à Z''_d , qui correspond à une zone améliorée sur l’indice i_1 et dégradée sur l’indice i_2 , par rapport à la zone courante \hat{Z}_d .

En revanche, il n’y a aucune garantie que, dans le cas où il n’existe pas de point non dominé dans Z'_d ni dans Z''_d , le point généré appartienne à \hat{Z}_d ou une zone la dominant, comme le montre le contre-exemple de la figure 3.11, ce qui met à mal les exigences. Étant donné la configuration des zones \hat{Z}_d , Z'_d et Z''_d et l’ordre de “priorité” donné sur ces zones pour satisfaire les exigences, il est peu probable qu’il existe un placement de \bar{z}' et \tilde{z}' qui convienne pour respecter ces dernières en une seule itération.

le niveau d'expertise nécessaire à la méthode interactive correspondant à la section 3.7.2 (le décideur possède une expertise), nous avons donc décidé d'adopter la méthode interactive correspondant au cas où le décideur n'a pas d'expertise particulière (section 3.7.1).

Chapitre 4

Comparaison des modèles sur une itération et résultats sur l'utilisation du modèle hybride en temps réel

Dans ce chapitre, nous expérimentons le système élaboré au chapitre 3 sur le carrefour C présenté au chapitre 1 (figure 4.1). Les différents modèles y sont ainsi testés tour à tour, ainsi que la méthode interactive temps réel choisie pour les expérimentations.

La méthode d'optimisation choisie est la méthode Branch&Bound, mise en œuvre par le solveur CPLEX (version 12.2) d'ILOG, sur Intel Core 2 duo CPU 1.8 GHz (2Go RAM).

Nous avons choisi le solveur CPLEX car il s'agit d'un solveur reconnu dans les communautés industrielles ou universitaires, qui sont régulièrement confrontées à des problèmes d'optimisation combinatoire à la fois complexes et de taille importante.

Les options de CPLEX communes à l'ensemble des expérimentations sont les options par défaut, à l'exception des options :

- Special Ordered Set (type 2) utilisée pour le modèle par dates (voir section 3.4).
- Set Priority : les variables principales et les variables représentant les longueurs de files d'attente ont été priorisées.

Les variables représentant des dates de commutation dans le modèle par dates ou dans le modèle hybride peuvent être choisies discrètes ou continues. Néanmoins, étant donné que le gain de performance observé pour le modèle par dates lorsque les dates de commutation sont choisies entières (voir la sous-section 3.4.5.1), et que le même phénomène a

été observé pour le modèle hybride, nous avons choisi de fixer les dates de commutation comme étant entières.

Les débits utilisés dans les expérimentations sont fournis avec une précision de 0.01 véhicules par unité de temps. Les longueurs de files d'attente sont alors imposées discrètes avec un *pas* de 0.01, comme suggéré en sous-section 3.4.5.1 pour le modèle par dates (suggestion qui s'applique aussi au modèle hybride).

Ces choix seront appliqués dans tout ce chapitre.

Dans un premier temps (section 4.1), nous comparons les trois modèles de la thèse entre eux. Il s'agit de comparer les performances des modèles en terme de qualité des solutions obtenues et de temps consommé pour fournir les solutions. Les expérimentations ont lieu sur une seule itération, c'est-à-dire que les instances et les paramètres préférentiels utilisés sont fixés.

Dans un second temps (section 4.2), après avoir identifié le modèle le plus performant au regard de la section 4.1, nous utilisons le système muni de ce modèle sur plusieurs itérations (les instances évoluent dans le temps), pour réaliser des expérimentations portant sur des cas simulés de trafic, en utilisant la procédure interactive temps réel introduite en sous-section 3.7.1. Nous comparons les résultats à ceux obtenus avec un système semi-adaptatif.

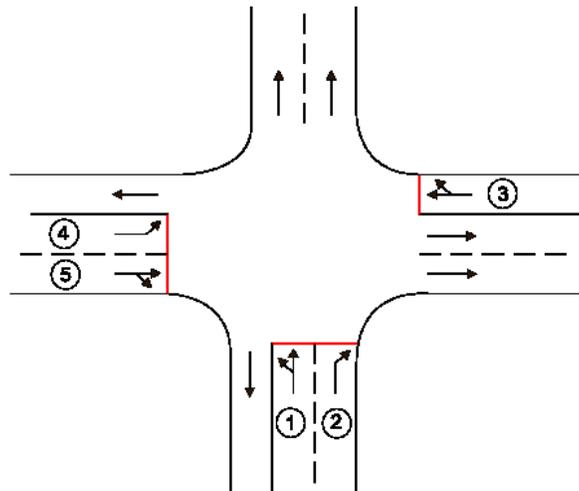


FIGURE 4.1 – Carrefour *C*

4.1 Comparaison des modèles sur une seule itération

On compare ici les performances des différents modèles de la thèse. Pour ce faire, il est nécessaire de ne pas se placer dans un cadre “itératif”, c’est-à-dire où les modèles sont appliqués sur une période de temps donnée sur un carrefour, et où les instances futures dépendent des décisions prises au moment présent. On veut, en effet, pouvoir comparer les modèles sur des instances qui sont exactement les mêmes, et avec la même fonction objectif.

4.1.1 Méthodologie

Pour chacun des trois modèles (par états, par dates et hybride), nous avons réalisé un ensemble de mesures, relevées sur un horizon de temps $H = 60$ secondes.

4.1.1.1 Paramètres

- Le temps alloué à l’optimisation : 10 secondes, 30 secondes ou 10 minutes. Le dernier temps est destiné à essayer d’atteindre l’optimum.
- Le type de trafics : fluide, dense ou asymétrique. Pour un trafic plutôt fluide, nous avons pris des débits d’arrivée tous égaux à 0.1 véhicule par seconde, et pour un trafic plutôt dense 0.3 véh/s. Pour le trafic asymétrique, trois tronçons sont à 0.1 véh/s (les trois premiers en respectant la numérotation du carrefour C), les deux autres à 0.3 véh/s.
- La présence de bus ou non : soit aucun bus n’est prévu, soit trois bus sont prévus. Dans ce dernier cas, un bus est prévu sur la ligne 1 à $t = 15$ s, un autre sur la ligne 2 à $t = 35$ s, et encore un autre sur la ligne 4 à $t = 10$ s.
- La politique d’optimisation : le point de référence est placé à différentes positions significatives de l’espace des critères. Trois valeurs par critères sont considérées, ce qui représente 27 cas de positionnement.
- Les files d’attente sont toutes nulles à $t = 0$.

4.1.1.2 Mesures

Nous observons les mesures suivantes :

- la solution dans l’espace des critères, retournée par le solveur
- la valeur de la fonction objectif, retournée par le solveur

Comme les valeurs des critères renvoyées par le solveur ne sont pas exactes, à cause des approximations dans la modélisation et des linéarisations, nous avons également recalculé les valeurs des critères *a posteriori*, à partir de la solution obtenue (états des feux ou dates de commutations). Nous fournissons donc également :

- la solution dans l’espace des critères, recalculée
- la valeur de la fonction objectif, recalculée

Les valeurs des critères retournées par le solveur sont appelées *valeurs calculées*. Elles traduisent la “vision qu’a le solveur des critères”.

Les *valeurs recalculées* correspondent aux valeurs que prendront “réellement” les critères si on applique les décisions issues de la résolution par le solveur, *en supposant que tout se passe conformément à la modélisation* du trafic sur laquelle on s’est appuyé pour bâtir les modèles (section 3.1).

4.1.1.3 Hypothèses supplémentaires

- En ce qui concerne le nombre de commutations maximales N_C , pour le modèle par dates, il est apparu expérimentalement que fixer $N_C = 8$ était un bon compromis pour le carrefour étudié. Ce nombre peut sembler faible puisqu’il s’agit du nombre maximum de commutations sur l’ensemble des feux pendant l’horizon de temps. Cependant, rappelons que, pour ce modèle, lorsque deux feux distincts commutent au même instant, il n’y a qu’une commutation de comptée. Or, compte tenu des antagonismes, il est très fréquent que plusieurs feux commutent simultanément lors d’une même itération. Le nombre de commutations par feu est donc en général du même ordre de grandeur que pour le modèle hybride, où le nombre maximal de commutations par feu N_{C_f} sera pris égal à 4, $f \in F$.
- En ce qui concerne les demandes relatives aux bus, on a pris $TAB_b^0 = 0$ pour chaque bus $b \in B$ (on veut minimiser l’attente de chaque bus).

4.1.2 Résultats

Comme nous l'avons vu en sous-section 4.1.1.1, un certain nombre de paramètres peuvent varier et contribuent à la richesse des instances étudiées. C'est le cas en particulier de la position du point de référence.

Pour simplifier l'analyse, les valeurs présentées dans cette sous-section sont des valeurs moyennes calculées sur des ensembles d'instances "similaires". On a ainsi regroupé les instances de type de trafic identique. Les trois premières lignes correspondent respectivement aux instances où le trafic est de type "fluide", "dense" et "asymétrique". La dernière ligne ("général") correspond à la moyenne sur l'ensemble des instances.

Dans une première étude, on a cherché à étudier les écarts relatifs, exprimés en pourcentage, qui peuvent exister entre les valeurs calculées et les valeurs recalculées. Dans le cas où aucune solution réalisable n'est obtenue, on compte un écart de 100% sur tous les critères. Si de grands écarts existent entre les valeurs calculées et les valeurs recalculées cela peut donc traduire deux phénomènes (qui peuvent être combinés) : l'erreur sur le critère (lié à l'approximation faite) est grande, ou bien il y a une insuffisance significative de solution réalisable.

Dans une seconde étude, on a cherché à comparer les modèles sur la base de la *valeur recalculée* de leur fonction objectif. Les fonctions objectifs recalculées sont obtenues en agrégeant *a posteriori* les valeurs recalculées des critères.

Utiliser des valeurs recalculées est nécessaire pour pouvoir comparer les modèles, puisque ce sont elles qui traduisent l'"efficacité" de ces derniers par rapport à une même instance, et relativement à la même modélisation du trafic.

4.1.2.1 Comparaison des valeurs calculées et des valeurs recalculées

Dans les différents tableaux présentés, le critère ECB n'apparaît pas car l'écart relatif est toujours expérimentalement nul ou négligeable. L'expression de ce critère utilise un très petit nombre de variables à cause du faible nombre de bus généralement mis en jeu. Les différentes linéarisations utilisées pour ce critère comprennent par conséquent beaucoup moins de variables que celles utilisées pour les autres critères. Ceci permet d'expliquer la qualité des écarts relatifs obtenus pour ce critère.

Modèle par états

En ce qui concerne le modèle par états, les résultats du tableau 4.1 sont sans appel quant à la pertinence du modèle sur des optimisations à temps limité (10 secondes). Les très mauvais résultats obtenus ont principalement pour cause le fait qu'aucune solution réalisable n'est trouvée dans la majorité des cas (99 cas sur 162).

On peut constater, en revanche, sur les tableaux 4.2 et 4.3 que sur des optimisations avec une limite de temps suffisamment grande, les valeurs calculées coïncident presque toujours avec les valeurs recalculées.

Ceci est lié, en ce qui concerne *TA*, à sa formulation linéaire dans ce modèle. En ce qui concerne *NA*, on aurait pu, en revanche, s'attendre à des résultats plus médiocres étant donné le nombre de variables utilisées dans la linéarisation, par rapport à la formulation dans le modèle par dates.

	TA	NA
Fluide	90,74%	90,74%
Dense	25,92%	25,97%
Asymétrique	66.66%	66.68%
Général	61,11%	61,13%

TABLE 4.1 – Écarts relatifs pour les critères TA et NA du modèle par états, entre les valeurs calculées et les valeurs recalculées, avec une limite de temps de 10 secondes pour l'optimisation

	TA	NA
Fluide	0%	0,03%
Dense	0%	0,05%
Asymétrique	0%	0,01%
Général	0%	0,03%

TABLE 4.2 – Écarts relatifs pour les critères TA et NA du modèle par états, entre les valeurs retournées par le programme et les valeurs recalculées, avec une limite de temps de 30 secondes pour l'optimisation

	TA	NA
Fluide	0%	0%
Dense	0%	0%
Asymétrique	0%	0%
Général	0%	0%

TABLE 4.3 – Écarts relatifs pour les critères TA et NA du modèle par états, entre les valeurs retournées par le programme et les valeurs recalculées, avec une limite de temps de 10 minutes pour l’optimisation

Modèle par dates

On peut constater, sur les tableaux suivants (4.4, 4.5 et 4.6), la convergence des valeurs des critères calculées vers les valeurs des critères recalculées au fur et à mesure que le temps alloué à l’optimisation augmente. D’autre part, précisons que les valeurs calculées sont toujours plus grandes que valeurs recalculées. Le modèle surestime donc les valeurs des critères.

La convergence des valeurs calculées vers les valeurs recalculées s’explique par le fait que les variables utilisées pour linéariser les critères qui le nécessitaient peuvent être nombreuses et sont dans tous les cas non prioritaires par rapport aux variables liées directement aux décisions portant sur les couleurs des feux. Pour qu’elles jouent complètement leur rôle, il faut donc attendre un certain délai. Avant cela, l’approximation est d’autant plus mauvaise que les variables d’approximation n’ont pas eu le temps d’être affectées optimalement.

En ce qui concerne la surestimation des valeurs du programme, il s’agit d’effets relatifs à la formulation même des critères. Le lecteur pourra se référer à la sous-section 3.4.5. Le fait que les valeurs des critères soient surestimées plutôt que sous-estimées par le solveur n’a *a priori* pas d’impact sur l’"efficacité" des modèles, puisque ce qui est important est l’exactitude des modèles. Il s’agit simplement d’une constatation expérimentale.

On voit clairement, en particulier sur le tableau 4.4 (optimisation à 10 secondes), que les plus grandes différences relatives concernent le critère TA. Il s’agit du critère, pour le modèle par dates, qui a fait l’objet de la linéarisation la plus poussée, avec un certain nombre d’approximations, et de contraintes supplémentaires.

	TA	NA
Fluide	12,83%	1,92%
Dense	12,56%	1,85%
Asymétrique	12,09%	2,23%
Général	12,49%	2%

TABLE 4.4 – Écarts relatifs pour les critères TA et NA du modèle par dates, entre les valeurs retournées par le programme et les valeurs recalculées, avec une limite de temps de 10 secondes pour l’optimisation

	TA	NA
Fluide	3,36%	0,83%
Dense	2,94%	0,72%
Asymétrique	2,58%	0,66%
Général	2,96%	0,73%

TABLE 4.5 – Écarts relatifs pour les critères TA et NA du modèle par dates, entre les valeurs retournées par le programme et les valeurs recalculées, avec une limite de temps de 30 secondes pour l’optimisation

	TA	NA
Fluide	2,70%	0,74%
Dense	2,24%	0,64%
Asymétrique	2,25%	0,53%
Général	2,39%	0,63%

TABLE 4.6 – Écarts relatifs pour les critères TA et NA du modèle par dates, entre les valeurs retournées par le programme et les valeurs recalculées, avec une limite de temps de 10 minutes pour l’optimisation

Modèle hybride

Pour ce qui est du modèle hybride, les valeurs calculées et les valeurs recalculées coïncident parfaitement, sauf un cas à 10 secondes, et un cas à 30 secondes. On a donc amélioré ce point grâce au modèle hybride.

En ce qui concerne le critère TA , comme on utilise la formulation du modèle par états, on retrouve les 0% d'écart des tableaux 4.1, 4.2 et 4.3 correspondant au modèle par état (voir le paragraphe correspondant - 4.1.2.1).

Pour le critère NA , qui a une formulation légèrement différente dans le modèle hybride et dans le modèle par dates, à cause en particulier de la gestion simplifiée des antagonismes dans le modèle hybride, on obtient également des écarts de 0%.

4.1.2.2 Comparaison des modèles

Les trois tableaux présentés dans cette sous-section contiennent les valeurs moyennes des fonctions objectif recalculées des trois modèles, les temps limites d'optimisation étant respectivement de 10 secondes, 30 secondes et 10 minutes.

Rappelons que plus les valeurs sont faibles, meilleur est le résultat, puisque nous cherchons à minimiser notre fonction d'agrégation (sur les mêmes instances).

	États	Dates	Hybride
Fluide	0.2603	0.1762	0.1553
Dense	1.1706	1.1790	1.1388
Asymétrique	0.6773	0.4486	0.4297
Général	0.9123	0.5966	0.5747

TABLE 4.7 – Comparaison des moyennes des fonctions objectifs recalculées des trois modèles, pour une limite de temps de 10 secondes

	États	Dates	Hybride
Fluide	0.2085	0.1754	0.1502
Dense	1.1270	1.1670	1.1262
Asymétrique	0.4356	0.4455	0.4118
Général	0.5807	0.5918	0.5534

TABLE 4.8 – Comparaison des moyennes des fonctions objectifs recalculées des trois modèles, pour une limite de temps de 30 secondes

	États	Dates	Hybride
Fluide	0.1477	0.1695	0.1470
Dense	1.1260	1.1632	1.1258
Asymétrique	0.4104	0.4398	0.4096
Général	0.5518	0.5835	0.5515

TABLE 4.9 – Comparaison des moyennes des fonctions objectifs recalculées des trois modèles, pour une limite de temps de 10 minutes

Comparaison du modèle par états et du modèle par dates (fonction objectif recalculée)

Les résultats montrent que le modèle par dates est sensiblement meilleur que le modèle par états, dans le cas où l’optimisation doit être faite rapidement (tableau 4.7 - 10 secondes). En revanche, il devient moins bon que ce dernier lorsque les limites de temps augmentent (tableaux 4.8 et 4.9 - 30 secondes et 10 minutes). On peut expliquer ces différents résultats en deux points :

1. Le fait que le modèle par dates comporte globalement moins de variables 0-1 que le modèle par états, et en particulier beaucoup moins de variables “principales”, c’est-à-dire les variables représentant directement les états des feux, peut expliquer les bonnes solutions réalisables trouvées rapidement par ce modèle.
2. Le fait que les résultats retournés par le modèle par dates soient relativement stables, pour des optimisations plus longues, peut être lié au nombre de commutations explicitement imposé au modèle par dates sur l’horizon de temps (déterminé expérimentalement - voir section 4.1.1). Ceci peut également être attribué à l’approximation faite pour le critère TA dans le modèle par dates.

Finalement, on peut dire que, dans la limite des instances testées et des paramètres choisis, le modèle par état permet une optimisation de qualité moyenne au début puis régulièrement meilleure au fur et à mesure que le temps d’optimisation imparti augmente, tandis que le modèle par dates permet une optimisation “rapidement assez bonne” mais sans progrès significatif lorsqu’on augmente un peu le temps imparti.

Modèle hybride (fonction objectif recalculée)

On peut facilement observer sur les différents tableaux de cette sous-section que le modèle hybride fournit des solutions “meilleures” que les deux autres modèles, du point de vue de la fonction objectif, même si l’avance est parfois minime. Le modèle hybride permet donc, comme on le souhaitait, d’être au moins aussi bon que le meilleur des deux modèles dans une situation donnée. C’est donc ce modèle qui sera utilisé pour les expérimentations de la section 4.2, à savoir les expérimentations sur plusieurs itérations.

4.2 Optimisation sur plusieurs itérations

Cette section est dédiée à l’utilisation du modèle hybride sur plusieurs itérations. On entend par là une utilisation sur une période de temps donnée sur un carrefour dont le trafic est simulé au cours du temps et les décisions sur la couleur des feux du modèle hybride appliquées. Ces simulations, contrairement aux précédentes, permettent :

- d’utiliser une politique générale de régulation (sous-section 3.7)
- de comparer le modèle utilisé à un système de régulation ressemblant à l’existant

4.2.1 Fréquence des optimisations

Des résultats obtenus en section 4.1, on retient, en outre, qu’une itération peut largement dépasser la seconde en temps d’exécution. Ce qui signifie qu’un fonctionnement du système à la fréquence d’une optimisation par seconde, soit la fréquence à laquelle sont révisées les informations provenant des capteurs, est *a priori* exclu. Cependant, on peut demander à CPLEX de trouver une solution réalisable en un temps fixé. De plus, on peut éventuellement passer au solveur la dernière solution obtenue, afin qu’il parte de cette solution (qui demeure partielle) pour trouver rapidement une solution réalisable. Il s’agit de l’option Set Vector, qui a été utilisée lors de ces expérimentations sur plusieurs itérations.

Cependant, limiter la durée des optimisations à un temps très court présente plusieurs inconvénients majeurs :

- On prend le risque que le solveur ne trouve pas de solution réalisable en moins d’une seconde. De plus, en pratique, ce dernier dispose de moins d’une seconde

pour effectuer l'optimisation, car il faut prendre en compte toutes les autres opérations effectuées dans le pas de temps (modification de l'instance, écriture sur disque, etc.)

- Même si le solveur trouve une solution réalisable, à partir de la dernière solution fournie, il y a de fortes chances qu'en moins d'une seconde il ne modifie pas la "structure" de la solution courante, ce qui limite fortement les possibilités.
- Plus généralement, on ne profite pas suffisamment selon nous, en moins d'une seconde, des capacités du solveur à résoudre des problèmes de programmation linéaire mixte complexes.

Ceci nous conduit à considérer que l'optimisation pourrait se faire toutes les $d_{it} \geq 2$ secondes plutôt que toutes les secondes, en faisant l'hypothèse que la situation n'aura pas beaucoup évolué en d_{it} secondes. On peut par exemple anticiper l'optimisation en lançant le solveur pour un plan de feux commençant à $t = d_{it}$ secondes plutôt que $t = 0$, ce qui laisse effectivement d_{it} secondes au solveur pour conclure.

La figure 4.2 montre, avec une représentation par dates de commutation (du type utilisé par modèle hybride), les différentes parties du plan de feux courant considérées comme fixées (parties grisées) selon que l'on optimise toutes les secondes (en haut de la figure), ou toutes les d_{it} secondes (en bas de la figure).

Si on optimise toutes les secondes, on fixe uniquement la première seconde, c'est-à-dire qu'on ne pourra modifier ultérieurement cette région. C'est la partie qui est déjà transmise au contrôleur pour application sur le carrefour.

En revanche, si on optimise toutes les d_{it} secondes, cette partie fixe (grisée) augmente, comme on peut le voir sur la partie basse de la figure.

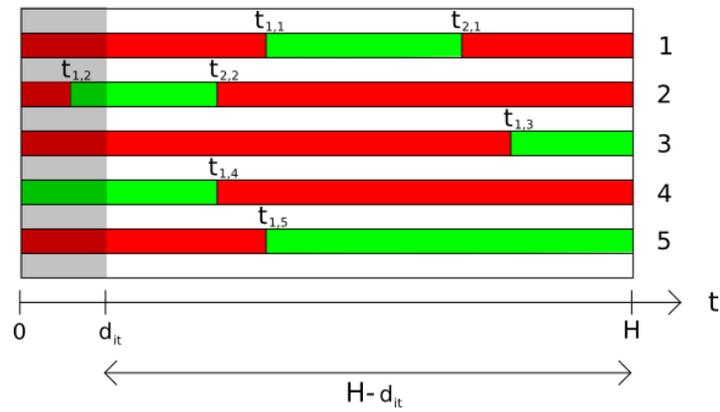
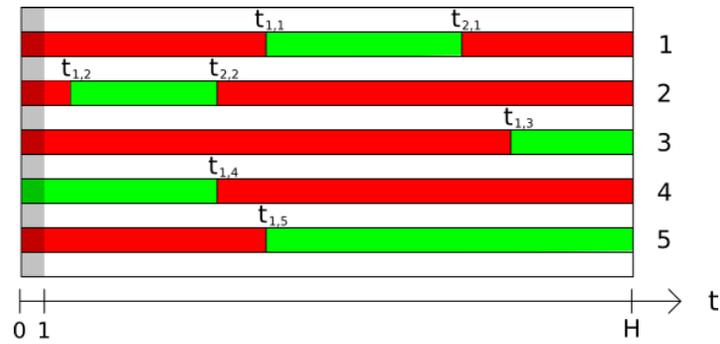


FIGURE 4.2 – Représentation d’une solution pour une optimisation à une seconde et pour une optimisation à d_{it} secondes

Ceci laisse au solveur le temps nécessaire pour au moins trouver une solution réalisable. L’inconvénient est que l’instance à considérer (longueur des files, débits d’arrivées) doit être calculée à la date $t = d_{it}$ et non pas $t = 1$, ce qui suppose que le trafic se comporte conformément au modèle pendant d_{it} unités de temps.

Le solveur, lui, considère l’intervalle de temps $[d_{it}, H]$ pour établir le plan de feux, en se servant des décisions prises sur la partie fixée courante pour calculer la nouvelle instance à cet instant. L’hypothèse fondamentale consiste à supposer que la situation du trafic ne change pas, en général, brutalement en d_{it} secondes, pour d_{it} représentant un faible nombre de secondes.

Enfin, une conséquence bénéfique sur les performances est la réduction de l’horizon de temps vue par le solveur, qui passe de H à $H - d_{it}$. Optimiser sur un horizon de temps plus faible, du point de vue du solveur peut avoir un effet bénéfique sur la qualité des solutions obtenues.

4.2.2 Méthodologie

Les expérimentations ont consisté en l’utilisation du système développé en section 3.2, avec comme modèle d’optimisation le modèle hybride, et comme méthode interactive la méthode présentée en sous-section 3.7.1, sur le carrefour C , pour contrôler le trafic sur une période d’une heure. On parlera, pour simplifier du “modèle hybride” pour désigner le système. Les itérations sont effectuées toutes les $d_{it} = 10$ secondes (voir 4.2.1 pour une description détaillée de ce que représente d_{it}). De plus, un autre système a été mis en œuvre afin de pouvoir effectuer des comparaisons. Il s’agit d’un système semi-adaptatif.

Plusieurs simulations de trafic ont été envisagées, appelées “scénarios”, ces derniers étant définis par les débits d’arrivées véhicules, les longueurs de files d’attentes à $t = 0$, et enfin des arrivées de bus, programmées dans l’heure.

Précisons que les capteurs sont supposés “parfaits”, c’est-à-dire qu’ils sont capables de détecter les files sans limitation de longueur.

4.2.2.1 Politique générale de régulation

La politique de régulation se fait conformément aux règles énoncées en 3.7.1 (le décideur ne possède pas d’expertise particulière), avec prise en compte des remontées de files ou non suivant le scénario. Le point de référence est recalculé à chaque itération, en fonction de la valeur du taux des arrivées et de la prise en compte éventuelle des remontées de file, selon le scénario.

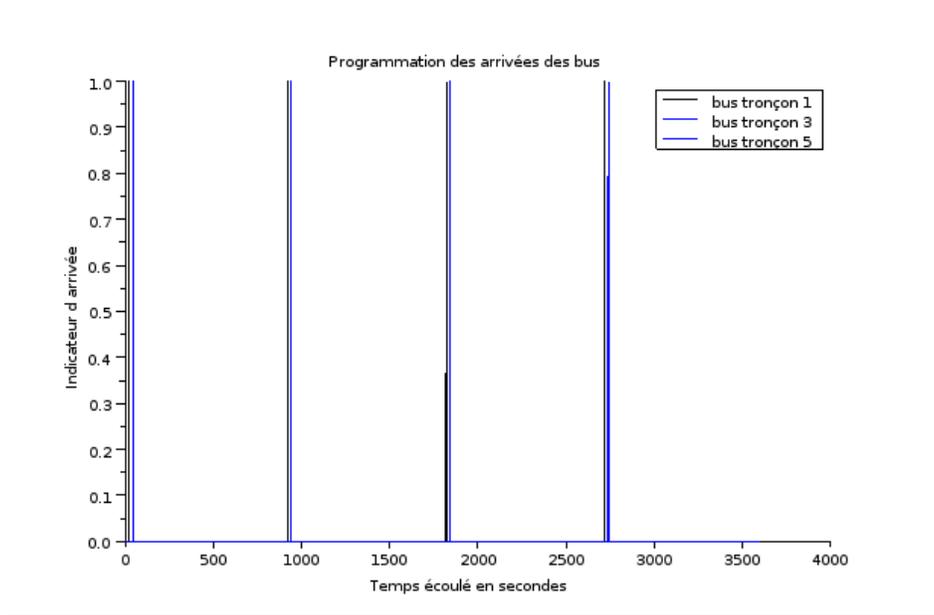


FIGURE 4.3 – Plan d’arrivées des bus utilisé dans les différents scénarios

4.2.2.2 Simulation du trafic général

Pour pouvoir tester le modèle hybride en temps réel il est nécessaire d’être en mesure de simuler un trafic, de façon la plus réaliste possible.

La perception qu’a le système du trafic reposant uniquement sur la connaissance des débits et des files d’attente de chaque tronçon la simulation va consister à fournir ces données en temps réel.

Plusieurs scénarios d’évolution des débits d’arrivée ont été envisagés (voir sous-section 4.2.2.4 pour la description de chaque scénario), correspondant, dans la mesure du possible, à des cas que l’on peut rencontrer dans la réalité. En ce qui concerne la file d’attente $l_f^0(t)$ sur le tronçon f à l’instant t de la simulation, elle est fonction du débit QA_f et de la couleur du feu f de leur tronçon. En théorie, sur un horizon discret, la loi d’évolution de l_f^0 est donnée, conformément à la modélisation de trafic considérée (section 3.1), par :

$$l_f^0(t + \Delta t) = l_f^0(t) + QA_f(t) \times \Delta t$$

si le feu f est au rouge de t à $t + \Delta t$, et

$$l_f^0(t + \Delta t) = \max \{l_f^0(t) + (QA_f(t) - QS) \times \Delta t, 0\}$$

si le feu f est au vert de t à $t + \Delta t$.

Cependant, il s'agit d'une évolution théorique. Pour prendre en compte le fait que lors d'une application réelle l'évolution des files d'attente pourrait être perturbée par des aléas, nous transformons la loi d'évolution en :

$$l_f^0(t + \Delta t) = \xi(t) \times (l_f^0(t) + QA_f(t) \times \Delta t)$$

si le feu f est au rouge de t à $t + \Delta t$, et

$$l_f^0(t + \Delta t) = \xi(t) \times \max \{l_f^0(t) + (QA_f(t) - QS) \times \Delta t, 0\}$$

si le feu f est au vert de t à $t + \Delta t$, où pour chaque $t \in \mathbb{Z}$, $\xi(t)$ est la réalisation d'une variable aléatoire suivant une loi normale d'espérance 1 et d'écart type 0.1. Toutes les variables $\xi(t)$ sont indépendantes deux à deux.

Enfin, dans notre contexte on choisit $\Delta t = u = 1s$.

A un instant t donné de la simulation, les $l_f^0(t)$, $f \in F$ peuvent être passées au système comme longueurs de files initiales (à l'instant 0 de son horizon de temps). C'est-à-dire que l'on effectue, pour $f \in F$:

$$\begin{aligned} le_{0,f} &\leftarrow l_f^0(t) \\ ld_{0,f} &\leftarrow l_f^0(t) \end{aligned}$$

où $le_{0,f}$ et $ld_{0,f}$ sont respectivement les longueurs initiales des files d'attentes relativement aux variables par états et aux variables par dates.

Le modèle hybride n'effectue qu'une optimisation toutes les d_{it} secondes et n'interroge donc les capteurs que toutes les d_{it} secondes. Il ne peut donc pas prendre en compte les aléas sur les arrivées fréquemment, ce qui engendrera une erreur sur l'optimisation, d'autant plus grande que d_{it} le sera. Un système qui interrogerait les capteurs toutes les secondes ou moins serait beaucoup plus protégé de ces phénomènes. C'est pour prendre en compte cet inconvénient (pour le modèle hybride) qu'il nous a semblé nécessaire de modéliser ces perturbations.

4.2.2.3 Simulation des bus

Les bus arrivent suivant le plan donné sur la figure 4.3. Le choix des arrivées a été fait de telle sorte qu'elles mettent en scène des bus, en conflit, arrivant à des moments où la situation de trafic est différente. Les dates d'arrivées ne sont pas perturbées directement. Néanmoins, à partir du moment où un bus pénètre sur un tronçon, l'interaction qu'il subit avec les véhicules joue un rôle perturbateur. Précisons que, dans notre contexte, une perturbation portant sur la date d'arrivée d'un véhicule en général n'a de sens que si celle-ci est inférieure à l'horizon (l'instant présent étant l'instant 0).

4.2.2.4 Scénarios

Scénario 1

Dans le scénario 1, les débits augmentent régulièrement et de la même manière sur chaque tronçon, pendant la période impartie. Une représentation continue est fournie figure 4.4.

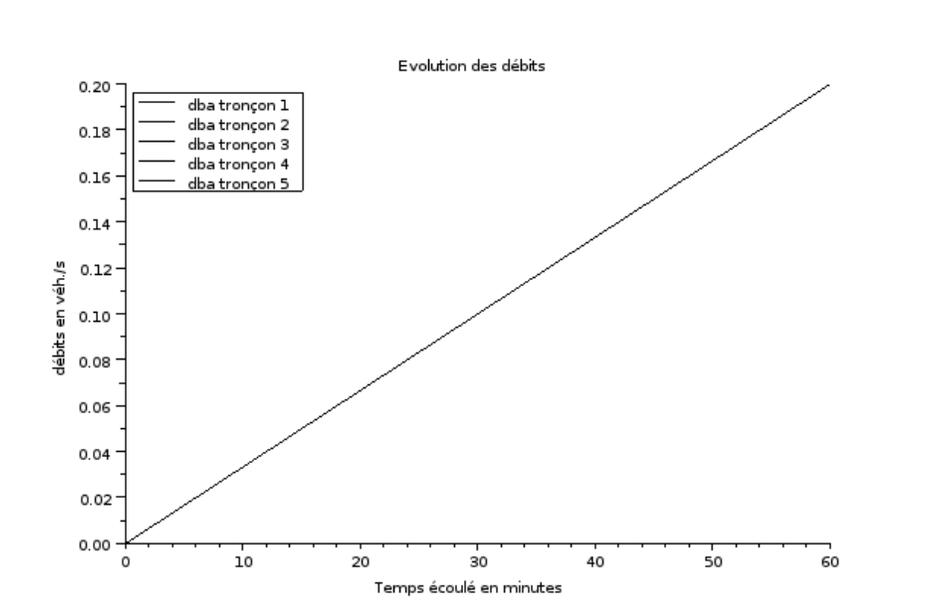


FIGURE 4.4 – Évolution des débits scénario 1

Étant donné la nature discrète des débits (voir l'introduction du chapitre 4), la courbe réelle de leur évolution a un aspect discret, comme le montre la figure 4.5.

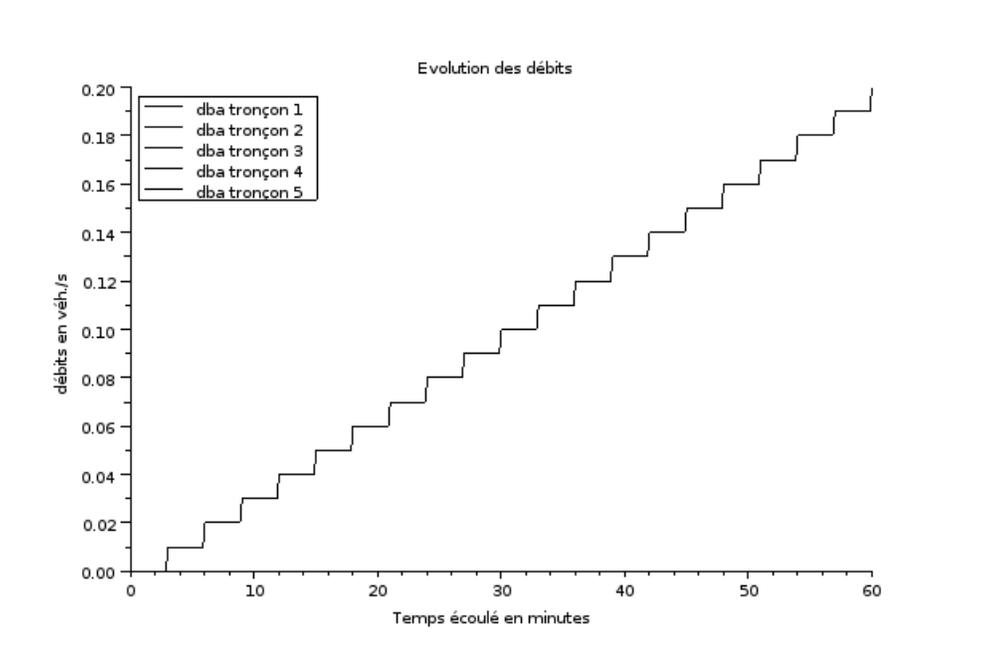


FIGURE 4.5 – Évolution des débits en tenant compte de la nature discrète de ces derniers - scénario 1

Cette forme en escaliers se retrouvera également sur la courbes du taux d’arrivées.

Pour les autres scénarios, nous ne présenterons que des courbes d’évolution de débits continues, afin de mieux appréhender l’évolution de l’ensemble des débits plutôt que leur nature, laquelle ayant moins d’intérêt.

Scénario 2

Le but du scénario 2 est multiple. On essaie, par ce scénario, de représenter une situation “réelle” où l’aspect adaptatif est sollicité, pour chaque système. Pour cela on fait varier significativement les débits d’arrivées sur chaque tronçon, et ce de manière asymétrique. De plus, les débits d’arrivées sont en moyenne relativement élevés à plusieurs moments du scénario, ce qui permet d’inclure, dans le scénario, des périodes où le carrefour est saturé.

Le scénario 2 représente un moment de la journée où :

- Pour une raison quelconque, un événement fait qu’un groupe important d’usagers souhaite se rendre du sud-est du carrefour C à l’ouest, puis effectuer le trajet

- inverse. Dans un premier temps, ils empruntent, pour ce faire, les tronçons 1 et 3 (trajet aller). Puis, dans un second temps, les tronçons 4 et 5 pour le retour.
- Les arrivées véhicules sur les tronçons 1 et 2 sont plus importants à la fin de l'heure qu'en début d'heure, pour des raisons liées, par exemple, à un déplacement domicile-travail des usagers. La somme des débits de ces deux tronçons est en effet croissante durant le premier quart d'heure, puis se stabilise (autour de la valeur 0.3 véh./seconde). La direction empruntée, qui se traduit par le choix du tronçon 1 ou du tronçon 2, varie au cours du temps : les usagers préfèrent progressivement le tronçon 2 au tronçon 1, ceci leur permettant par exemple de contourner l'endroit où a lieu l'événement évoqué au premier point.

L'évolution des débits proposés pour ce scénario est représentée sur la figure 4.6

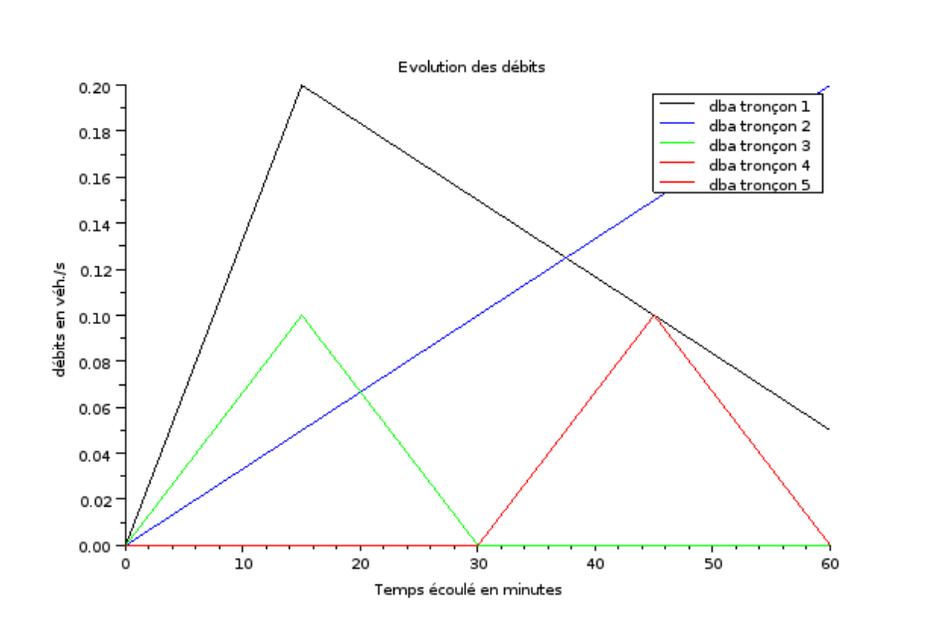


FIGURE 4.6 – Évolution des débits scénario 2

On présentera deux types de résultats liés à ce scénario. L'un sans prise en compte des remontées de files, l'autre avec prise en compte des remontées de files. Les seuils de remontée sont les mêmes pour chaque tronçon. Si la longueur d'une file atteint un seuil, une remontée est déclarée, et la politique de régulation prend en compte celle-ci. La remontée est annulée seulement si la longueur de la file passe en-dessous de 90% du seuil, pour éviter d'éventuelles oscillations brutales de la politique de régulation.

Pour savoir exactement comment sont prises en compte les remontées, le lecteur peut se reporter à la section 3.7.1.

Scénario 3

Le scénario 3 est une variante du scénario 2 dans lequel les “retours”, par les tronçons 4 et 5, se font plus tôt (figure 4.7).

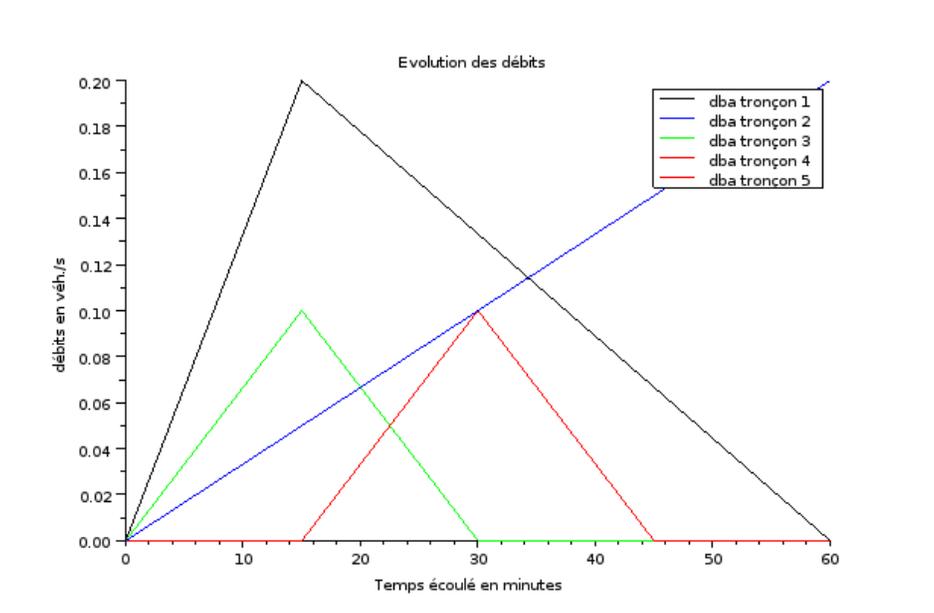


FIGURE 4.7 – Évolution des débits scénario 3

Le but premier est d’observer l’adaptation des deux systèmes à ce scénario différent, par rapport au scénario 2, toujours en comparant leurs performances.

De plus, on présentera deux types de résultats pour ce scénario : l’un où le phasage choisi pour le système S est le meilleur phasage possible pour l’ensemble des instances du scénario (on parlera de *phasage optimal*), et l’autre où le phasage choisi est différent du phasage optimal, afin d’observer l’impact de la rigidité du système S sur un scénario comme le scénario 3.

4.2.2.5 Paramètres

Les paramètres “statiques” sont les suivants :

- Durée minimale de vert : 10 secondes

- Durée maximale de vert : 30 secondes
- Durée minimale rouge : 20 secondes
- Durée maximale de rouge : 50 secondes
- Durée de rouge de dégagement : 5 secondes

Ces paramètres correspondent aux paramètres usuellement imposés par l’exploitant.

4.2.2.6 Comparaison avec un autre système

Pour chaque scénario, on souhaite comparer les résultats obtenus avec le modèle hybride à des résultats qui seraient obtenus avec un système semi-adaptatif. Pour cela nous utilisons un appelé système S , dont le but est de faire de la régulation de trafic en s’inspirant de la méthode employée dans le système SCOOT au niveau local.

Les caractéristiques et la fonctionnement du système S sont les suivants :

- Le système S possède un phasage fixe, qu’il convient de déterminer à l’avance. On doit aussi lui fournir des temps initiaux de durée de phase.
- A partir d’un plan de feu initial, on peut agir en temps réel sur la durée, en secondes, de la phase en cours, avec une limite maximum de +4 secondes et minimum de -4 secondes, tout en respectant les durées minimales et maximales de rouge et de vert, pour chaque feu, imposées par l’exploitant.
- La nouvelle durée de phase est mémorisée pour les prochaines itérations. Ceci permet de pouvoir modifier progressivement sa durée dans toute la plage autorisée par l’exploitant, dans les limites du phasage imposé.

Les paragraphes suivant soulignent les aspects où il existe une différence notable entre le système S et le modèle hybride. Tous les autres aspects importants (modélisation du trafic à la base du modèle, politique de régulation, etc.) sont similaires.

Aspect adaptatif

Les deux systèmes ont pour instance, à chaque itération, les débits courants ainsi que les longueurs de files d’attente. Le système S possède un phasage fixe, il appartient donc à la classe des systèmes semi-adaptatifs. Le modèle hybride, lui, est un système adaptatif.

Cependant, une différence majeure entre les deux systèmes, en faveur du système S , est la fréquence d’optimisation. En effet, le système S optimise chaque seconde,

tandis que le modèle hybride effectue ces opérations une fois toutes les d_{it} secondes. Le système S possède donc, en général, de l'information plus récente pour effectuer son optimisation.

Exactitude de la méthode d'optimisation

Le modèle hybride est une méthode exacte. Le système S , quant à lui, peut être considéré comme une heuristique, car il n'envisage pas toutes les possibilités à chaque itération. En effet, outre le fait qu'il possède un phasage fixe, il ne peut agir que sur la durée d'une seule phase à la fois (la phase courante). De plus, son action sur cette durée, à une itération donnée, est limitée. Il ne peut, en effet, la modifier que d'une amplitude de $2 \times 4 = 8$ secondes. Cependant, comme il effectue une itération par seconde, il peut modifier la durée de la phase courante jusqu'à un extremum seulement en quelques secondes.

Choix du phasage pour le système S

Un plan de feu de type cyclique appliqué sur carrefour possédant au moins trois feux mutuellement antagonistes, comme c'est le cas pour le carrefour C , possède un phasage comprenant au moins trois phases. Lors de la construction de tels phasages, il est naturel (et usuel) de privilégier les phasages comportant le plus petit nombre de phases possibles. Dans notre cas, on a le choix entre plusieurs phasages comportant uniquement trois phases.

Une fois établi l'ordre de passage au vert des trois feux mutuellement antagonistes (les feux 1, 3 et 4), et étant donné le graphe des antagonismes, on a le choix entre mettre le feu 2 ou le feu 5 au vert pendant deux verts successifs parmi les verts des feux $\{1, 3, 4\}$ (figure 4.8).

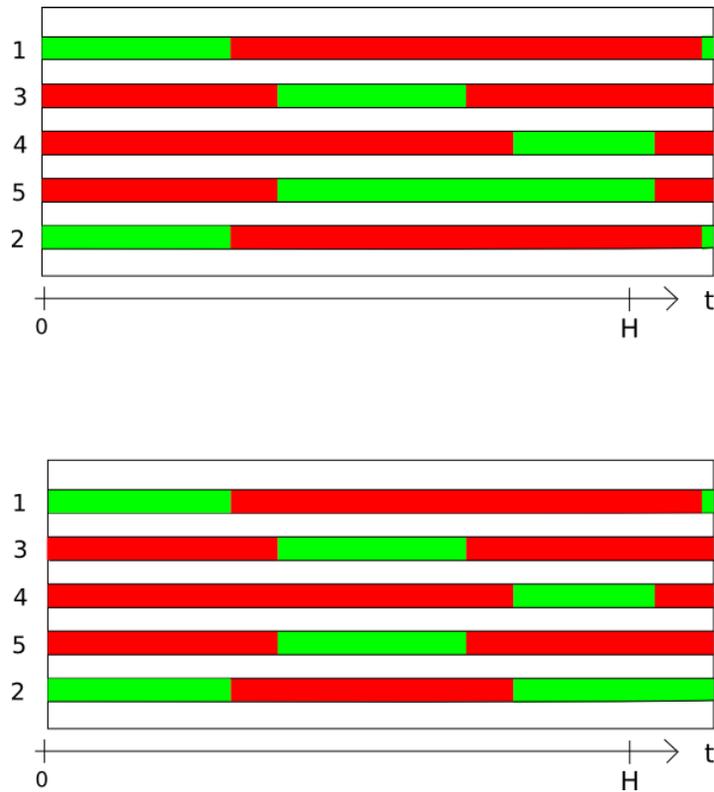


FIGURE 4.8 – Deux types de phasages pour le système S

Pour faire un choix de phasage, on peut réaliser une optimisation off-line en s'appuyant sur les débits moyens estimés sur l'heure pendant laquelle on effectue les optimisations. Dans notre cas, on a retenu le phasage correspondant à la solution (optimale) fournie par le modèle hybride en considérant les débits moyens sur chaque tronçon.

4.2.2.7 Mesures

Modèle hybride

En ce qui concerne le modèle hybride, pour chaque scénario et pour chaque itération on mesure :

- la valeur de chaque critère recalculé sur $d_{it} = 10$ secondes.
- le temps d'optimisation". La limite de temps pour toutes les opérations qu'effectue le programme relatif au modèle hybride est de 10 secondes.

Système S

En ce qui concerne le système S , pour chaque scénario et pour chaque itération (de durée 1 seconde pour le système S), on mesure la valeur de chaque critère recalculé sur 1 secondes, que l'on cumule sur 10 secondes afin de pouvoir comparer les résultats à ceux du modèle hybride.

Mesures et considérations communes

Mesure commune :

On relève le taux des arrivées, ainsi que les remontées de files lorsqu'elles ont lieu, toutes les 10 secondes. On mesure ainsi la *politique* de régulation courante, chaque 10 secondes.

Considérations communes :

Les remarques suivantes s'appliquent aux deux systèmes :

- Les valeurs recalculées le sont à partir des valeurs perturbées des files d'attente (voir sous-section 4.2.2.2).
- Pour le critère NA , dont la variation locale est très grande, les courbes présentées, pour une meilleure lisibilité des résultats, sont des courbes lissées par moyenne mobile (voir la figure 4.32 pour un exemple sans lissage).
- Le nombre de mesure par “type” de mesure (critère, temps d'optimisation, politique) est toujours de 360. En effet, lors de chaque expérimentation, le trafic est contrôlé sur une période d'une heure.

4.2.3 Résultats

4.2.3.1 Scénario 1

On compare ici les résultats obtenus par le modèle hybride et ceux obtenus avec le système S , tous deux confrontés au scénario 1.

Temps d'attente

On constate, sur la figure 4.9 que le temps d'attente total croît durant la période d'expérimentation. De plus, on peut déjà constater, sur ce scénario simple, une certaine

”inertie” dans la variation de ce critère (ce qui sera moins le cas pour les deux autres scénarios). On entend ici par inertie le fait qu’il y ait un certain retard dans la variation des valeurs du critère par rapport à la variation moyenne des débits, en fonction du temps. Dans ce scénario en effet, il faut attendre que les files d’attente se constituent pour observer un retard significatif, ce qui permet d’observer une inertie franche.

On note que le modèle hybride fournit des solutions significativement meilleures sur ce critère.

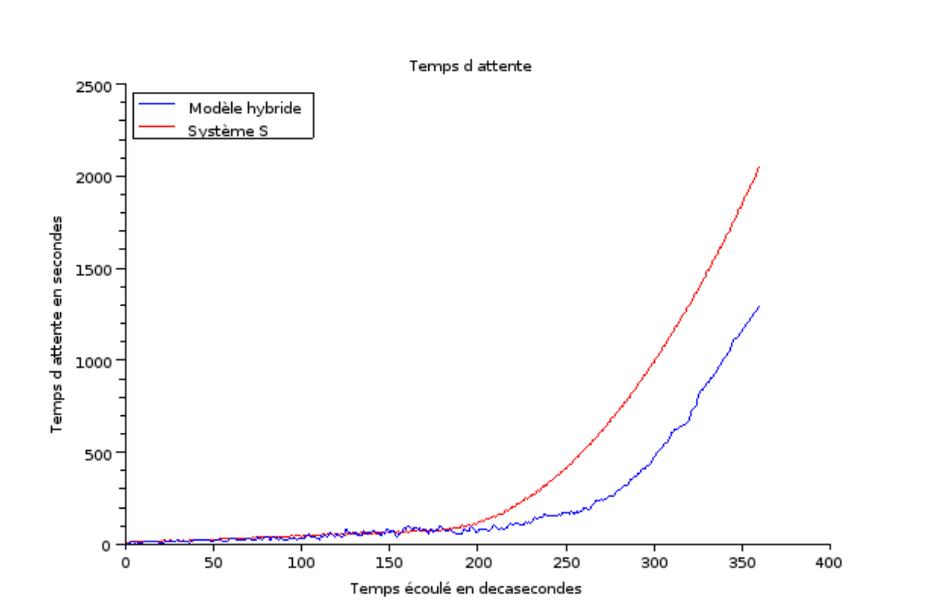


FIGURE 4.9 – Temps d’attente Scénario 1

Nombre d'arrêts

Dans ce scénario, le carrefour est vite saturé. Il est rapidement difficile de pouvoir vider complètement une file sans que le temps d'attente n'en soit fortement impacté. C'est pourquoi, malgré des résultats légèrement meilleurs en moyenne pour le modèle hybride, on a globalement, pour les deux systèmes, une évolution des nombres d'arrêts qui correspond à celle des débits d'arrivées (voir figure 4.10).

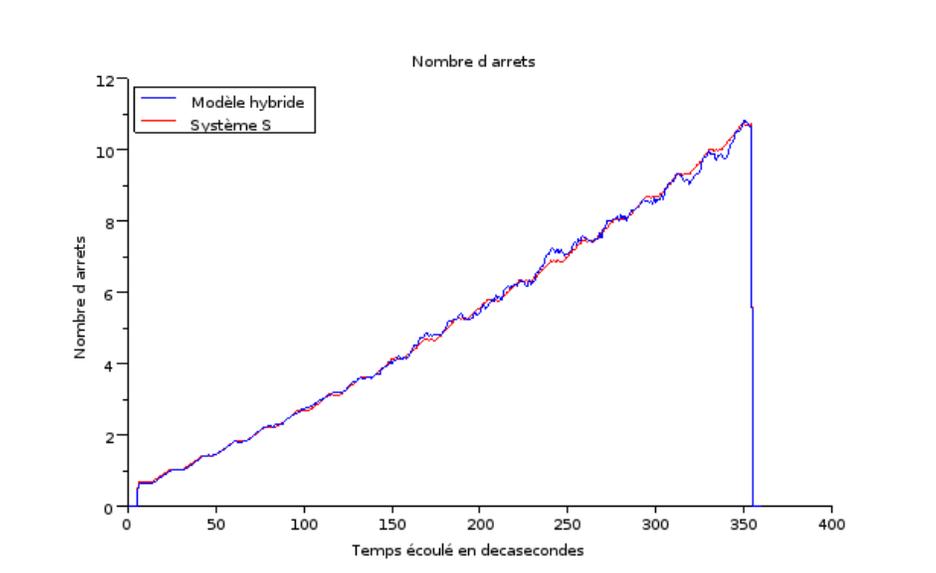


FIGURE 4.10 – Nombre d'arrêts Scénario 1

Erreur de consigne bus

En ce qui concerne le critère relatif aux bus (figure 4.11) on constate que :

- Le modèle hybride fournit des solutions qui sont significativement meilleures, pour ce critère, que celles fournies par le système S
- Dans les deux cas, l'erreur de consigne augmente à peu près conformément à l'augmentation des débits d'arrivée.

Comme le temps d'attente total des bus, qui est égal au critère bus ici, est directement lié à la longueur des files d'attente sur les tronçons concernés, à cause de l'interaction bus-véhicules (voir la modélisation des bus - sous-section 3.1.5), il est normal de retrouver une différence d'autant plus importante, sur ce critère, que le différentiel est grand entre les valeurs de TA fournies par le modèle hybride et le système S .

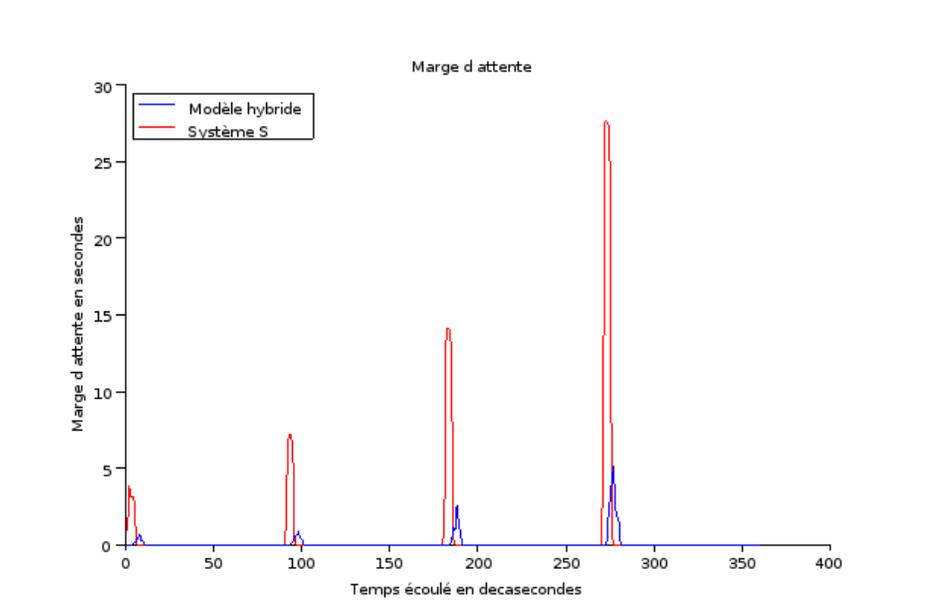


FIGURE 4.11 – Erreur de consigne, Scénario 1

Politique

Pour ce scénario, on n'utilise pas de règle concernant les remontées de files, mais uniquement le taux des arrivées, dont on peut voir l'évolution sur la figure 4.12.

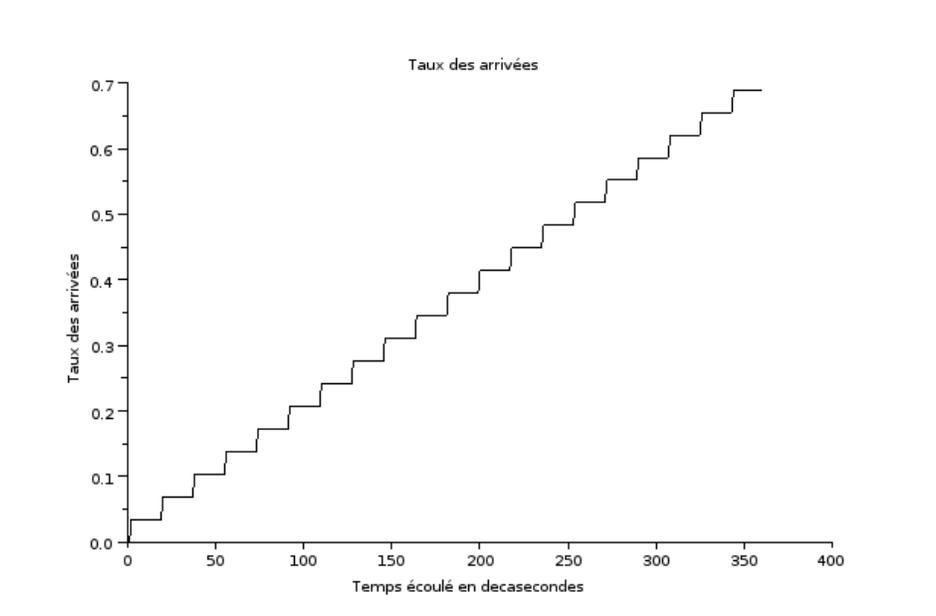


FIGURE 4.12 – Taux d'arrivées Scénario 1

Durée d'optimisation pour le modèle hybride

Sur la figure 4.13, qui représente l'évolution du temps passé par le solveur à essayer de résoudre le problème correspondant à chaque itération, on constate que le modèle hybride trouve régulièrement une solution "optimale", à partir de l'itération 60, en moins de 5 secondes. Les temps d'optimisation sont dans tous les cas très variables. La durée moyenne d'optimisation est de 2.61 secondes. La décroissance en fonction du temps peut être attribuée à l'utilisation de la ré-optimisation au niveau du solveur, qui prend en compte plusieurs itérations précédentes pour trouver une première solution réalisable.

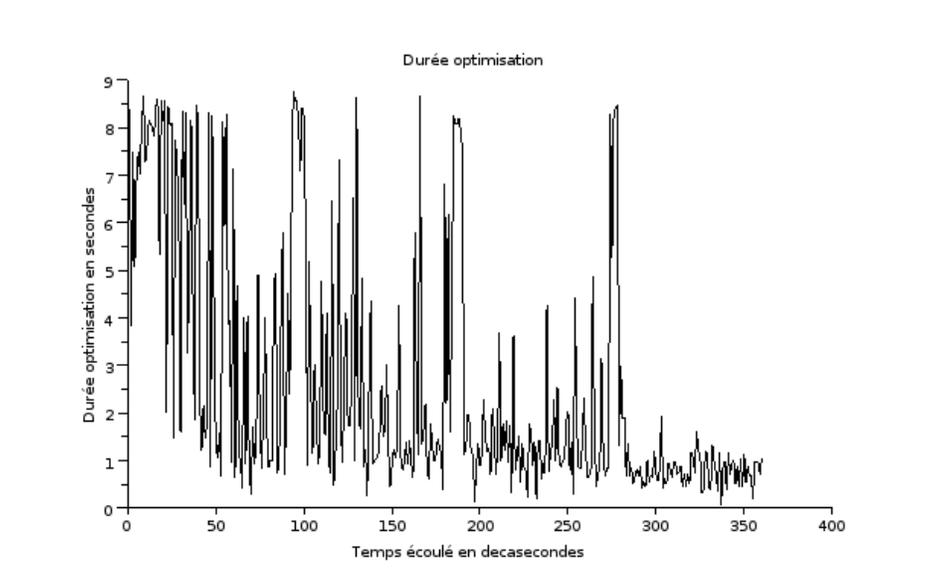


FIGURE 4.13 – Durée d'optimisation Scénario 1

Analyse globale

Pour ce scénario 1, on peut dire que le modèle hybride est aussi bon en moyenne que le modèle S sur le critère NA . Cependant, étant donné le scénario 1, qui est assez particulier, comme tous les véhicules doivent s'arrêter, ce critère n'est pas discriminant.

Le modèle hybride est, en revanche, significativement meilleur sur les deux autres critères, tout au long de la période, avec des valeurs moyennes largement inférieures à celles fournies par le système S .

Il paraît étonnant que dans un scénario aussi simple que le scénario 1, le modèle hybride surclasse aussi distinctement le système S , notamment en ce qui concerne le temps d'attente, alors que la marge de manœuvre semble assez limitée (les débits d'arrivées sur les différents tronçons sont identiques).

On peut observer en particulier que le temps d'attente est nettement meilleur pour le modèle hybride que pour le système S , notamment à partir de l'itération 180 environ (voir figure 4.9).

Observons la vue d'ensemble des solutions fournies par le modèle hybride pendant toute l'expérimentation (figure 4.14).

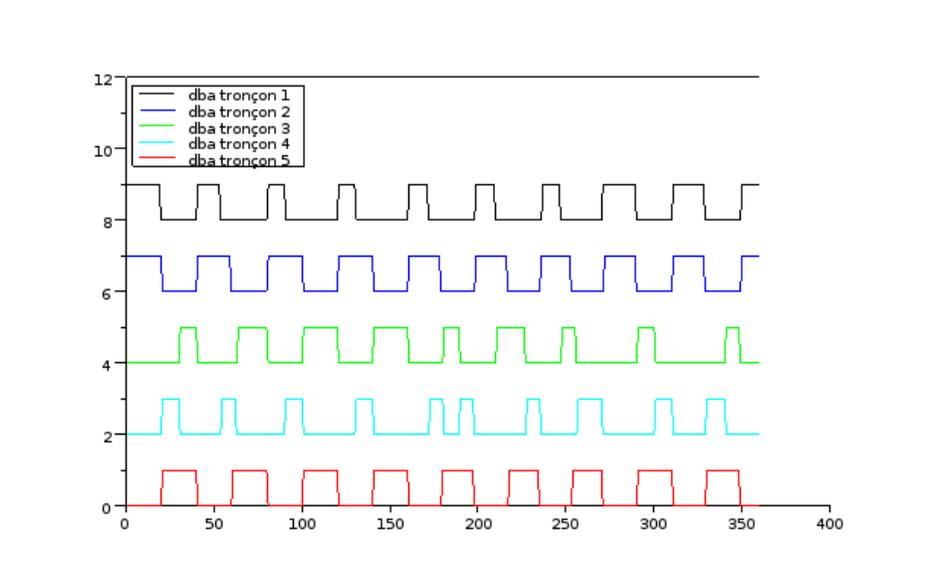


FIGURE 4.14 – Solutions scénario 1

On observe, sur la figure 4.14, plusieurs changements de phasage, dont un particulièrement visible aux alentours de l'itération 180, où l'on passe de l'utilisation de phases

de type $\{2, 4\}$, à celle de phases de type $\{4, 5\}$.

Le système S , quant à lui, ne peut pas effectuer de changement de phasage.

Il semble donc que les bons résultats du modèle hybride sur le temps d'attente, par rapport au système S , obtenus sur un scénario simple comme le scénario 1, soient liés à sa capacité à changer de phasage "lorsque cela est nécessaire".

Un cas théorique simplifié est présenté en annexe C (4.3) pour tenter de montrer que le changement de phasage peut être utile même dans le cas particulier du scénario 1 où les débits d'arrivées sur les différents tronçons sont identiques.

Le critère bus dépendant directement de la longueur des files sur les tronçons concernés, de par les interactions bus-véhicules pris en compte dans la modélisation (voir sous-section 3.1.5), il est normal de retrouver un impact fort sur ce critère également.

Dans $48,61\%$ des cas, le modèle hybride fournit une solution qui *domine* celle fournie par le système S (sur les trois critères). Le tableau 4.10 donne les valeurs moyennes (sur l'heure) de chaque critère recalculé, pour chaque système, et les écarts relatifs correspondants, en pourcentage.

	Modèle hybride	Système S	Écart relatif (%)
TA (secondes)	226,42	406,36	44,28%
NA (nombre)	4,723	4,726	0,63%
ECB (secondes)	0,09	0,59	84,74%

TABLE 4.10 – Comparaison des valeurs moyennes des critères - Scénario 1

4.2.3.2 Scénario 2 sans prise en compte des remontées de file

On compare ici les résultats obtenus par le modèle hybride et ceux obtenus avec le système S, tous deux confrontés au scénario 2, sans prise en compte des remontées de file.

Temps d'attente

Sur la figure 4.15, on peut voir que le modèle hybride est au moins aussi bon, sur ce critère, que le système S à partir de l'itération 80, et significativement meilleur à partir de l'itération 200, sur le critère TA .

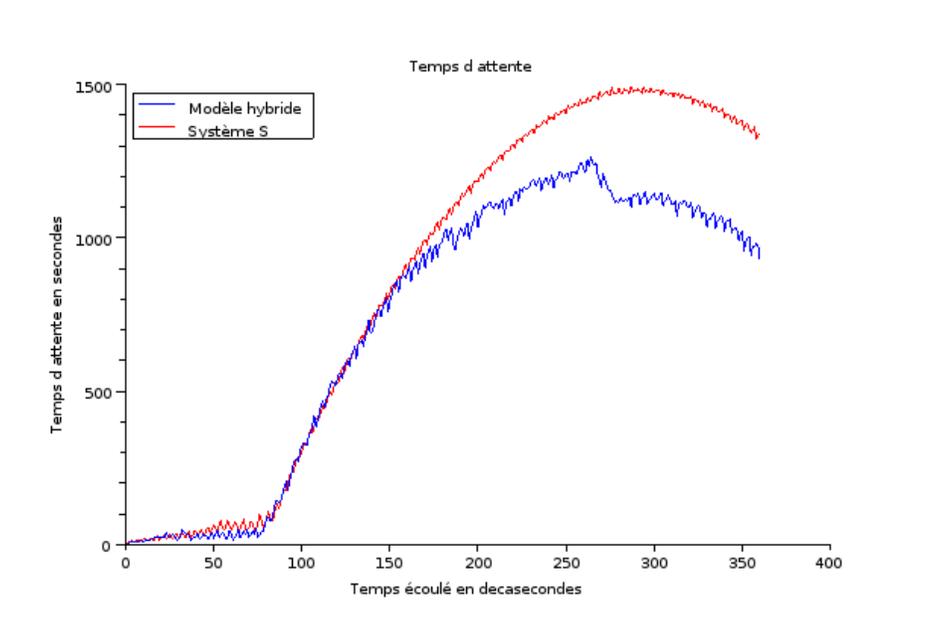


FIGURE 4.15 – Temps d'attente Scénario 2

Nombre d'arrêts

Le modèle hybride est quasiment toujours meilleur que le système S pour le critère NA (figure 4.16). Les grandes lignes des deux courbes ressemblent fortement, ici encore, à celles de la courbe représentant l'évolution du taux d'arrivées (figure 4.18). Ceci est tout à fait normal étant donné la modélisation de trafic choisie et le calcul effectué pour calculer ce critère (voir 3.1). En effet, à chaque seconde et sur chaque tronçon, soit la file est vide et il n'y a pas d'arrêt, soit le nombre d'arrêts est considéré comme étant égal à dba_f , $f \in F$, c'est-à-dire au débit d'arrivée sur le tronçon f .

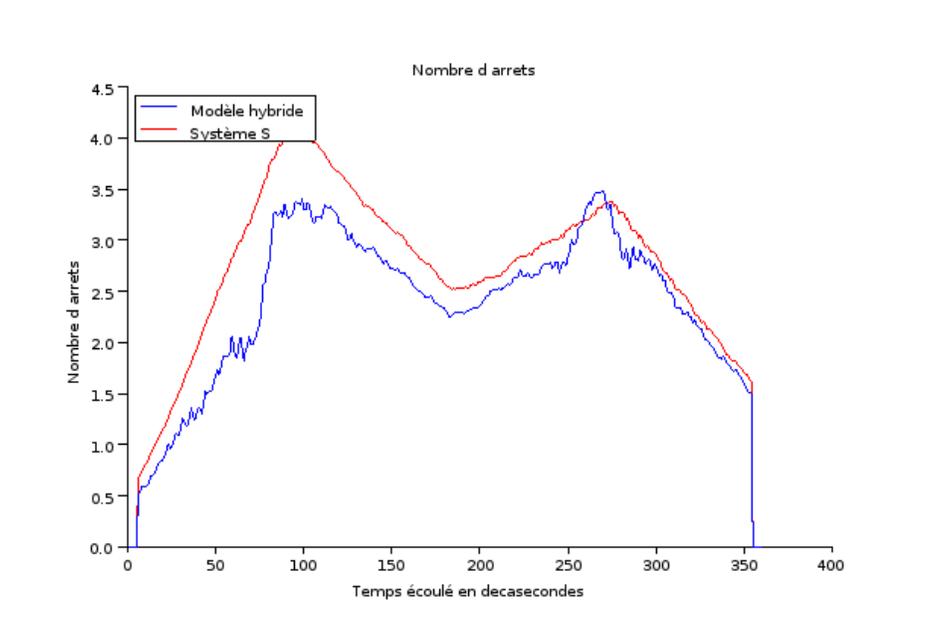


FIGURE 4.16 – Nombre d'arrêts Scénario 2

Erreur de consigne bus

Une fois de plus, le modèle hybride est largement meilleur sur ce critère que le modèle S (figure 4.17).

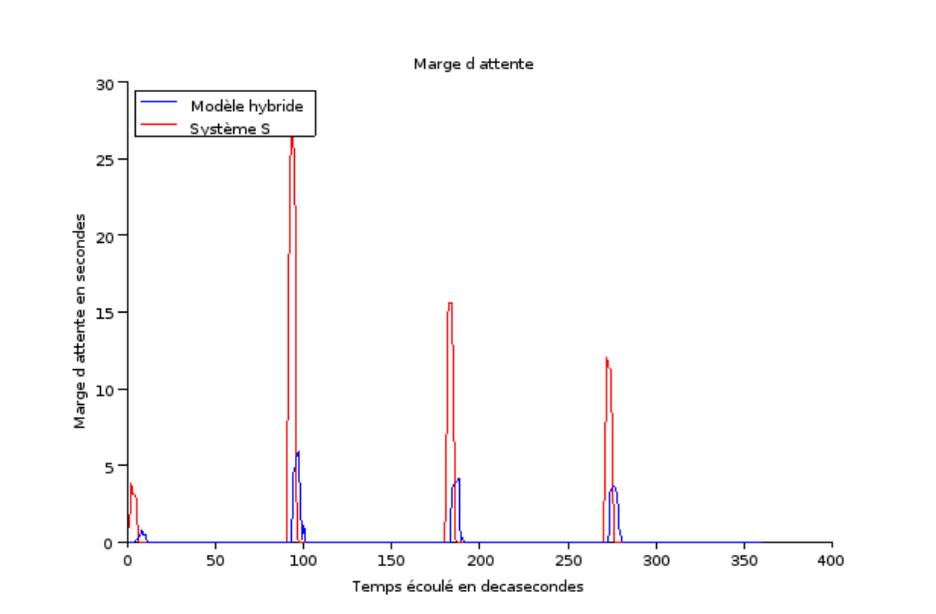


FIGURE 4.17 – Erreur de consigne Scénario 2

Politique

Dans cette version du scénario, on n'utilise pas de règle concernant les remontées de files, mais uniquement le taux des arrivées, dont on peut voir l'évolution sur la figure 4.18.

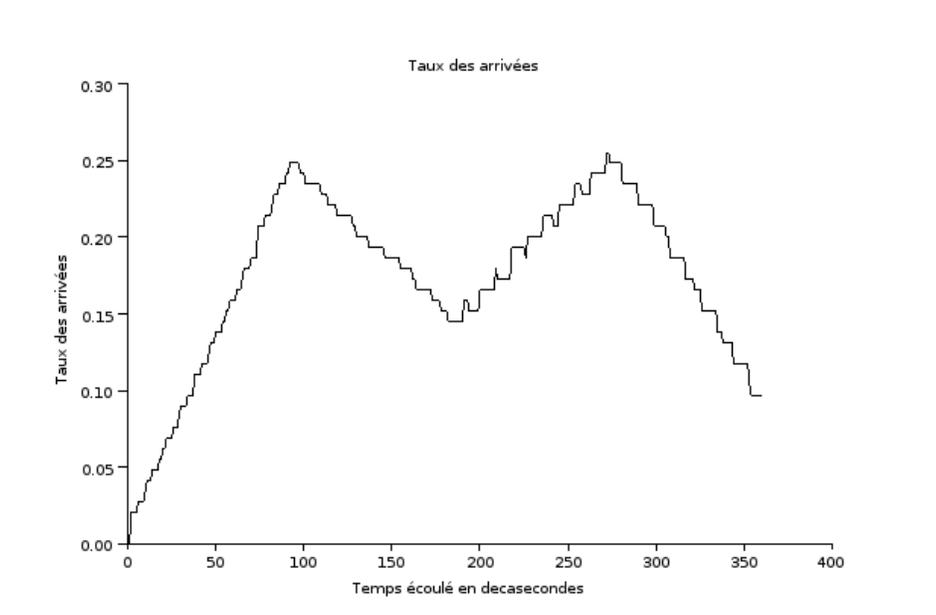


FIGURE 4.18 – Taux d'arrivées Scénario 2

Durée d'optimisation pour le modèle hybride

Les temps d'optimisation observés sur la figure 4.19 sont très variables. Cependant, l'optimum est trouvé un certain nombre de fois en très peu de temps. La durée moyenne d'optimisation est de 2.39 secondes.

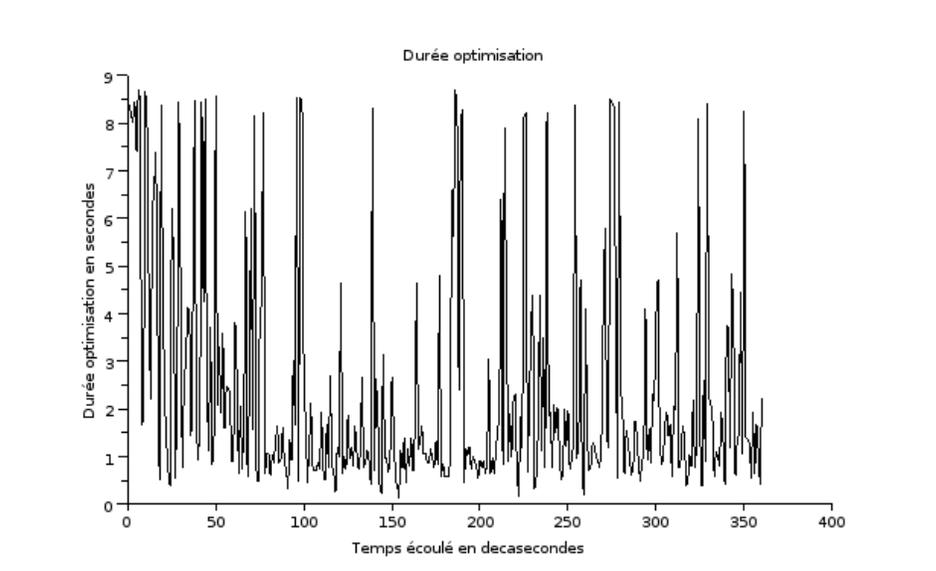


FIGURE 4.19 – Durée d'optimisation Scénario 2

Analyse globale

Dans 58,61% des cas, le modèle hybride fournit une solution qui domine celle fournie par le système S. Le tableau suivant donne les valeurs moyennes sur chaque critère pour chaque système, et les écarts relatifs, en pourcentage.

	Modèle hybride	Système S	Écart relatif (%)
TA (secondes)	728,95	865,71	15,79%
NA (nombre)	2,2237	2,4854	8,11%
ECB (secondes)	0,1813	0,6576	72,42%

TABLE 4.11 – Comparaison des valeurs moyennes des critères - Scénario 2

4.2.3.3 Scénario 2 avec prise en compte des remontées de file

On compare ici les résultats obtenus par le modèle hybride et ceux obtenus avec le système S, tous deux confrontés au scénario 2, avec prise en compte des remontées de file.

Temps d'attente

Comme on peut l'apercevoir sur la figure 4.23, à partir de l'itération 185 pour le modèle hybride et l'itération 175 pour le système S, il existe des remontées de files, qui se traduisent par un changement de la politique en faveur d'une minimisation accrue de TA , ce qui est équivalent à considérer le taux des arrivées τ_A égal à 1.

Sur la figure 4.20 et en comparant à la figure 4.15, on constate l'influence de ce changement de politique sur le critère TA du modèle hybride, dont la courbe est plus "écrasée" à partir de l'itération 185 environ. Pour le système S, la différence est difficilement observable sur la courbe. Il faudra consulter le tableau 4.12 de l'analyse globale pour se rendre compte qu'il y a effectivement une différence.

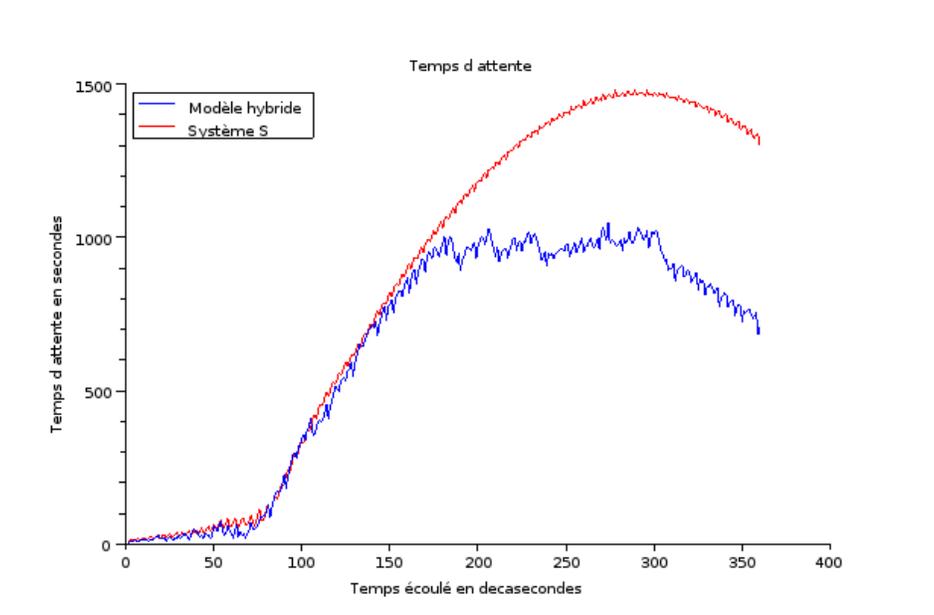


FIGURE 4.20 – Temps d'attente Scénario 2 (avec remontées de files)

Nombre d'arrêts

Le modèle hybride est quasiment toujours meilleur que le système S pour le critère NA (figure 4.21), tout comme pour le scénario 2 sans prise en compte des remontées.

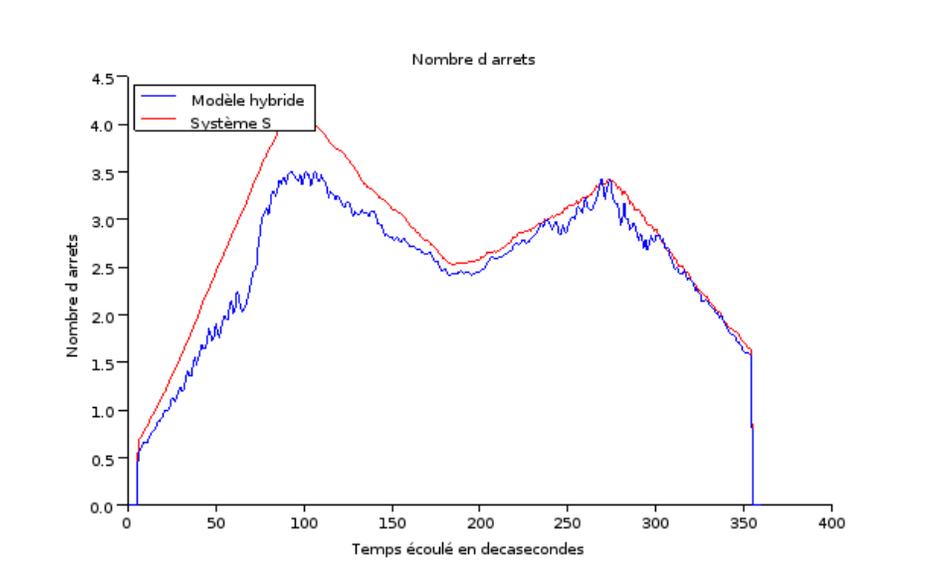


FIGURE 4.21 – Nombre d'arrêts Scénario 2 (avec remontées de files)

Erreur de consigne bus

Le modèle hybride est toujours largement meilleur sur ce critère que le modèle S (figure 4.22).

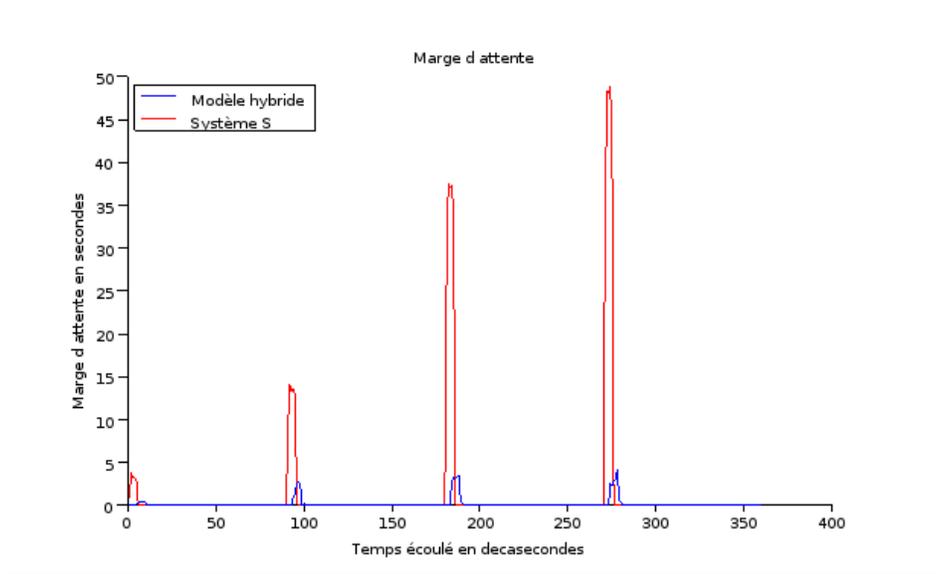


FIGURE 4.22 – Erreur de consigne Scénario 2 (avec remontées de files)

Politique

La figure 4.23 représente l'évolution du coefficient qui fixe la politique de régulation. Ce coefficient vaut τ_A s'il n'y a pas de remontée de file, et vaut 1 s'il y a une remontée déclarée. En ce qui concerne le système S , une remontée de file au moins est déclarée à partir de l'itération 175. Comme le système n'arrive pas à faire baisser suffisamment les longueurs de files, même avec une politique tournée vers la minimisation de TA , la politique reste bloquée dans cet état. En revanche, le modèle hybride, qui déclare une remontée à l'itération 185 par exemple, est capable de faire baisser considérablement TA par cette politique, et dans le même temps d'annuler la remontée de file, ce en seulement 5 itérations.

Les longues files sont très liées à des temps d'attente élevés. Il suffit de voir l'écriture de TA pour s'en convaincre (voir sous-section 3.4.5). On constate ici que, au moins pour le modèle hybride, la règle choisie pour la prise en compte des remontées de files permet de limiter ces dernières.

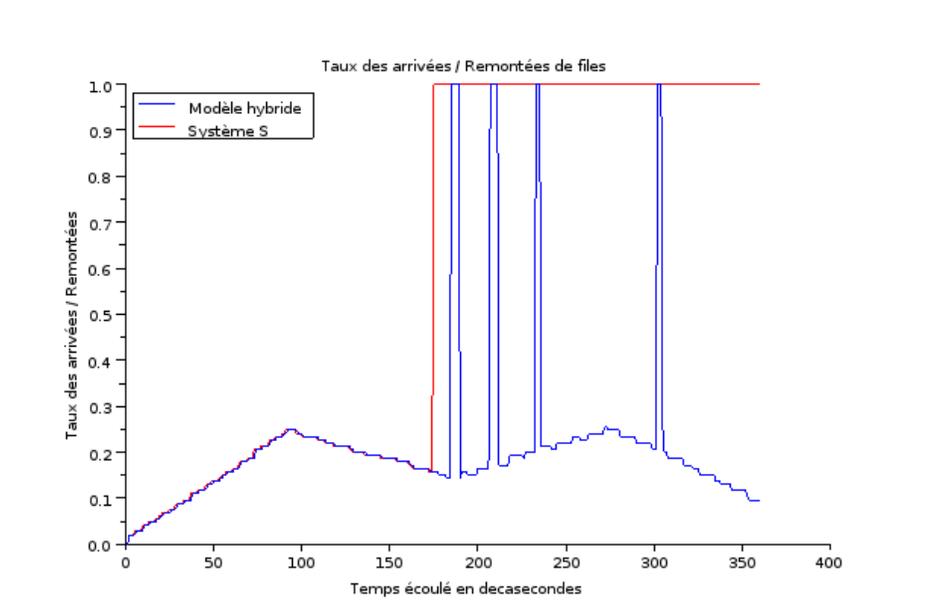


FIGURE 4.23 – Taux d'arrivées Scénario 2 (avec remontées de files)

Durée d'optimisation pour le modèle hybride

La prise en compte des remontées ne change pas significativement les temps de calcul par rapport au scénario 2 joué sans remontée.

Analyse globale

Dans 68,33% dans cas, le modèle hybride fournit une solution qui *domine* celle fournie par le système S. Le tableau 4.12 suivant donne les valeurs moyennes sur chaque critère pour chaque système, et les écarts relatifs, en pourcentage. On constate que les valeurs du critère *TA* sont plus faibles, pour chacun des modèles, que celles du tableau correspondant à la version sans prise en compte des remontée (tableau 4.11), de par la politique de régulation qui inclut cette fois-ci les remontées de file. En revanche, le critère *NA* est plus élevé, ce qui est une contre partie de la politique choisie. Quant au critère bus, qui, nous l'avons vu dans la première partie de cette sous-section 4.2.3, peut être partiellement corrélé au critère *TA*, il est également plus faible.

	Modèle hybride	Système S	Écart relatif (%)
TA (secondes)	635,33	862,20	26,31%
NA (nombre)	2,2670	2,5116	9,7%
ECB (secondes)	0,1280	1,2985	90,13%

TABLE 4.12 – Comparaison des valeurs moyennes des critères - Scénario 2 (avec remontées de files)

4.2.3.4 Scénario 3 avec phasage optimal

On compare ici les résultats obtenus par le modèle hybride et ceux obtenus avec le système S muni d'un phasage "optimal" pour répondre à l'évolution globale du scénario.

Comme le scénario prévoit que les tronçons 1 et 2 supporteront un débit élevé, en fin de scénario, par rapport aux autres, et dans la mesure où l'on a cette information, il serait logique de choisir un phasage pour lequel les files d'attentes sur les tronçons 1 et 2 peuvent atteindre leur durée maximum de vert simultanément.

Le phasage retourné par le solveur lorsqu'on pose le problème avec les débits moyens sur une itération est justement :

$$\{1, 2\}, \{3, 5\}, \{4, 5\}$$

Ce phasage est donc cohérent. Un exemple de plan de feux utilisant ce phasage peut être observé sur la partie basse de la figure 4.8 (sous-section 4.2.2.6).

Temps d'attente

Sur la figure 4.24, on peut voir que le modèle hybride est quasiment meilleur sur toute la période, avec un différentiel assez élevé plus de la moitié du temps.

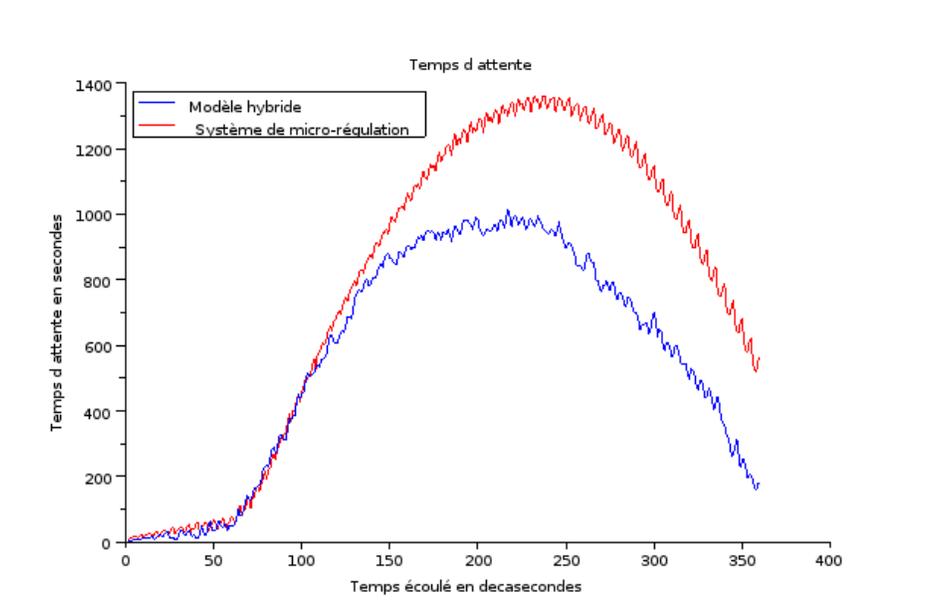


FIGURE 4.24 – Temps d'attente Scénario 3 - Phasage optimal

Nombre d'arrêts

Le modèle hybride est au moins aussi bon que le système S sur presque toute la période, comme le montre la figure 4.25.

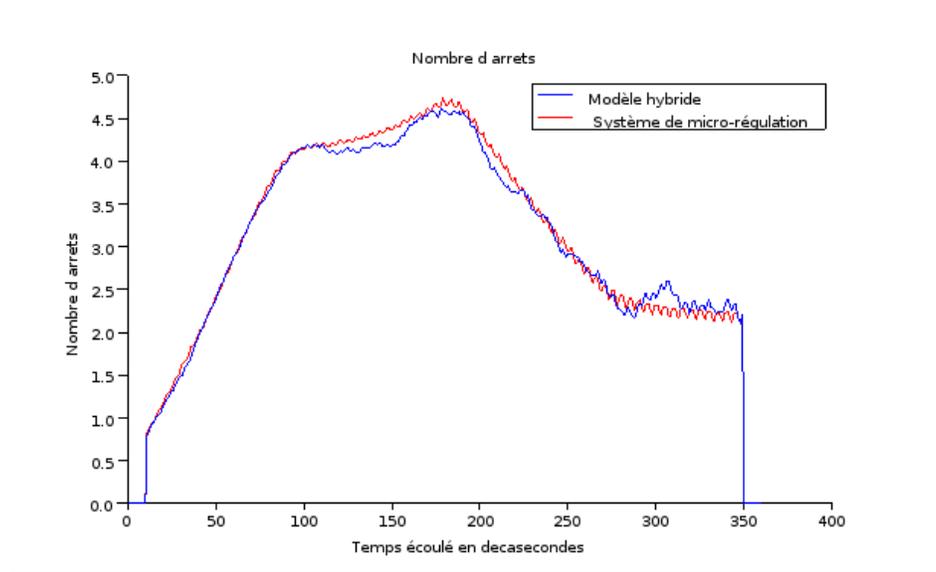


FIGURE 4.25 – Nombre d'arrêts Scénario 3 - Phasage optimal

Erreur de consigne bus

Les erreurs de consigne sont significativement meilleures pour le modèle hybride (figure 4.26).

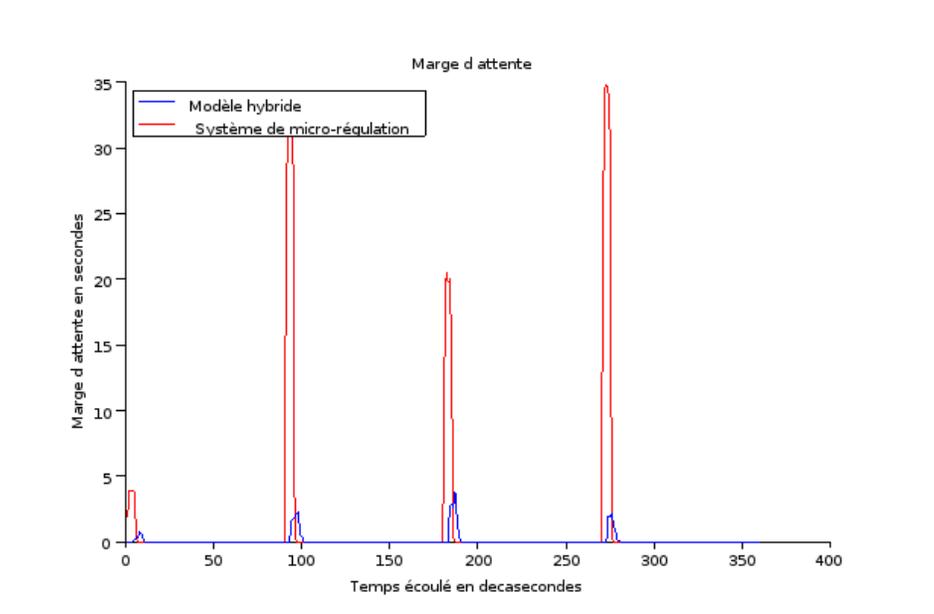


FIGURE 4.26 – Erreur de consigne Scénario 3 - Phasage optimal

Politique

Pour ce scénario, on n'utilise pas de règle concernant les remontées de files, mais uniquement le taux des arrivées, dont on peut voir l'évolution sur la figure 4.27.

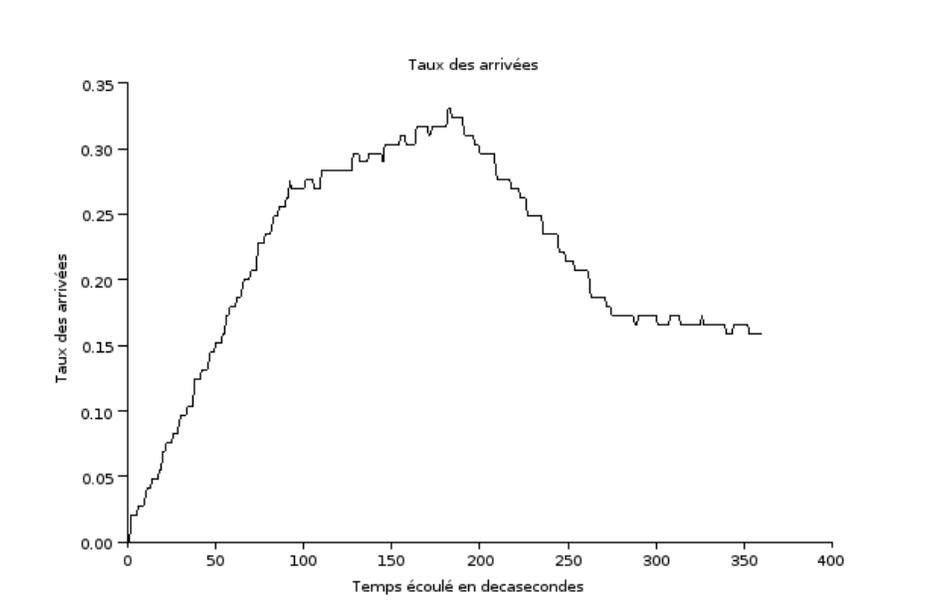


FIGURE 4.27 – Taux d'arrivées Scénario 3

Durée d'optimisation pour le modèle hybride

La durée moyenne d'optimisation est de 2.23 secondes. Le détail est disponible sur la figure 4.28.

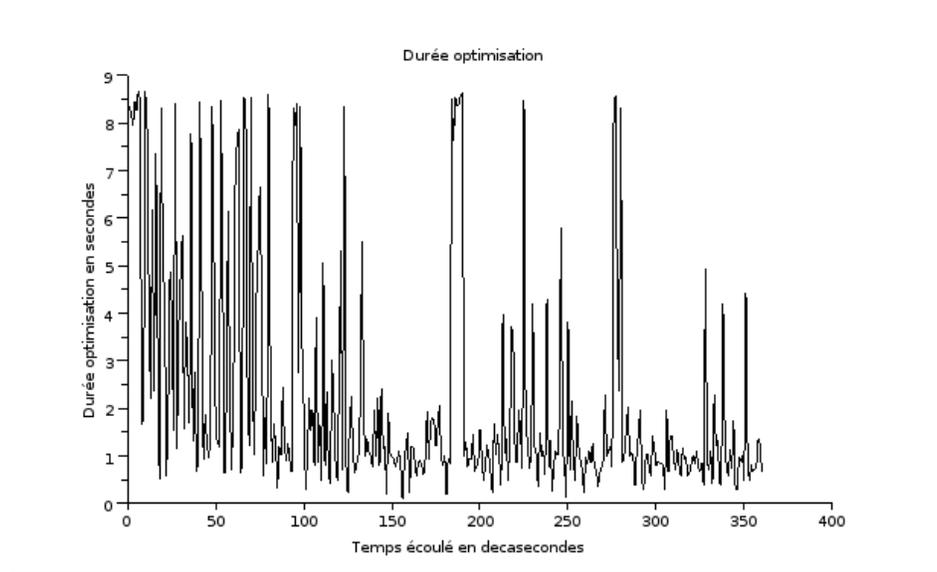


FIGURE 4.28 – Durée d'optimisation Scénario 3

Analyse globale

Dans 66,66% dans cas, le modèle hybride fournit une solution qui *domine* celle fournie par le système S. Le tableau suivant donne les valeurs moyennes sur chaque critère pour chaque système, et les écarts relatifs, en pourcentage.

	Modèle hybride	Système S	Écart relatif (%)
TA (secondes)	397,87	777,85	48,84%
NA (nombre)	2,6032	2,9076	10,46%
ECB (secondes)	0,096	0,5667	83,05%

TABLE 4.13 – Comparaison des valeurs moyennes des critères - Scénario 3

On observe une augmentation plus prématurée du temps d'attente et des nombres d'arrêts que lors du scénario 2. Par rapport au scénario 2, les valeurs moyennes de TA et ECB sont plus faibles, tandis que les valeurs moyennes de NA sont plus élevées. En regardant plus précisément le critère TA , il semble que le modèle hybride ait une meilleure capacité d'adaptation, de ce point de vue, à ce genre d'évolution du trafic (scénario 3), que le système S . On observe, en effet, une plus grande inertie, dans l'évolution des valeurs de critère, pour le système S , ce qui permet d'expliquer une partie de ces résultats.

4.2.3.5 Scénario 3 sans phasage optimal

Si on ne possède pas les informations clés sur les arrivées futures, il se peut qu'on applique un plan de feu, pour le système S , avec un phasage mal adapté. Nous avons pris pour exemple le phasage $\{1, 2\}, \{3, 5\}, \{4, 5\}$, auquel correspond l'exemple de plan de feux présentée en partie haute de la figure 4.8 (sous-section 4.2.2.6). On obtient alors les résultats décrits dans cette sous-section.

Temps d'attente

Sur la figure 4.29, on voit que le système S a beaucoup plus de difficultés à baisser le temps d'attente, en fin de période, que le modèle hybride. Le phasage force le système à ne considérer que des solutions avec des grandes périodes de rouge sur le tronçon 2, ce dernier subissant de fortes arrivées par rapport aux autres en fin de période.

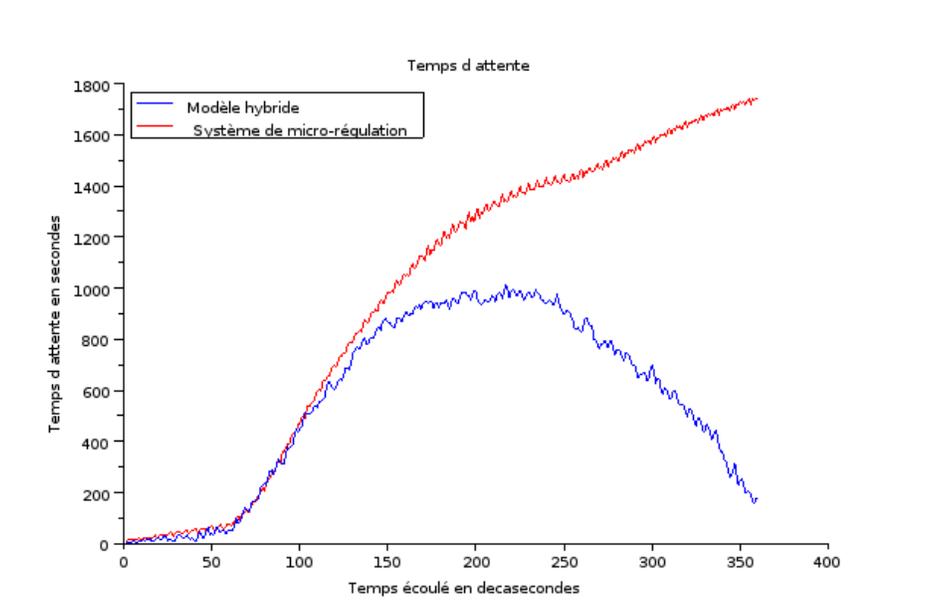


FIGURE 4.29 – Temps d'attente Scénario 3 - Phasage mal adapté

Nombre d'arrêts

Le “mauvais” phasage initial a aussi des répercussions négatives sur le nombre d'arrêts pour le système S (figure 4.30).

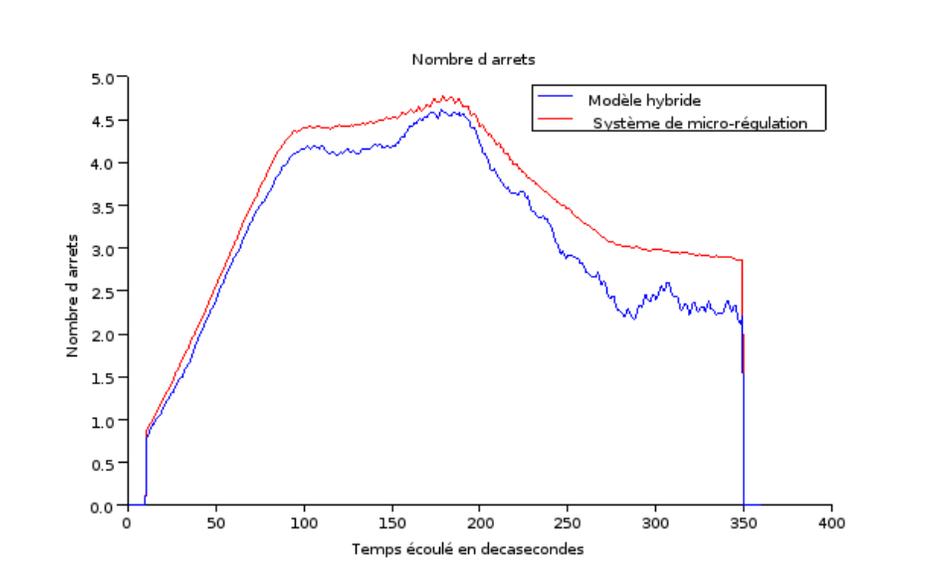


FIGURE 4.30 – Nombre d'arrêts Scénario 3 - Phasage mal adapté

Erreur de consigne bus

Dans ce scénario, les erreurs de consigne atteignent des valeurs très élevées, pour le système S (figure 4.31). Cela est sans doute partiellement dû à la mauvaise gestion du temps d'attente, qui est relié à la longueur des files d'attentes.

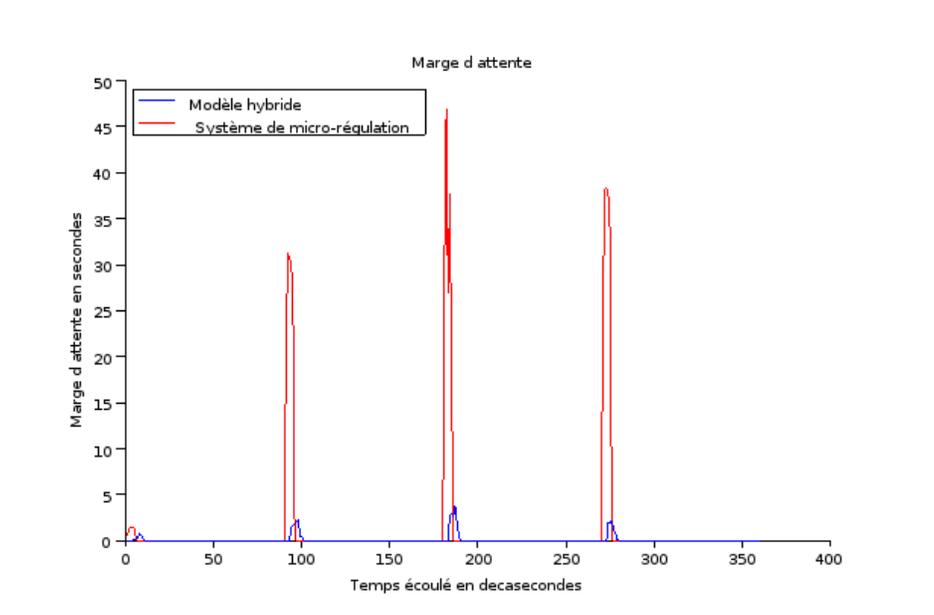


FIGURE 4.31 – Erreur de consigne Scénario 3 - Phasage mal adapté

Évolution du taux des arrivées et durée d'optimisation pour le modèle hybride

L'évolution du taux des arrivées et de la durée d'optimisation sont, bien sûr, les mêmes que pour le scénario 3 avec phasage optimal pour le système S .

4.2.4 Conclusions et remarques sur les résultats obtenus

Conclusion

La conclusion importante, au regard des résultats obtenus pour l'utilisation sur plusieurs itérations, est que les solutions fournies par le *système de S* sont, la majeure partie du temps, dominées sur tous les critères par celles fournies par le *modèle hybride*.

Ces résultats permettent donc de montrer l'intérêt de considérer, dans notre contexte, un système qui possède une grande capacité d'adaptation mais dont la fréquence d'optimisation est moindre, plutôt qu'un système possédant les caractéristiques inverses.

Quelques remarques sur le comportement global des critères :

Comme nous l'avons mentionné en sous-section 4.2.3.1, on peut observer que le critère *TA* possède une certaine inertie dans son évolution, liée à l'évolution plus ou moins lente de la longueur des files. Le critère *NA* est, en revanche, un critère beaucoup plus "volatile", dans le sens où l'on observe de très grandes variations en très peu de temps. La figure 4.32 représente l'évolution de *NA* sans lissage.

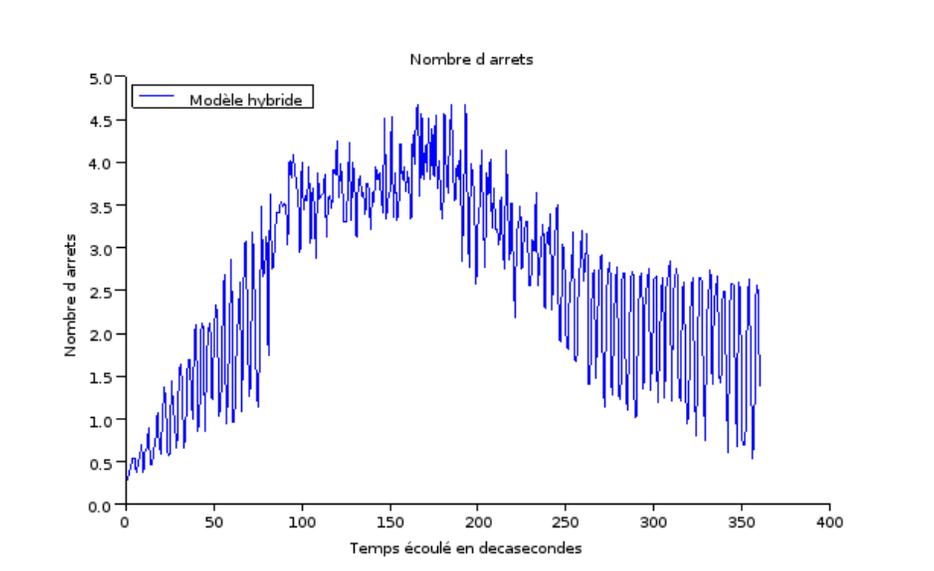


FIGURE 4.32 – Exemple d'évolution du critère *NA* sans lissage - Scénario 3

Ceci s'explique par la façon dont les systèmes font baisser le nombre d'arrêts, au regard de la modélisation du trafic : il suffit qu'il existe une file suffisamment courte pour que le système puisse l'annuler sur un laps de temps suffisant pour éviter un certain

nombre d'arrêts. Ainsi, les courbes présentées relatives à ce critère ont été lissées, afin de mieux dégager l'évolution globale des valeurs.

Temps d'optimisation

Au niveau du temps d'optimisation pour le modèle hybride, on constate que les plus longues durées en moyenne correspondent aux premières itérations, pour des raisons liées à la ré-optimisation. Cependant, ces durées peuvent atteindre leur limite maximale imposée à peu près à n'importe quel moment sur la période, avec en particulier des pointes au moment de l'arrivée des bus.

4.3 Conclusion du chapitre

Nous avons donc observé, dans ce chapitre, deux points importants. Le premier point concerne la comparaison des modèles de la thèse. Le second point concerne l'évaluation du modèle hybride par rapport à un système "temps-réel" au sens de la section 1.1.4.

En ce qui concerne le premier point, les résultats permettent de montrer que la construction d'un modèle hybridé entre le modèle par états et le modèle par dates est pertinent.

En ce qui concerne le second point, les résultats permettent en particulier de montrer l'intérêt de considérer, dans notre contexte, un système qui possède une grande capacité d'adaptation mais dont la fréquence d'optimisation est moindre, plutôt qu'un système possédant les caractéristiques inverses.

Ces résultats permettent également de montrer une application d'un exemple de politique générale de régulation, portant sur trois critères considérés dans la thèse et dont l'évolution des valeurs va effectivement dépendre.

Conclusion générale

Synthèse

L'objectif de cette thèse était le développement d'une méthode d'optimisation pour la régulation du trafic, adaptative, multi-objectif et multi-mode, en milieu urbain.

On trouve, dans la littérature, plusieurs travaux menés dans le but de réguler le trafic sur un carrefour, un groupe de carrefours ou un réseau, avec un ou plusieurs critères et des façons différentes de les prendre en compte si plusieurs critères sont considérés. On trouve des méthodes très diverses, avec des flexibilités plus ou moins grandes, des rapidités d'exécution et des exigences de "qualité de solution" plus ou moins importantes. Mais on en trouve très peu, pour ne pas dire aucune, qui sont adaptées aux carrefours complexes avec une réelle problématique multi-objectif. De plus, les systèmes existants ne proposent que très rarement de consacrer un critère aux transports en commun.

L'approche proposée dans cette thèse essaie de combler cette lacune en utilisant une approche multi-objectif interactive, adaptée pour fonctionner en temps réel, avec les critères : TA (temps total d'attente des véhicules particuliers), NA (nombre d'arrêts des véhicules particuliers), et ECB (erreur de consigne pour les bus, soit l'écart entre temps d'attente réel et souhaité par la compagnie, des bus traversant le carrefour). Pour ce faire, trois modèles de programmation linéaire mixte en nombre entiers ont été élaborés. L'un d'entre eux, le modèle hybride, a été conçu pour tirer parti le plus possible des deux autres afin d'améliorer les performances. Les trois modèles s'intègrent dans un système capable d'utiliser un processus d'optimisation multi-objectif interactif permettant d'adapter une politique de régulation en fonction de la situation du trafic. Deux types de méthodes interactives ont été proposées dans la thèse, dont le but est de faire évoluer une politique de régulation du trafic en fonction de la situation du trafic.

Après avoir comparé expérimentalement les trois modèles entre eux, il a été confirmé que le modèle hybride est le plus performant, par rapport à des critères précis, explicités dans la thèse. Le système développé dans la thèse muni du modèle hybride et d'une des deux méthodes interactives a été comparé avec un système semi-adaptatif, et ce sur plusieurs scénarios types, offrant les meilleurs résultats, avec des solutions qui dominent les solutions fournies par le système semi-adaptatif dans la majorité des cas.

La plupart des systèmes de régulation sont des systèmes semi-adaptatifs, c'est-à-dire des systèmes "très rapides car peu exacts". Nous avons montré que, bien utilisée, la méthode exacte qu'est la méthode Branch&Bound en programmation linéaire mixte permet de faire aussi bien, voire mieux que de tels systèmes, même dans un contexte où la contrainte de temps est primordiale (on cherche une solution le plus rapidement possible).

La méthode est certes lente à l'exécution si l'on souhaite obtenir une solution optimale, mais si l'on s'autorise à stopper le processus d'optimisation en cours et à ne récupérer que la solution réalisable courante, on s'aperçoit que l'utilisation d'une telle méthode est pertinente.

On peut résumer cela ainsi : pour certains problèmes d'optimisation devant être exécutés dans un contexte temps réel, il peut être préférable de ne pas s'appuyer sur l'intuition qu'on a du problème pour en faire une simplification réductrice et dommageable, mais d'améliorer le plus possible la modélisation du problème afin de permettre à l'algorithme utilisé de résoudre au moins partiellement le problème en temps limité.

Perspectives

Vision zonale

Une utilisation du système à l'échelle d'un groupe de carrefours (vision zonale) peut être envisagée, éventuellement après quelques modifications. Le modèle hybride étant vraisemblablement le modèle le plus performant à l'échelle d'un seul carrefour, il est naturel d'essayer de l'adapter à une problématique zonale. De nouvelles difficultés se dessinent alors, puisque l'hypothèse d'arrivées régulières des véhicules particuliers sur les différents tronçons impliqués ne tient plus, ce qui impacte directement le modèle.

Prise en compte des piétons

La prise en compte des piétons est également une perspective possible. On pourrait par exemple vouloir minimiser un temps d'attente piétons. En supposant que l'on dispose de capteurs permettant de détecter le nombre de piétons à un endroit donné, de nouvelles problématiques apparaissent naturellement, ne serait-ce qu'au niveau de la modélisation, par rapport à ce qui a déjà été considéré dans la thèse. En particulier, il conviendrait d'étudier :

- La modélisation du comportement des piétons lors de l'attente à un feu. Contrairement aux véhicules particuliers, les piétons peuvent décider de quitter l'endroit où ils attendaient initialement.
- Les interactions avec les véhicules. Les piétons sont en effet autorisés, dans certains cas, à partager un même espace avec les véhicules.
- L'écoulement des piétons sur le passage piéton. Cet écoulement est sensiblement différent de celui des véhicules particuliers. Les piétons peuvent en effet traverser par "paquets".

Amélioration des méthodes interactives d'évolution de la politique de régulation du trafic

Nous avons montré que même si l'ensemble des critères ne représente pas directement la situation de trafic, situation qui dépend alors d'indicateurs externes au modèle, il est possible de trouver une méthode qui permet d'orienter la recherche de solutions en prenant en compte ces indicateurs. En particulier, nous utilisons un "profil", dans l'espace des critères, qui traduit la politique à mener à partir de ces indicateurs, en se basant sur la définition de profils extrêmes, correspondant à des situations de trafic extrêmes. Dans le cas où un décideur aurait à disposition un jeu d'indicateurs importants couplés à un ensemble de nombreuses situations extrêmes de trafic, de nouvelles questions apparaîtraient naturellement, notamment en ce qui concerne la façon de faire évoluer le profil représentant la politique de régulation courante, qui devra être un compromis entre les différents profils extrêmes. Cet axe de recherche dépasse d'ailleurs le cadre de la régulation de trafic : le problème se pose à partir du moment où l'on doit, dans un contexte d'optimisation multi-objectif, déduire des paramètres préférentiels à partir d'indicateurs autres que les critères eux-mêmes.

Bibliographie

- AFNOR (1991). Normalisation française, régulation du trafic routier. Rapport technique P99-000, Afnor.
- AHUJA, R. et ORLIN, J. (1992). The scaling network simplex algorithm. *Operations Research*, 40:5–13.
- ALOULOU, M. A. et ARTIGUES, C. (2010). Flexible solutions in disjunctive scheduling : general formulation and study of the flow-shop case. *Computers and Operations Research*, 37(5):890–898.
- BARCELO, J., GRAU, R., EGEA, P. et BENEDITO, S. (1991). CARS : A demand-responsive traffic control system. *In Applications of Advanced Technologies in Transportation Engineering*.
- BARTO, A., SUTTON, R. et ANDERSON, W. (1983). Neuronlike adaptive elements that can solve difficult learning control problem. *IEEE Transactions on Systems, Man and Cybernetics*, 13:834–846.
- BARTO, G. et SUTTON, R. (1998). *Reinforcement learning : an introduction*. Adaptive Computation and Machine Learning. MIT Press.
- BELLMAN, R. (1957). *Dynamic Programming*. Princeton University Press.
- BELLMAN, R. et DREYFUS, S. (1959). Functional approximation and dynamic programming. *Math. Tables and other Aids to Computation*, 13:247–251.
- BERTSIMAS, D. et SIM, M. (2004). The price of robustness. *Operations Research*, 52:35–53.

- BIELEFELDT, C. et BUSCH, F. (1994). MOTION : A new one-line traffic signal network control system. *In IEE*, éditeur : *Road Traffic Monitoring and Control*, numéro 391, pages 55–59.
- BILLIONNET, A. (2007). *Optimisation discrète*. Dunod, Paris, France.
- BOILLOT, F. et BRABAN, C. (2003). Les systèmes temps réel de commande de feux en milieu urbain. Rapport technique, INRETS-GRETIA.
- BOILLOT, F., MIDENET, S. et PIERRELÉE, J. (2006). The real-time urban traffic control system CRONOS : Algorithm and experiment. *Transportation Research*, 14:18–38.
- BOUMGHAR, S. (2000). *Nouvelle approche de l'optimisation en temps réel des feux d'un carrefour complexe isolé par la programmation linéaire mixte en nombres entiers*. Thèse de doctorat, Lamsade, Université Paris Dauphine, Paris, France.
- BRETHERTON, R., HUNT, P., ROBERTSON, D. et ROYLE, M. (1982). The SCOOT on-line traffic signal optimization technique. *In International Conference on Road-Traffic Signalling*, London, UK.
- BRETHERTON, R. D. (2007). SCOOT mc3 and current developpement. *In Transportation Research Board 86th Annual Meeting*, Washington, DC.
- BRETHERTON, R. D. et BROWEN, G. T. (1990). Recent enhancements to SCOOT - SCOOT version 2.4. *In 3rd International Conference on Road Traffic Control*, London. IEE.
- CAI, C., WONG, C. K. et HEYDECKER, B. (2009). Adaptive traffic signal control using approximate dynamic programming. *Transportation Research*, 17:456–474.
- CANTARELLA, E. et IMPROTA, G. (1984). Control system design for an individual signalized junction. *Transportation Research*, 18:147–167.
- CANTARELLA, G. et IMPROTA, G. (1988). Capacity factor or cycle time optimization for signalized junction : A graph theory approach. *Transportation Research*, 22:1–23.
- CHANG, E., CHAUDARY, N., HOGGS, G. et MESSER, C. (1986). Optimization of left turn phase sequence in signalized networks using MAXBAND 86-vol. 2. user's manual. Rapport technique, US Department of Transportation, Washington, DC.

- CHANG, E., LEI, J. et MESSER, C. (1988). Arterial signal timing optimization using PASSER ii-87 - microcomputer user's guide. Rapport technique, Texas Transportation Institute, TX.
- COHEN, S. (1990). *Ingénierie du trafic routier, éléments de théorie du trafic et applications*. Presses de l'Ecole Nationale des Ponts et Chaussées (ENPC).
- CONRAD, M., DION, F. et YAGAR, S. (1998). Real-time traffic signal optimization with transit priority : Recent advances in the signal priority procedure for optimization in real-time model. *Transportation Research Record : Journal of the Transportation Research Board*, 1634:100–109.
- CULIOLI, J. (1994). *Introduction à l'optimisation*. Ellipse.
- DÄCHTER, K., GORSKI, J. et KLAMROTH, K. (2012). An augmented weighted tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. *Computers and Operations Research*, 39:2929–2943.
- DAELLENBACH, H. et KLUYVER, C. D. (1980). Note on multiple objective dynamic programming. *Journal of the Operational Research Society*, 31:591–594.
- DANTZIG, G., ORDEN, A. et WOLFE, P. (1954). Note on linear programming. part i : The generalized simplex method for minimizing a linear form under linear inequality restraints. Rapport technique, RAND Corporation.
- DELL'OLMO, P. et MIRCHANDANI, P. (1995). REALBAND : An approach for real-time coordination of traffic flows on network. *Transportation Research Record 1494*, pages 106–116.
- DION, F. et HELLINGA, B. (2002). A rule-based real-time traffic responsive signal control system with transit priority : Application to an isolated intersection. *Transportation Research*, 36:325–343.
- DOTOLI, M., FANTI, M. P. et MELONI, C. (2006). A signal timing plan formulation for urban traffic control. *Control Engineering Practice*, 14:1297–1311.

- DUDGEON, I. et HEYDECKER, B. (1993). Calculation of signal setting to minimize delay at a junction. *In Transportation and Traffic Theory*, Berkeley, CA. International Symposium on the theory of traffic flow and transportation.
- EBERHART, R., KENNEDY, J. et SHI, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann.
- FARGES, J. et HENRY, J. (1994). P.T. priority and prodyn. *In 1st World Congress on Application of Transport Telematics and Intelligent Vehicle-Highway Systems*, volume 6, pages 3086–3093.
- FARGES, J., HENRY, J. et TUFFAL, J. (1983). The PRODYN real time traffic algorithm. *In 4th IFAC/IFIP/IFORS International Conference on Control in Transportation Systems*, pages 307–311, Baden Baden, Germany.
- GALLIVAN, S. et HEYDECKER, B. (1988). Optimising the control performance of traffic signal at a single junction. *Transportation Research*, 22:357–370.
- GARDINER, L. et VANDERPOOTEN, D. (1997). Interactive multiple criteria procedures : Some reflections. *In CLÍMACO, J., éditeur : Multicriteria Analysis*, pages 290–301. Springer Verlag, Berlin.
- GARTNER, N. (1983). Developpement of demand-responsive strategies for urban traffic control. *In Lecture Notes in Control and Information Sciences, Proceeding of the 11th IFIP Conference*, Copenhagen, Denmark.
- GOEFFRION, A. (1968). Proper efficiency and the theory of vector minimization. *Journal of Mathematical Analysis and Applications*, 22:618–630.
- HAN, B. et YAGAR, S. (1994). A procedure for real-time signal control that considers transit interference and priority. *Transportation Research*, 28:315–331.
- HEAD, K. et SEN, S. (1997). Controlled optimization of phases at an intersection. *Transportation Science*, 31:5–17.
- HEAD, L. et MIRCHANDANI, P. (2001). RHODES : A real-time traffic signal control system : architecture, algorithms, and analysis. *Transportation Research*, 9:415–432.

- ISERMANN, H. et STEUER, R. E. (1987). Computational experience concerning payoff tables and minimum criterion values over the efficient set. *European journal of operational research*, 33:9197.
- KACHROUDI, S. et BHOURI, N. (2009). A multimodal traffic responsive strategy using particle swarm optimization. In *12th IFAC Symposium on Control in Transportation (CTS'09)*.
- KARWAN, M. et VILLARREAL, B. (1982). Multicriteria dynamic programming with an application to the integer case. *Journal of optimization theory and applications*, 38.
- KIM, Y. (1990). *Development of optimization models for signalized intersections during oversaturated conditions*. Thèse de doctorat, Texas A&M University.
- LASDON, L. S., FOX, R. L. et RATNER, M. W. (1973). Nonlinear optimisation using the generalized reduced gradient methode. Rapport technique, Office of Naval Research.
- LIN, L. et TSAY, H. (1988). A new algorithm for solving the maximum progression bandwidth. In *Transportation Research Board*, Washington, DC.
- LIN, W.-H. et WANG, C. (2004). An enhanced 0-1 mixed-integer lp formulation for traffic signal control. In *IEEE Transactions on Intelligent Systems*, volume 5.
- LOWRIE, P. (1982). The sydney co-ordinated adaptive traffic system : Principles, methodology, and algorithms. In *Proceedings of the IEE conference on Road Traffic Signalling*, numéro 207, pages 67–70. IEE Pub.
- MAURO, V. et di TARANTO, C. (1990). UTOPIA. In *Proceeding of the 6th IFAC/IFIP/IFORS Symposium on Control and Communication in Transportation*.
- MIDENET, S., BOILLOT, F. et PIERRELÉE, J. (2004). Signalized intersection with real-time adaptive control : on-field assessment of CO2 and pollutant emission reduction. *Transportation Research*, 9:29–47.
- MIRCHANDANI, P., KNYAZYAN, A., HEAD, L. et WU, W. (2001). An approach towards the integration of bus priority, traffic adaptive signal control, and bus information/scheduling systems. *Lecture Notes in Economics and Mathematical System : Computer-Aided Scheduling of Public Transport*, 505:319–334.

- NAKAYAMA, H. et SAWARAGI, Y. (1983). Satisficing trade-off method for multiobjective programming. In GRAUER, M. et WIERZBICKI, A. P., éditeurs : *Lecture Notes in Economics and Mathematical Systems*, numéro 229, pages 113–122.
- PERNY, P. et VANDERPOOTEN, D. (1998). An interactive multiobjective procedure for selecting medum-term countermeasures after nuclear accidents. *Journal of multicriteria decision analysis*, (7):48–60.
- RELJIC, S. (1996). Multicriteria optimal control in a network of signalised intersections. In *International symposium on transportation and traffic theory*, numéro 13, Lyon, France. Pergamon Press.
- ROBERTSON, D. (1969). Transyt, a traffic network study tool. Rapport technique LR253, Road Research Laboratory, Crowthorne, UK.
- ROY, B. (1968). Classement et choix en présence de points de vue multiples. *Revue française d'automatique, d'informatique et de recherche opérationnelle*, 2:57–75.
- SCHÖCKER, J.-D., AHUJA, S. et BELL, M. G. (2008). Multi-objective signal control of urban junctions - framework and a london case study. *Transportation Research*, 16:454–470.
- SILCOCK, J. (1997). Designing signal-controlled junctions for group-based operation. *Transportation Research*, 31:157–173.
- STEUER, R. (1986). *Multiple criteria optimization : theory, computation and application*. John Wiley and Sons, New York.
- STEUER, R. E. et CHOO, E. (1983). An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical programming*, 26:326–344.
- UKKUSURI, S. V. (2010). A robust transportation signal control problem accounting for traffic dynamics. *Computers and operation research*, 37:869–879.
- VANDERPOOTEN, D. (1989). The interactive approach in MCDA : a technical framework and some basic conceptions. *Mathematical and Computer Modelling*, 12:1213–1220.

- VANDERPOOTEN, D. (1990). Multiobjective programming : Basic concepts and approaches. In R. SLOWINSKI, J. T., éditeur : *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*, pages 7–22.
- VINCENT, R. et PEIRCE, J. (1988). MOVA : Traffic responsive, self-optimising signal control for isolated intersections. Rapport technique, Transport and Road Research Laboratory, UK.
- WEBSTER, F. (1958). Traffic signal settings. Rapport technique 39, Road Research.
- WEY, W.-M. (2000). Model formulation and solution algorithm of traffic signal control in an urban network. *Computers, Environment and Urban Systems*, 24:355–378.
- WIERZBICKI, A. (1980). The use of reference objectives in multiobjective optimisation. In FANGEL, G. et GAL, T., éditeurs : *MCDM theory and Application, Proceedings*, numéro 177 de Lecture notes in economics and mathematical systems, pages 468–486, Hagen.
- WIERZBICKI, A. (1999). Reference point approaches. In T. GAL, T. S. et HANNE, T., éditeurs : *Multicriteria Decision Making : Advances in MCDM Models, Algorithms, Theory, and Applications*, pages 9.1–9.39. Springer Verlag.
- WILLIAMS, H. (1985). *Model Building in Mathematical Programming, 2nd ed.* Wiley, New York.
- WONG, C. et WONG, S. (2003). Lane-based optimization of signal timings for isolated junctions. *Transportation Research*, 37:63–84.
- WONG, S. (1996). Group-based optimisation of signal timings using the TRANSYT traffic model. *Transportation Research*, 30:217–244.
- WONG, S. (1997). Group-based optimisation of signal timings using parallel computing. *Transportation Research*, 5:123–139.

Annexes

Annexe A : Contraintes additionnelles pour les bus (modèle par dates)

Pour chaque bus b on rappelle que f_b désigne le tronçon qu'empruntera le bus b . Pour traduire les contraintes relatives à la présence du bus sur le tronçon (sous-section 3.1.5.2), on considère deux cas pour chaque bus b : le bus est "non arrivé", le bus est "arrivé". $\Delta T_{traverse}^b$ désignera le temps nécessaire au bus b pour traverser le carrefour.

Pour un bus b non "arrivé" :

On se place dans le cas simple où les commutations intervenant immédiatement avant T_b et après T_b correspondent à des commutations sur le tronçon f_b , où passera le bus.

Alors la contrainte correspondant au point 1 du paragraphe relatif aux contraintes supplémentaires pour les bus de la sous-section 3.1.5.2 s'écrit :

$$\sum_{c \in \{1, \dots, N_C - 2\}} (t_{c+2} - t_{c+1}) x b_c^b \geq \sum_{c \in \{1, \dots, N_C - 2\}} x b_c^b (1 - x_{c, f_b}) \times \frac{l_{f_b}(T_b)}{Q S_{f_b}}$$

Et la contrainte correspondant au point 2 du même paragraphe s'écrit :

$$\sum_{c \in C'} (t_{c+1} - T_b) x b_c^b \geq \sum_{c \in C'} x b_c^b \times x_{c, f_b} \times \max \left\{ \frac{l_{f_b}(T_b)}{Q S_{f_b}}, \Delta T_{travel}^b \right\}$$

Soit :

$$\sum_{c \in C'} (t_{c+1} - T_b) x b_c^b \geq \sum_{c \in C'} x b_c^b \times x_{c, f_b} \times \frac{l_{f_b}^0(T_b)}{Q S_{f_b}}$$

$$\sum_{c \in C'} (t_{c+1} - T_b) x b_c^b \geq \sum_{c \in C'} x b_c^b \times x_{c, f_b} \times \Delta T_{travel}^b$$

Toutes ces contraintes peuvent être linéarisées en utilisant la linéarisation de Wu (Williams, 1985).

Pour un bus b arrivé :

Si f_b est au rouge et si on est dans le cas particulier où t_1 et t_2 sont des commutations de f_b (point 1 du paragraphe de la sous-section 3.1.5.2) :

$$t_2 - t_1 \geq \frac{l_{f_b}^0}{Q S_{f_b}}$$

Si à $t = 0$, f_b est au vert et si on est dans le cas particulier où t_1 est une commutation de f_b (point 2 du paragraphe) :

$$t_1 \geq \max \left\{ \frac{l_{f_b}^0}{Q S_{f_b}}, \Delta T_{traversee}^b \right\}$$

ou encore :

$$t_1 \geq \frac{l_{f_b}^0}{Q S_{f_b}}$$

$$t_1 \geq \Delta T_{traversee}^b$$

$l_{f_b}^0$ est la longueur de la file précédant le bus b à $t = 0$. Il s'agit d'une constante, pour le programme. On se ramènera, une fois de plus, au cas vu en 3.4.4.4 pour le cas général.

Annexe B : Respect des exigences du mécanisme de régulation pour la procédure interactive sans expertise du décideur dans le cas général

Le but de cette annexe est de montrer qu'en ajoutant certaines hypothèses, les exigences énoncées en sous-section 3.7.1.2 portant sur le mécanisme de régulation de la procédure itérative proposée en sous-section 3.7.1 peuvent également être satisfaites dans le cas où λ représente une direction de projection *quelconque* mais fixe.

Nous considérons, comme dans le cas initial, que $z_{S_{TA}} = (m_{TA}, M_{NA}, M_{ECB})$ et que $\overline{z_{S_{TA}}} = (M_{TA}, m_{NA}, m_{ECB})$. Les notations et définitions utilisées dans cette annexe sont les mêmes que celles utilisées en sous-section 3.7.1.

Définition 9. On suppose que l'on utilise l'une des fonctions d'agrégation $f_{\bar{z},\lambda}$ ou $g_{\bar{z},\lambda}$ (fonctions définies en sous-section 3.7.1.4). Un critère "limitatif", relativement à une solution $\hat{z} = (\hat{z}_{TA}, \hat{z}_{NA}, \hat{z}_{ECB})$ donnée, est un critère $\hat{j} \in \arg \max_{j=TA,NA,ECB} \lambda_j (\hat{z}_j - \bar{z}_j)$.

Les exigences du mécanisme de régulation s'énoncent alors comme suit :

- Lorsque TA est un critère "limitatif" pour la solution courante, alors une modification du profil vers un profil plus proche de S_{TA} impliquera que la nouvelle solution fournie sera meilleure sur le critère TA .
- Réciproquement, si NA ou ECB sont "limitatifs", alors une modification du profil vers un profil plus proche de $\overline{S_{TA}}$ impliquera que la nouvelle solution fournie sera meilleure sur au moins un des critères NA ou ECB .

Dans notre contexte, pour que la méthode soit efficace, il est donc fait implicitement l'hypothèse que lorsque le trafic se densifie et/ou lorsqu'il y a une ou plusieurs remon-tée(s) de file(s), la valeur de TA à l'optimum augmente et TA devient rapidement limitatif, tandis que pour un trafic relativement fluide, NA ou ECB sont limitatifs.

Lemme 3. Soit $\bar{z} = (\bar{z}_1, \dots, \bar{z}_p)$ et $\bar{z}' = (\bar{z}'_1, \dots, \bar{z}'_p)$ deux points de Z et $\lambda \in (\mathbb{R}_+^*)^p$ tels que (hypothèse H_1) :

$$\begin{aligned} \bar{z}'_1 &\leq \bar{z}_1 \\ \forall j \in \{2, \dots, p\}, \bar{z}'_j &\geq \bar{z}_j \end{aligned}$$

Soit respectivement \hat{z} et \hat{z}' des solutions du problème multi-objectif (P) générés en utilisant respectivement comme fonction d'agrégation $g_{\bar{z},\lambda}$ et $g_{\bar{z}',\lambda}$.

Alors, si le critère 1 est "limitatif" pour \hat{z} (hypothèse H_2), c'est-à-dire si $1 \in \arg \max_{j \in \{1, \dots, p\}} \lambda_j (\hat{z}_j - \bar{z}_j)$, on a $\hat{z}'_1 \leq \hat{z}_1$.

Démonstration. Remarque préliminaire () :* Soit s et s' les sommets des cônes de projection respectifs des cônes $C_{\bar{z},\lambda}$ et $C_{\bar{z}',\lambda}$. Il est clair qu'on ne peut avoir : $\forall j \in \{1, \dots, p\}$, $s'_j > s_j$. Car on aurait $g_{\bar{z}',\lambda}(\hat{z}') > g_{\bar{z},\lambda}(\hat{z})$, or $\hat{z}' = \min_{z' \in Z_X} g_{\bar{z}',\lambda}(z')$.

Nous voulons montrer que $\hat{z}'_1 \leq \hat{z}_1$. Pour ce faire, nous procédons en plusieurs étapes. Dans un premier temps (partie a) nous exprimons s' en fonction d'un nouveau point \bar{z}'' ayant même valeur sur le critère 1 que le point \bar{z} . Ceci nous permet de prouver relativement aisément, dans un second temps (partie b), que si l'on suppose $s'_1 > s_1$, nous obtenons une contradiction avec la remarque préliminaire (*). Enfin nous concluons en dernière partie (partie c), remarquant que $s'_1 \leq s_1 \Rightarrow \hat{z}'_1 \leq \hat{z}_1$.

Partie a)

Soit $\mathbf{u} = (u_1, \dots, u_p) = \left(\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_p}\right)$ le vecteur directeur de la droite commune de projection de \bar{z} et \bar{z}' sur $ND(Z_X)$.

On a :

$$s = \bar{z} + t\mathbf{u}, t \in \mathbb{R}$$

Ainsi que :

$$\begin{aligned} \exists \bar{z}'' \in Z, s' = \bar{z}'' + t''\mathbf{u}, t'' \in \mathbb{R} \\ \bar{z}''_1 = \bar{z}_1, \bar{z}''_j \geq \bar{z}_j, j \in \{2, \dots, p\} \end{aligned}$$

Le deuxième point provient en particulier de l'hypothèse H_1 . En effet, on a $s' = \bar{z}' + t'\mathbf{u}$, $t' \in \mathbb{R}$ et on a aussi que :

$$\begin{aligned} \bar{z}'_1 &\leq z_1 \\ \forall j \in \{2, \dots, p\}, \bar{z}'_j &\geq \bar{z}_j \end{aligned}$$

Comme $\mathbf{u} \in (\mathbb{R}_+)^p$, il existe $\Delta > 0$ tel que $\bar{z}'_1 + \Delta u_1 = \bar{z}_1$. Soit $t'' = t' + \Delta$ et $\bar{z}'' = \bar{z}' + \Delta\mathbf{u}$.

Alors, $\bar{z}''_1 = \bar{z}'_1 + \Delta u_1 = \bar{z}_1$

De plus, $\forall j \in \{2, \dots, p\}, \bar{z}_j'' = \bar{z}_j' + \Delta u_j \geq \bar{z}_j'$ car $\mathbf{u} \in (\mathbb{R}_+)^p$. Comme $\bar{z}_j' \geq \bar{z}_j$ on a aussi que $\bar{z}_j'' \geq \bar{z}_j, j \in \{2, \dots, p\}$.

Enfin, $\bar{z}'' + t''\mathbf{u} = \bar{z}' + \Delta\mathbf{u} + t''\mathbf{u} = \bar{z}' + t'u = s'$.

Partie b)

On a, d'après la partie a et en reprenant les mêmes notations :

$$\begin{cases} s_1 &= \bar{z}_1 + tu_1 \\ s_1' &= \bar{z}_1 + t''u_1 \end{cases}$$

Supposons que $s_1' > s_1$. Alors, d'après les deux égalités précédentes on aurait $t'' > t$.

Alors, pour $j \in \{2, \dots, p\}$ on aurait,

$$\begin{aligned} s_j' - s_j &= \bar{z}_j'' + t''u_j - (\bar{z}_j + tu_j) \\ &= \bar{z}_j'' - \bar{z}_j + (t'' - t)u_j \end{aligned}$$

Comme $\bar{z}_j'' \geq \bar{z}_j, j \in \{2, \dots, p\}$ et $t'' > t$, on a alors que $s_j' > s_j, j \in \{2, \dots, p\}$.

Finalement, on a $s_j' > s_j, j \in \{1, \dots, p\}$, ce qui est impossible, en vertu de la remarque préliminaire (*).

Donc l'hypothèse $s_1' > s_1$ est fausse. C'est-à-dire qu'on a nécessairement $s_1' \leq s_1$.

Partie c)

D'après l'hypothèse $H_2, \hat{z}_1 = s_1$. Comme on a $s_1' \leq s_1$, on a donc aussi $s_1' \leq \hat{z}_1$. Or, d'après la définition de $s', \hat{z}'_j \leq s'_j, j \in \{1, \dots, p\}$.

En particulier, $\hat{z}'_1 \leq s'_1$. D'où $\hat{z}'_1 \leq \hat{z}_1$. □

Lemme 4. Soit $\bar{z} = (\bar{z}_1, \dots, \bar{z}_p)$ et $\bar{z}' = (\bar{z}'_1, \dots, \bar{z}'_p)$ deux points de Z et $\lambda \in (\mathbb{R}_+^*)^p$ tels que :

$$\begin{aligned} \bar{z}'_1 &\leq \bar{z}_1 \\ \forall j \in \{2, \dots, p\}, \bar{z}'_j &\geq \bar{z}_j \end{aligned}$$

Soit respectivement \hat{z} et \hat{z}' des solutions du problème multi-objectif (P) en utilisant respectivement comme fonction d'agrégation : $f_{\bar{z},\lambda}$ et $f_{\bar{z}',\lambda}$, avec $f_{\bar{z},\lambda}$ la fonction définie en sous-section 3.7.1.4.

Alors, si le critère 1 est "limitatif", c'est-à-dire si $\arg \max_{j \in \{1, \dots, p\}} \lambda_j (\hat{z}_j - \bar{z}_j) = 1$, on a $\hat{z}'_1 \leq \hat{z}_1$.

Démonstration. Par la construction de la fonction $f_{\bar{z}, \lambda}$, en particulier par le choix de ρ (voir section 1.2.4), \hat{z} est aussi solution de $\min_{z \in Z_X} g_{\bar{z}, \lambda}(z)$ et \hat{z}' est aussi solution de $\min_{z \in Z_X} g_{\bar{z}', \lambda}(z)$.

Le Lemme 3 implique alors que $\hat{z}'_1 \leq \hat{z}_1$. □

La Proposition 4 suivante correspond à la première exigence du mécanisme de régulation.

Proposition 4. *On suppose que le point de référence correspond à un profil tel que défini en 3.7.1.3.*

Soit $\tau, \tau' \in [0, 1]$ tels que $\tau' \geq \tau$, $z_{S_1} = (m_1, M_2, M_3, \dots, M_p)$, $z_{\bar{S}_1} = (M_1, m_2, m_3, \dots, m_p)$, $\bar{z}(\tau) = \tau \times z_{S_1} + (1 - \tau) z_{\bar{S}_1}$ et $\bar{z}(\tau') = \tau' \times z_{S_1} + (1 - \tau') z_{\bar{S}_1}$. Enfin $\lambda \in (\mathbb{R}_+^)^p$ est fixé.*

On suppose également que le critère 1 est limitatif.

Soit respectivement \hat{z} et \hat{z}' des solutions, dans l'espace des critères, de $\min_{z \in Z_X} f_{\bar{z}, \lambda}(z)$ et de $\min_{z \in Z_X} f_{\bar{z}', \lambda}(z)$ respectivement.

Alors $\hat{z}'_1 \leq \hat{z}_1$.

Démonstration. En reprenant les hypothèses et les notations de la Proposition 4, on a :

$\tau' \geq \tau$ et $m_1 \leq M_1$, donc :

$$\begin{aligned} \bar{z}_1(\tau') &= \tau' \times m_1 + (1 - \tau') M_1 \\ &= \tau' \times (m_1 - M_1) + M_1 \\ &\leq \tau \times (m_1 - M_1) + M_1 \\ &\leq \tau \times m_1 + (1 - \tau) M_1 \\ &\leq \bar{z}_1(\tau) \end{aligned}$$

$\tau' \geq \tau$ et $\forall j \in \{2, \dots, p\}$, $m_j \leq M_j$, donc :

$$\begin{aligned} \bar{z}_j(\tau') &= \tau' \times M_j + (1 - \tau') m_j \\ &\geq \tau \times M_j + (1 - \tau) m_j \\ &\geq \bar{z}_j(\tau) \end{aligned}$$

Le critère 1 étant limitatif, nous sommes donc dans les conditions du Lemme 4, d'où $\hat{z}'_1 \leq \hat{z}_1$. \square

La propriété suivante (5) correspond à la deuxième exigence du mécanisme de régulation.

Proposition 5. *En reprenant les notations de la Proposition 4, on considère maintenant que $\tau' < \tau$.*

Soit respectivement \hat{z} et \hat{z}' des solutions, dans l'espace des critères, de $\min_{z \in Z_X} f_{\bar{z}, \lambda}(z)$ et de $\min_{z \in Z_X} f_{\bar{z}', \lambda}(z)$ respectivement.

Alors $\exists j_0 \in \{2, \dots, p\}$ tel que $\hat{z}'_{j_0} \leq \hat{z}_{j_0}$.

Démonstration. Nous sommes cette fois dans les conditions exactement inverses du Lemme 3. En appelant s le sommet du cône de projection de $\bar{z}(\tau)$ et s' celui de $\bar{z}(\tau')$ il est clair que par un raisonnement analogue à celui proposé dans la preuve de la Proposition 3 (partie b), nous avons : $s'_1 > s_1$.

Supposons que $\forall j \in \{2, \dots, p\}$, $\hat{z}'_j > \hat{z}_j$. Alors on aurait : $\forall j \in \{2, \dots, p\}$, $\hat{z}_j < \hat{z}'_j \leq s'_j$.

D'où, on aurait $\hat{z}_1 \geq s'_1$, puisque \hat{z} ne peut dominer s' . Or cette dernière inégalité est impossible puisque $\hat{z}_1 \leq s_1 < s'_1$.

Finalement, on n'a donc pas $\forall j \in \{2, \dots, p\}$, $\hat{z}'_j > \hat{z}_j$. Donc $\exists j_0 \in \{2, \dots, p\}$ tel que $\hat{z}'_{j_0} \leq \hat{z}_{j_0}$. \square

Illustrations

Sur la figure 4.33, un cas de révision d'un point de référence $\bar{z}(\tau) = \tau \times z_{S_1} + (1 - \tau) z_{\bar{S}_1}$, correspondant à un profil et appartenant au segment $[z_{S_1}, z_{\bar{S}_1}]$, en dimension 2, est présenté. Le point révisé est le point $\bar{z}' = \bar{z}(\tau')$; il est choisi sur le segment $[z_{S_1}, z_{\bar{S}_1}]$, un peu plus proche de z_{S_1} que \bar{z} ne l'était ($\tau' > \tau$). On constate que le nouveau point obtenu \hat{z}' est meilleur sur le critère 1. s et s' sont les sommets des cônes de projection pour la fonction $f_{\bar{z}, \lambda}$ utilisée.

- le débit de sortie vaut deux fois le débit d’arrivée sur chaque tronçon.
- le critère qui nous intéresse principalement est le temps d’attente.
- à l’instant initial, il n’y a aucune file d’attente.
- un premier système S_1 utilise un plan de feux fixe ayant pour phasage : $\{1, 2\}$, $\{3, 5\}$, $\{2, 4\}$.
- un deuxième système S_2 utilise un plan de feux dont le phasage est partiellement modifiable : $\{1, 2\}$, $\{3, 5\}$, P , où P vaut $\{2, 4\}$ ou $\{4, 5\}$.
- les rouges de dégagements sont de durée nulle.
- les deux systèmes utilisent le même phasage initial $\{1, 2\}$, $\{3, 5\}$, $\{2, 4\}$.

Pour un système donné, on désignera par *périodes* les intervalles de temps pendant lesquels le plan de feu utilisé par le système est appliqué. La longueur d’une période est par conséquent égal au cycle du système.

D’après les hypothèses, chaque phase a la même durée ΔT , et durant chaque phase, le bilan de véhicules reçus sur chaque tronçon j vaut $\Delta V_j \in \{-\Delta V, 0, \Delta V\}$, avec $\Delta V > 0$.

Les deux systèmes ayant en particulier le même cycle, les périodes des deux systèmes sont les mêmes. On ne précisera donc pas à quel système on fait référence lors de l’utilisation du terme *période*.

Dans les paragraphes suivants, on compare l’évolution des valeurs du temps d’attente sur les trois premières périodes, selon que le système utilisé est S_1 ou S_2 .

Le système S_1 utilise trois fois le phasage $\{1, 2\}$, $\{3, 5\}$, $\{2, 4\}$

- Les valeurs des files relatives aux tronçons 1, 2, 3, 4, 5 après $\{1, 2\}$ (première période) sont respectivement : 0, 0, ΔV , ΔV , ΔV , ce qui engendre un temps d’attente, pour cette phase, de $\frac{1}{2} \times 3\Delta V\Delta T$
- Les valeurs des files relatives aux tronçons 1, 2, 3, 4, 5 après $\{1, 2\}$, $\{3, 5\}$ (première période) sont respectivement : ΔV , ΔV , 0, $2\Delta V$, 0, ce qui engendre un temps d’attente, pour cette phase, de $\frac{1}{2} \times 4\Delta V\Delta T$

- Les valeurs des files relatives aux tronçons 1, 2, 3, 4, 5 à la fin de la première période sont respectivement : $2\Delta V$, 0, ΔV , ΔV , ΔV , ce qui engendre un temps d'attente, pour cette phase, de $\frac{1}{2} \times 5\Delta V\Delta T$

A ce stade, on peut voir que le tronçon 2 a été plus favorisé que les autres puisque la file d'attente du tronçon 2 vaut déjà 0. En général, le fait d'avoir une file nulle juste avant l'ouverture du feu correspondant n'aide pas à la minimisation du *temps d'attente* (attention, le même raisonnement n'est pas forcément vrai pour les autres critères). Dans notre cas, par exemple, la prochaine phase est la phase $\{1, 2\}$, qui ne permettra de *réduire* que la file 1, la file 2 étant déjà à 0. On appellera ce cas un *zéro avant ouverture*.

Jusqu'à présent, le temps d'attente par phase augmentait régulièrement, de $\frac{1}{2} \times \Delta V\Delta T$ *par phase*. Mais avec la présence du zéro avant ouverture pour le tronçon 2, on a que, pour la deuxième période :

- Les valeurs des files relatives aux tronçons 1, 2, 3, 4, 5 après $\{1, 2\}$ (deuxième période) sont respectivement : ΔV , 0, $2\Delta V$, $2\Delta V$, $2\Delta V$, ce qui engendre un temps d'attente, pour cette phase, de $\frac{1}{2} \times 7\Delta V\Delta T$, soit une *augmentation de $\Delta V\Delta T$* par rapport à la phase précédente.

Puis :

- Les valeurs des files relatives aux tronçons 1, 2, 3, 4, 5 après $\{1, 2\}$, $\{3, 5\}$ (deuxième période) sont respectivement : $2\Delta V$, ΔV , ΔV , $3\Delta V$, ΔV , ce qui engendre un temps d'attente, pour cette phase, de $\frac{1}{2} \times 8\Delta V\Delta T$
- Les valeurs des files relatives aux tronçons 1, 2, 3, 4, 5 à la fin de la deuxième période sont respectivement : $3\Delta V$, 0, $2\Delta V$, $2\Delta V$, $2\Delta V$, ce qui engendre un temps d'attente, pour cette phase, de $\frac{1}{2} \times 9\Delta V\Delta T$
- *A la fin de la prochaine phase $\{1, 2\}$ (troisième période), à cause du zéro avant ouverture, on aura une augmentation de $\Delta V\Delta T$ au lieu de $\frac{1}{2} \times \Delta V\Delta T$: les valeurs des files relatives aux tronçons 1, 2, 3, 4, 5 après $\{1, 2\}$ (troisième période) sont respectivement : $2\Delta V$, 0, $3\Delta V$, $3\Delta V$, $3\Delta V$, ce qui engendre un temps d'attente,*

pour cette phase, de $\frac{1}{2} \times 11\Delta V\Delta T$, soit une augmentation de $\Delta V\Delta T$ par rapport à la phase précédente.

- Les phases suivantes de la troisième période ne font pas apparaître de zéro avant ouverture, et on observe une augmentation de $\frac{1}{2} \times \Delta V\Delta T$ à chaque fois.

Le système S_2 peut utiliser un phasage $\{1, 2\}$, $\{3, 5\}$, $\{4, 5\}$ à la deuxième période par exemple

La première période et les deux premières phases de la deuxième période sont identiques à ceux du premier système. Le système S_2 ne peut donc pas empêcher la présence du premier zéro avant ouverture, déjà observé avec le système S_1 .

En revanche, la troisième phase de la deuxième période peut être choisie égale à $\{4, 5\}$. Dans ce cas :

- Les valeurs des files relatives aux tronçons 1, 2, 3, 4, 5 à la fin de la deuxième période sont respectivement : $3\Delta V$, $2\Delta V$, $2\Delta V$, $2\Delta V$, 0, ce qui engendre un temps d'attente, pour cette phase, de $\frac{1}{2} \times 9\Delta V\Delta T$. Pour le moment, on a les mêmes temps d'attente qu'avec le système S_1 . Cependant, la prochaine phase étant la phase $\{1, 2\}$, on note qu'il n'y a pas de zéro avant ouverture à ce stade, contrairement au cas S_1 .
- *A la fin de la prochaine phase $\{1, 2\}$ (troisième période), on aura une augmentation de $\frac{1}{2} \times \Delta V\Delta T$ (au lieu de $\Delta V\Delta T$ pour S_1) : les valeurs des files relatives aux tronçons 1, 2, 3, 4, 5 après $\{1, 2\}$ (troisième période) sont respectivement : $2\Delta V$, ΔV , $3\Delta V$, $3\Delta V$, ΔV , ce qui engendre un temps d'attente, pour cette phase, de $\frac{1}{2} \times 10\Delta V\Delta T$, soit une augmentation de $\frac{1}{2} \times \Delta V\Delta T$ par rapport à la phase précédente.*
- Les autres phases ne font pas apparaître de zéro avant ouverture, et on observe une augmentation de $\frac{1}{2} \times \Delta V\Delta T$ à chaque fois.

Au final, le système S_2 permet de résoudre le problème sur trois périodes avec un temps d'attente qui vaut $\frac{1}{2} \times \Delta V\Delta T$ moins que le temps d'attente obtenu par S_1 . Si l'on

continuait l'analyse sur les périodes suivantes, on verrait que l'utilisation de la phase $\{4, 5\}$ fait aussi apparaître un zéro avant ouverture ou bout d'un moment, qui porte sur le tronçon 5. Ce zéro peut être périodiquement annulé par l'utilisation de la phase $\{2, 4\}$, et on pourrait observer une alternance entre l'utilisation de la phase $\{2, 4\}$ et la phase $\{4, 5\}$, d'où l'apparition d'un zéro avant ouverture seulement *une période sur deux*. Quant au système S_1 , il est voué à générer un zéro avant ouverture *à chaque période*.

Le problème aurait été identique si on avait considéré que le premier phasage était $\{1, 2\}, \{3, 5\}, \{4, 5\}$ pour les deux systèmes et que le deuxième phasage était $\{1, 2\}, \{3, 5\}, \{2, 4\}$ pour le système S_2 , et on aurait observé le même phénomène, en faveur du système S_2 , les zéros avant ouverture portant cette fois sur la file relative au tronçon 5.

Enfin, un raisonnement similaire peut être mené si le "critère prioritaire" est cette fois le nombre d'arrêts (on veut maximiser le nombre de zéros avant ouverture) ou bien le critère relatif aux transports en commun. Dans tous les cas, il est utile de pouvoir contrôler la présence des zéros avant ouverture et le changement de phasage s'avère être utile.

Il s'agit certes d'un exemple simpliste, et on pourrait objecter qu'en réalité la durée des phases est variable, donc un système de type S_1 pourrait jouer sur cette durée pour minimiser le nombre de zéros avant ouverture, en prenant des durées d'ouverture très courtes, par exemple, et de ce fait n'écoulant jamais entièrement les files. Cependant cela engendrerait plus d'ouvertures pour écouler la même quantité de véhicules dans le même temps. Or, dans les systèmes réels, on prend aussi en compte les rouges de dégagement, et le fait d'augmenter le nombre d'ouvertures engendrerait un coût supplémentaire sur l'ensemble des critères car cela augmenterait le nombre de rouges de dégagement. Donc le fait de s'extraire d'un phasage imposé, avec des hypothèses moins restrictives et plus proches de notre problématique, est également pertinent.

En conclusion, on retiendra qu'un changement de phasage en temps réel, *même dans le cas où il n'y a pas d'asymétrie entre les différentes valeurs de débits d'arrivées des différents tronçons*, peut s'avérer pertinent.

Résumé

Afin de répondre à la problématique de la régulation multi-objectif et multi-mode des carrefours à feux, nous proposons trois modèles de programmation linéaire mixte en nombres entiers constituant les moteurs d'un système de régulation pleinement adaptatif, ainsi que deux procédures interactives d'optimisation multi-objectif permettant d'adapter itérativement une "politique de régulation" à la situation de trafic. Les critères pris en compte, tous à minimiser, sont le temps d'attente et le nombre d'arrêts des véhicules particuliers, et un critère dédié aux transports en commun permettant de fixer un temps d'attente souhaité pour chaque bus. Des expérimentations ont montré qu'un des trois modèles, dit hybride, se démarque positivement des deux autres. Ce modèle a alors été mis en œuvre avec une des deux procédures interactives, permettant de contrôler un trafic simulé sur une période d'une heure dans différents scénarios types, et comparé à un système de régulation semi-adaptatif.

Abstract

In order to answer the multi-objective and multi-mode adaptive traffic control problem, we propose three models of mixed integer linear programming, usable with two multi-objective optimization interactive methods, allowing to adapt a "traffic control policy" iteratively to the current traffic situation. The considered criteria, all of them to be minimized, are the total waiting time and the number of stops for private vehicles and a criterion dedicated to public transports allowing to set a target waiting time for every bus. Experiments showed that one of the three models, called hybrid model, distinguishes itself positively from the others. This model was implemented with one of the two interactive methods, allowing to control a traffic simulated over one hour in different scenarios, and was compared to a semi-adaptive traffic control system.