

---

# Intersections intelligentes pour le contrôle de véhicules sans pilote

## Coordination locale et optimisation globale

Mohamed Tlig<sup>1</sup>, Olivier Buffet<sup>1</sup>, Olivier Simonin<sup>2</sup>

1. Université de Lorraine, INRIA, CNRS  
Nancy, France  
prenom.nom@loria.fr

2. Université de Lyon, INSA Lyon, INRIA, CITI,  
69621 Villeurbanne, France  
prenom.nom@insa-lyon.fr

---

**RÉSUMÉ.** Dans cet article, nous nous intéressons à la gestion du trafic au sein d'un réseau routier pour véhicules sans pilote, fondée sur des prises de décision au niveau des intersections. Les intersections perçoivent et contrôlent les véhicules à l'approche, au travers de communications infrastructure-à-véhicule, afin d'assurer un passage coordonné et sans arrêt. Nous proposons une approche originale à deux égards : d'une part, elle explore un principe de passage en alternance des flux aux intersections et, d'autre part, elle propose des algorithmes distribués permettant l'optimisation du trafic global du réseau. Nous présentons successivement les choix de modélisation, les algorithmes et l'étude en simulation de leurs performances comparées à des approches existantes.

**ABSTRACT.** In this paper, we address the problem of traffic management in a road network for driverless vehicles, based on local decisions in each intersection. Intersections perceive and control approaching vehicles, through infrastructure-to-vehicle communications, in order to ensure a coordinated and stop-free crossing. Our approach is original in two ways: on the one hand, it explores a principle alternating flows at intersections, and, on the other hand, it proposes distributed algorithms that optimize the global traffic in the network. We present the modeling choices, the algorithms, and the simulation study of our approach, and we compare its performances with existing approaches.

**MOTS-CLÉS :** systèmes multi-agents, véhicules autonomes, simulation de trafic.

**KEYWORDS:** multi-agent systems, autonomous vehicles, traffic simulations.

---

DOI:10.3166/RIA.30.353-382 © 2016 Lavoisier

## 1. Introduction

La demande croissante de mobilité dans nos sociétés a engendré de nouveaux problèmes de transport. A partir de la deuxième moitié du 20<sup>e</sup> siècle, les phénomènes de congestion du trafic sont apparus, connus sous le nom d'embouteillages. Leur formation est due au fait que de nombreux véhicules veulent utiliser les mêmes infrastructures routières au même moment.

Réguler le trafic dans un réseau routier est au moins un problème de *satisfaction* puisqu'il faut satisfaire les besoins des usagers de la route, c'est-à-dire qu'ils atteignent leur destination. Nous nous intéressons dans cet article à la gestion des intersections, car elles sont le premier facteur de perturbations du trafic. Une façon simple et courante de gérer le trafic aux intersections est, à un instant donné, d'arrêter une partie des flux pour ne laisser passer que les autres, comme le permettent les feux de signalisation dans les villes. Toutefois, il a été montré que cette stratégie est coûteuse en termes de temps (Fok *et al.*, 2012) puisqu'elle oblige l'arrêt systématique des flux. La régulation du trafic est en effet aussi un problème d'optimisation puisqu'on cherche généralement à trouver les règles de fonctionnement du réseau (au sens large) qui induisent la meilleure qualité de trafic possible. Par exemple, le fluidifier permet de réduire les temps d'attentes de chaque véhicule.

La régulation du trafic peut être réalisée en définissant les règles de comportement des véhicules (leurs conducteurs) et des systèmes de signalisation placés dans l'environnement (tels que les marquages au sol, les panneaux stop, les panneaux priorité et les feux de circulation). Le code de la route définit ainsi le comportement que doit suivre un conducteur en fonction des actions des autres véhicules et de la signalisation environnante. Ce code assure la sécurité de chacun, mais pas nécessairement d'arriver à destination au plus vite ou à moindre coût. Nous identifions différents leviers sur lesquels jouer pour améliorer le trafic sur un réseau : (i) les *besoins des utilisateurs* (en répartissant mieux les zones d'habitation, de commerce et d'industrie), (ii) le *réseau*, en ajoutant des routes ou des voies, mais avec des implications financières et environnementales (bruit, pollution, gêne visuelle) fortes, (iii) la *disposition de la signalisation*, (iv) le *contrôle* (les règles de fonctionnement) *des signalisations* actives/dynamiques (feux de circulation), et (v) le *contrôle* (le comportement) *des véhicules*.

Dans cet article, nous supposons que les besoins des utilisateurs, le réseau et la disposition de la signalisation sont fixes. Par contre, nous abordons le contrôle des véhicules, supposés ici sans pilote, c'est-à-dire contrôlés par une IA embarquée ou située dans l'infrastructure connectée. Nous considérons en effet des systèmes de transport intelligents, tels qu'il en existe déjà dans certains environnements, comme les grands ports maritimes, où le transport des conteneurs peut être assuré par des véhicules sans pilotes.<sup>1</sup> L'utilisation de véhicules autonomes pour le transport des personnes est envisagé à plus long terme, sur les autoroutes et dans les villes. Au cœur des villes, la gestion des intersections reste le principal problème.

---

1. Utilisation des AGV dans les ports de Rotterdam, Düsseldorf et Hambourg.

La gestion des intersections est aujourd'hui envisagée à l'aide de véhicules pouvant communiquer avec une infrastructure intelligente, équipée de capteurs et de capacités de prise de décision. Ce contexte constitue nos hypothèses de travail.

Dans cet article, nous présentons une approche qui cherche à optimiser le trafic en évitant aux véhicules de s'arrêter aux intersections (permettant un gain de temps et d'énergie). Cette approche, nommée « passage sans arrêt », repose sur un mécanisme de synchronisation temporelle permettant le passage alterné des véhicules provenant de routes se croisant. Dans cet article nous ne traitons pas le changement de voie, c'est-à-dire que les véhicules traversent les intersections en restant sur la même route, suivant une trajectoire prédéfinie. Ce modèle a été développé dans le cadre de la thèse de Tlig (2015), effectuée dans l'équipe Inria MAIA au LORIA. Elle a été financée par le projet européen InTraDE, visant à étudier un système de transport intelligent dédié au transport automatique des conteneurs dans les grands ports maritimes (<http://www.intrade-nwe.eu/fr>). Ce contexte a permis de définir notre problématique et de dimensionner nos expérimentations (réseau de routes, véhicules et flux).

La première partie de cet article présente le mécanisme de croisement sans arrêt. Pour ce faire, en tenant compte des paramètres spatiaux d'une intersection (taille, angle entre les routes), nous déterminons l'intervalle de temps minimal entre deux véhicules d'un flux pour permettre le passage entre eux d'un véhicule d'un autre flux. Ceci conduit à placer, en chaque intersection, un agent de contrôle chargé de calculer et communiquer aux véhicules des profils d'accélération assurant l'alternance des flux. Entre les intersections, la vitesse des véhicules est contrôlée par le modèle (Scheuer *et al.*, 2009) qui permet d'éviter toute collision avec le véhicule précédent.

Dans la deuxième partie, nous étudions la possibilité de coordonner les agents de contrôle des différentes intersections pour améliorer le trafic global du réseau. Chaque intersection (ou agent de contrôle) cherche à optimiser le retard ou l'énergie dépensée des véhicules, en tenant compte des informations disponibles sur ses intersections voisines. Ceci nous conduit à explorer plusieurs techniques d'optimisation décentralisée. Nous montrons expérimentalement que les solutions obtenues tendent à produire des « ondes vertes » qui facilitent la traversée rapide d'une succession d'intersections.

En section 2 nous commençons par présenter et discuter les approches existantes pour la gestion des intersections, en soulignant leurs avantages et leurs limites. Ensuite, en section 3, nous définissons un mécanisme de passage alterné et sans arrêt des véhicules, qui est exécuté par un agent « intersection » prenant le contrôle des véhicules. Nous montrons que l'application de ce mécanisme en chaque intersection d'un réseau permet d'améliorer le trafic, en termes de retard, comparativement à une approche par feux de circulation et à une approche opportuniste inspirée d'un modèle référence de la littérature. En section 4, nous présentons une optimisation de cette approche fondée sur différents algorithmes de recherche locale, pour passer d'une synchronisation en chaque intersection à une synchronisation d'un réseau d'intersections. Enfin, en section 5, nous montrons en simulation l'efficacité des solutions proposées.

## 2. Approches existantes pour la gestion d'intersections

Bazzan (2009) présente un aperçu des approches du domaine de l'ingénierie du trafic pour surmonter les problèmes de congestion aux intersections. Dans ce travail, l'auteur choisit une classification principalement historique, à savoir à travers trois grandes classes. (1) *Les approches classiques* sont fondées sur des règles de coordination entre intersections. Elles sont les plus déployées parce qu'elles sont les premières proposées, mais aussi parce qu'elles sont simples et ne nécessitent pas beaucoup de technologie (TRANSYT (Robertson, 1969), SCOOT (Hunt *et al.*, 1981), SCATS (Lowrie, 1982), et PROLYN (Henry *et al.*, 1984)). (2) *Les approches avancées* sont fondées sur la physique, l'optimisation, l'intelligence artificielle (systèmes multi-agents) ou l'apprentissage. (3) *Les approches pour les véhicules autonomes*, les plus récentes, sont généralement utilisées dans un environnement fermé et uniquement pour des véhicules sans pilote. Le tableau 1, inspiré de (Bazzan, 2009), donne une synthèse de sa classification. On y remarque que presque toutes les solutions rencontrées optimisent une ou des intersections indépendamment de ce qui se passe au niveau du réseau. Peu de travaux s'intéressent à l'optimisation globale des intersections. C'est le cas concernant les approches dites avancées, et il n'en existe aucune à notre connaissance dédiée aux véhicules intelligents autonomes (IAV).

Tableau 1. Résumé des solutions identifiées

	Intersections Isolées	Intersections connectées (optim. locale)	Optim. globale
Approches de Référence	PROLYN		TRANSYT SCOOT SCATS
Physi.	(Gershenson, 2004) (Guler <i>et al.</i> , 2014)		(Brockfeld <i>et al.</i> , 2001) (Köhler <i>et al.</i> , 2005) (Mohring <i>et al.</i> , 2006)
SMA	(Bull <i>et al.</i> , 2004) (Donicic <i>et al.</i> , 2005)	(Kosonen, 2003) (Sánchez, Aguirre, 2007) (France, Ghorbani, 2003) (Bhourri <i>et al.</i> , 2011) (Hounsell, Shrestha, 2012) (Rochner <i>et al.</i> , 2006)	(De Oliveira, Bazzan, 2006) (De Oliveira, Bazzan, 2007)
Approches Avancées			
Apprent.	(Da Silva <i>et al.</i> , 2006)	(Camponogara, Kraus Jr, 2003) (Nunes, Oliveira, 2004) (Steingrover <i>et al.</i> , 2005) (Wiering, 2000) (Ferreira, Khosla, 2000) (Richter <i>et al.</i> , 2007) (Yin <i>et al.</i> , 2015)	(Bazzan, 2005)
Approches pour IAV (Intel. Autonom. Veh.)	(Dresner, Stone, 2008) (Balan, Luke, 2006) (Naumann, Rasche, 1997) (Carlino <i>et al.</i> , 2013) [ Notre approche : intersection ]		[ Notre approche : réseau ]

Par ailleurs, nous proposons ci-après une grille d'analyse pour essayer de mieux cerner les liens entre les approches existantes. Sa réalisation part du constat que la gestion du trafic est souvent traitée comme un problème d'optimisation et, plus précisément, de contrôle. Cela amène à identifier les entrées/sorties des contrôleurs, les critères d'optimisation, la localisation des contrôleurs, et leur forme :

– *Les perceptions* : les feux de signalisation les plus simples ne perçoivent rien, et ne peuvent alors s’adapter aux conditions du trafic (Robertson, 1969). Dans des approches plus évoluées (Hunt *et al.*, 1981 ; Lowrie, 1982), des feux de signalisation peuvent être reliés à des capteurs (tels que boucles magnétiques sous voies ou caméras) permettant d’estimer en continu le trafic (Rochner *et al.*, 2006 ; Bazzan, 2005), et même de prédire les arrivées des véhicules à court terme (Henry *et al.*, 1984).

*Adaptation à des conditions moyennes ou instantanées* : On peut en fait distinguer les contrôleurs selon qu’ils s’adaptent (1) à des conditions moyennes de trafic (flux moyens des véhicules) (Robertson, 1969 ; Hunt *et al.*, 1981 ; Gershenson, 2004) ou (2) à des situations instantanées (un état particulier du système) (Wiering, 2000 ; Steingrover *et al.*, 2005 ; Naumann, Rasche, 1997 ; Henry *et al.*, 1984).

– *Les variables de contrôle* : la majorité des travaux considèrent les allumages des feux de signalisation comme variables de contrôle (Brockfeld *et al.*, 2001 ; Steingrover *et al.*, 2005 ; France, Ghorbani, 2003), variant l’ordre des phases, la durée des cycles et les déphasages entre intersections. Cependant, quelques travaux considèrent les véhicules eux-mêmes comme variables de contrôle, bien que leurs parcours ne peuvent être modifiés (Dresner, Stone, 2008 ; Naumann *et al.*, 1997 ; Carlino *et al.*, 2013). Dans ces cas, ils peuvent être ralentis ou accélérés pour passer au bon moment à un certain endroit, principe que nous utilisons dans notre proposition.

– *Espace discret ou continu* : certaines approches emploient des modèles de l’environnement discrets (Brockfeld *et al.*, 2001 ; Gershenson, 2004 ; Wiering, 2000). Cela facilite la simulation mais a le défaut de contraindre les déplacements des véhicules à des arrêts et des accélérations brusques et donc peu réalistes. Les autres approches, soit emploient un modèle continu de l’environnement (Kosonen, 2003 ; Nunes, Oliveira, 2004 ; Köhler *et al.*, 2005), soit font abstraction de la notion d’espace (Rochner *et al.*, 2006 ; Balan, Luke, 2006).

– *Les critères d’optimisation* : la plupart des auteurs se sont concentrés sur la minimisation du retard moyen des véhicules. Ils considèrent le retard de tous les véhicules, à chaque instant, en comparant leurs temps de parcours à un temps calculé théoriquement (Mohring *et al.*, 2006 ; Köhler *et al.*, 2005 ; Bhourri *et al.*, 2011 ; Hounsell, Shrestha, 2012). Nous faisons de même dans cet article. D’autres travaux visent à minimiser les temps d’attentes des véhicules (Ferreira, Khosla, 2000 ; Bazzan, 2005 ; Camponogara, Kraus Jr, 2003), ou les taux d’occupation des voies –nombre de véhicules par rapport à la taille du réseau– (Da Silva *et al.*, 2006).

– *Architectures centralisées ou décentralisées* : le plus souvent les contrôleurs sont complètement décentralisés, c’est-à-dire situés au niveau de chaque intersection (parfois de chaque feu). En effet, un contrôleur centralisé serait trop complexe et non robuste aux problèmes de communications. Cependant, il existe des approches que nous qualifions de « centralisées par zones » (Robertson, 1969 ; Mohring *et al.*, 2006 ; Köhler *et al.*, 2005), lesquelles, pour un ensemble de feux de circulation, affectent un unique contrôleur en charge de réguler cette zone. Un cas particulier des architectures décentralisées est celui où les véhicules, autonomes, prennent eux-mêmes les décisions, l’infrastructure servant au plus à relayer des communications.

– *Algorithmes d'optimisation* : ils entrent dans trois grandes classes : (i) *les algorithmes heuristiques*, souvent fondés sur des règles prédéfinies par ses concepteurs, et qui ne garantissent pas d'optimalité (même locale) (Gershenson, 2004 ; France, Ghorbani, 2003) ; (ii) *les algorithmes d'optimisation boîte noire*, qui optimisent un critère sans exploiter le fait qu'il s'agit d'un problème de contrôle, par exemple l'algorithme Hill-Climbing (Robertson, 1969 ; Hunt *et al.*, 1981 ; Lowrie, 1982) ou l'optimisation mathématique (Brockfeld *et al.*, 2001 ; Guler *et al.*, 2014) ; (iii) *les algorithmes de contrôle*, qui optimisent explicitement la loi de contrôle d'un système dynamique, par exemple l'apprentissage par renforcement (Q-learning) (Ferreira, Khosla, 2000 ; Camponogara, Kraus Jr, 2003), la planification en-ligne (Henry *et al.*, 1984 ; Yin *et al.*, 2015), ou la théorie des jeux (Bazzan, 2005).

Nous nous intéressons dans cet article aux véhicules sans pilote, à savoir, le cas où les variables de contrôle correspondent à des véhicules plutôt qu'à des signaux (éventuellement les deux). Trois importantes catégories de solutions pour véhicules sans pilote (parfois qualifiés d'autonomes) sont les suivantes :

– Balan et Luke (2006) proposent un contrôleur fondé sur *l'historique des attentes des véhicules* et qui vise à fournir une sorte d'équité globale. Ils se sont inspirés de travaux antérieurs, tels que ceux de Wiering (2000) chez lequel les intersections connaissent les durées de voyage des conducteurs.

– Les *approches par réservation*, introduites par Dresner et Stone (2004 ; 2005 ; 2008), sont fondées sur la gestion de chaque intersection par un agent. Chaque véhicule voulant passer doit réserver un temps de passage et une route à prendre à travers une trajectoire discrète dans l'intersection. L'avantage de cette approche est que, si plusieurs véhicules veulent passer et que leurs trajectoires ne se croisent pas, on peut tous les satisfaire. Dans le cas contraire, on est obligé de donner la priorité à un véhicule par rapport aux autres.

– Les *approches décentralisées* introduites par Naumann et Rasche (1997) ; Naumann *et al.* (1997) reposent essentiellement sur la communication et la négociation entre véhicules afin de déterminer l'ordre de passage dans l'intersection. Cette approche est connue pour ses limites qui dépendent du nombre de véhicules voulant négocier leurs passages par l'intersection.

Le tableau 5 (page 382) présente un certain nombre de travaux selon cette grille d'analyse. Nous y situons notre approche en distinguant (i) le contrôle d'une intersection seule de (ii) l'optimisation du réseau complet. Au regard des tableaux 1 et 5, nos travaux se positionnent, en complément, sur l'optimisation des réseaux de transport d'IAV, tout d'abord d'un point de vue local, puis, ensuite, d'un point de vue global.

### 3. Passage alterné sans arrêt dans les intersections : approche Alt

Les travaux rencontrés dans l'état de l'art reposent souvent sur l'optimisation de feux de circulation, ou sur la planification de trajectoires de véhicules, mais avec toujours le risque que des véhicules doivent s'arrêter. Notre approche vise en premier lieu à éviter d'arrêter ainsi les véhicules. Elle repose sur une méthode de synchroni-

sation locale des véhicules, c'est-à-dire décentralisée au niveau des intersections, par opposition à un système unique couvrant le réseau complet. Elle consiste à faire passer *de façon alternée et sans arrêt* des véhicules de chaque route en respectant une inter-distance suffisante afin d'éviter les collisions, elle est dénommée Alt. La figure 1 illustre ce principe de passage alterné entre deux routes A et B, selon une période fixe ( $2T_C$ , voir section 3.2).

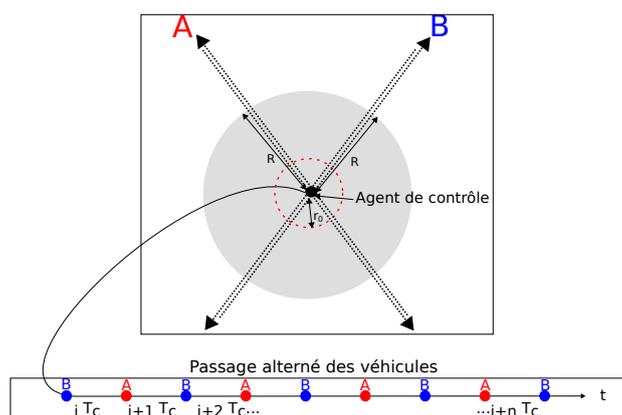


Figure 1. Principe de synchronisation d'une intersection

Dans cette approche, la gestion de chaque intersection est réalisée par un agent, dit agent de contrôle. Son rôle principal est de communiquer avec les véhicules afin de gérer leur passage. Cet agent dispose d'un rayon d'action limité qui définit la distance en dessous de laquelle il commence à communiquer avec les véhicules s'approchant de l'intersection pour réguler leurs vitesses. Nous détaillons dans ce qui suit comment déterminer la période d'alternance entre les routes (sec. 3.1), comment l'agent décide de l'ordre de passage des véhicules à son approche (sec. 3.2), et comment il détermine les profils d'accélération qu'ils doivent appliquer (sec. 3.3). Nous évaluons expérimentalement cette approche (sec. 3.5) par comparaison avec un modèle par réservation fondé sur (Dresner, Stone, 2008) (sec.3.4) et avec le modèle par feux.

### 3.1. Période minimale de croisement

Cette section détermine les conditions permettant le croisement en alternance de véhicules sur deux routes faites de deux voies opposées chacune, notées  $(A, C)$  et  $(B, D)$  dans la figure 2. On note  $\lambda$  la distance entre deux voies opposées de la même route. Les véhicules de deux voies opposées devraient traverser l'intersection en même temps, de sorte que l'ordre de passage soit  $(A_0, C_0), (B_0, D_0), (A_1, C_1), (B_1, D_1), \dots$  Tous les véhicules sont supposés *autonomes* et *identiques*, de largeur  $l$ , de longueur  $L$ , et de vitesse par défaut  $v$ .

La zone de croisement est définie par le losange correspondant à l'espace partagé par les deux routes. Pour une voie donnée, les distances sont calculées par rapport au

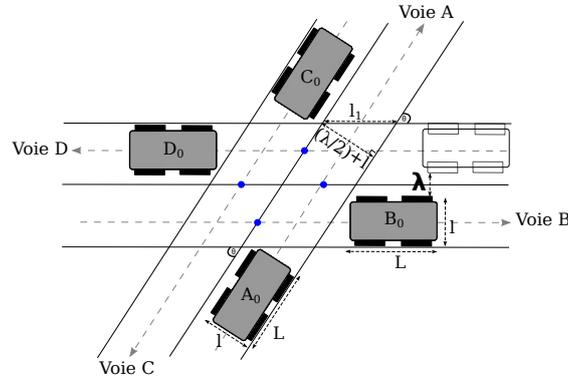


Figure 2. Zone de croisement de deux routes avec 4 flux

milieu du segment de la voie à l’intérieur de la zone de croisement, comme illustré par quatre points (bleus) sur la figure 2.

La figure 3 illustre une période complète de croisement entre les deux routes. Nous divisons cette période en quatre instants clés (les figures 3.1 à 3.4). Pour traverser complètement la zone de croisement, un véhicule prend un temps  $(2l_1 + L)/v$ . Sur la figure 3, cette demi-période se détermine aisément en évaluant le temps entre les instants 2 et 4. Quelques calculs donnent ainsi la période  $T$ , c’est-à-dire la durée minimale entre deux passages de véhicules d’un même flux (d’une même voie) :

$$T = 2 \left( \frac{2(l + \frac{\lambda}{2})}{\sin \theta} + L \right) / v. \tag{1}$$

Notons que la distance entre les points de référence, ainsi que l’angle de l’intersection, induisent les contraintes suivantes sur les temps de transition entre instants clefs : (i)  $T_{1,2} = T_{2,3}$ , (ii)  $T_{3,4} = T_{4,1}$  (par symétrie), et (iii)  $T_{1,2} \neq T_{3,4}$ . Cette dernière implique un léger déphasage entre les deux routes qu’on appellera  $\phi_\lambda$  (voir le cercle temporel au milieu de la figure 3).  $\phi_\lambda$  est la différence entre le quart de la période ( $\frac{T}{4}$ ) et la durée  $T_{4,1}$ . Avec le véhicule  $C$  comme véhicule de référence, d’après la figure 3 (droite), nous avons  $T_{4,1} = (l_1 + \frac{L}{2} - h)/v$  et  $\frac{T}{4} = (l_1 + \frac{L}{2})/v$ , d’où :  $\phi_\lambda = h/v = \left( \frac{l}{2 \tan(\theta)} \right) / v$ . Pour remédier à ce problème, il faut tenir compte de ces déphasages dans l’algorithme lors du calcul des instants de passage (section suivante). Ainsi, si l’on associe un signal périodique à chacune de ces deux routes, les phases de ces signaux ( $\phi_{B,D}$  et  $\phi_{A,C}$ ) doivent vérifier  $\phi_{B,D} \equiv \phi_{A,C} + \frac{T}{2} - 2\phi_\lambda [T]$ . Note : par la suite, (1) on omettra ce déphasage lié à  $\phi_\lambda$  (même s’il est considéré en pratique), et (2) l’une des phases servira de référence (l’autre se déduisant directement). En pratique pour tout  $\theta$ ,  $T$  sera augmentée de  $\epsilon$ , une marge de sécurité d’inter-distance.

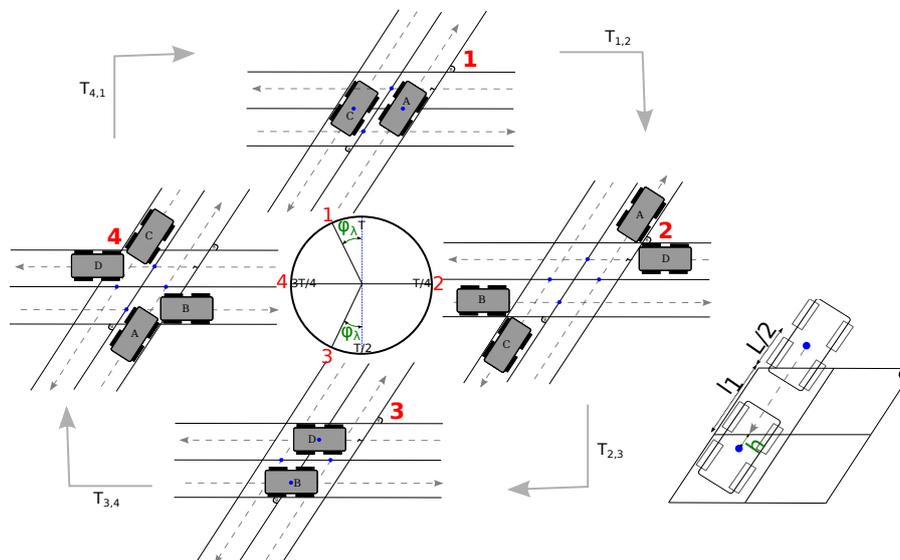


Figure 3. Croisement de deux routes à deux voies : (gauche) cycle de 4 instants clefs ; (droite) illustration du décalage (longueur  $h$ ) entre point à mi-traversée d'un véhicule et point de référence de sa voie

### 3.2. Algorithme de détermination des instants de passage

Pour ne pas arrêter les flux, notre approche fait passer les véhicules alternativement entre les deux routes, ce qui conduit à organiser le passage des véhicules toutes les demi-périodes  $T_c = T/2$ . Les véhicules d'une route passeront aux multiples pairs de  $T_c$  et les véhicules de l'autre route aux multiples impairs, en minimisant les changements de vitesse pour réduire la consommation d'énergie. On définit ainsi, pour chaque intersection, la *parité*  $p \in \{0, 1\}$  de chacune des deux routes la traversant (et donc des voies correspondantes). La figure 1 illustre ce principe de passage alterné.

L'agent de contrôle est chargé de calculer les consignes de vitesse des véhicules quand ils se trouvent dans son rayon d'action  $R$ , c'est à dire à une distance comprise entre  $r_0$  et  $R$  ( $0 < r_0 < R$ ),  $r_0$  étant la distance sous laquelle le véhicule devra rouler à la vitesse de croisière (voir figure 4). Chaque flux entrant peut être contrôlé indépendamment des autres. L'algorithme 1 présente le contrôle d'un flux entrant par l'agent étant donnée la parité  $p$  de ce flux (sans la correction de  $\phi_\lambda$ ). Il est exécuté à fréquence régulière, à laquelle il reçoit des informations sur les véhicules (via des capteurs ou des communications V2I).

L'algorithme traite toujours les véhicules du plus proche de l'intersection au plus éloigné pour assurer une affectation respectant le même ordre de passage. L'agent de contrôle a besoin de percevoir, pour chaque véhicule  $i$ , sa distance à l'intersection ( $r_i$ )

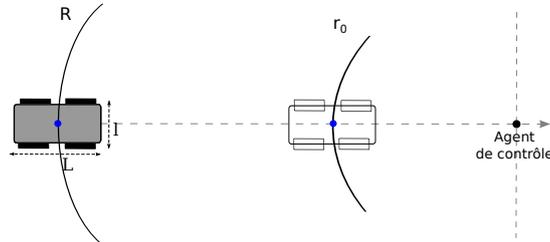


Figure 4. Un véhicule en direction de l'intersection entre  $R$  et  $r_0$

---

**Algorithme 1.** Contrôle d'un flux entrant sur un voie par l'agent de contrôle

---

**Entrées :** parité du flux  $p \in [0, 1]$ , pas de simulation actuel  $t$ ,  
liste des véhicules  $i \in lv$  perçus à une distance  $r_i \in ]r_0; R[$  et ordonnés en  
partant du plus proche

- 1  $last = 0$
- 2 **pour** chaque véhicule  $i \in lv$  **faire**
- 3      $(r_i, v_i) = rayonEtVitesse(i)$
- 4     **si**  $npp_i = -1$  /\*  $i$  vient d'entrer dans la zone contrôlée \*/
- 5     **alors**
- 6          $t_a = t + r_i/v_i$  ;  $npp_i = \lceil t_a/T - p/2 \rceil$
- 7         **si**  $npp_i \leq last$  **alors**  $npp_i = last + 1$
- 8      $last = npp_i$
- 9      $t_p = npp_i * T + p * T_c - r_0/v_{crois}$
- 10     /\* Calcul et envoi au véhicule  $i$  de son profil d'accélération. \*/  
        $EnvoyerProfilAcceleration(i, ProfilAcceleration(r_i, v_i, t_p))$

---

et sa vitesse ( $v_i$ ), afin de calculer (1) l'instant de passage  $t_p$  à l'entrée du cercle de rayon  $r_0$ , et (2) le profil d'accélération à adopter entre  $R$  et  $r_0$ .

On ne calcule le numéro de période de passage de  $i$  qu'une fois, en testant s'il a la valeur par défaut :  $npp_i = -1$  (ligne 4). Dans ce cas, la ligne 6 détermine l'instant de passage *a priori* au milieu de l'intersection, en maintenant la vitesse  $v_i$ .  $npp_i$  est alors le plus proche multiple (de parité  $p$ ) de  $T$  (ligne 6)<sup>2</sup>. La variable  $last$ , qui contient le dernier numéro de passage attribué à un véhicule, permet d'éviter les doubles attributions (ligne 7). Finalement, l'instant de passage à distance  $r_0$  de l'intersection est donné par la ligne 9.

---

2. Cette formule vient du fait que les véhicules ralentissent toujours.

### 3.3. Adaptation des vitesses

Dans cette section, nous voyons comment l'algorithme détermine le profil d'accélération à appliquer à chaque véhicule  $i$  entre sa position courante  $r_i$  et  $r_0$ . Pour un véhicule  $i$  donné, ce profil dépend de sa position  $r_i$ , sa vitesse  $v_i$  et le temps  $t_p$  qu'il lui reste pour arriver à distance  $r_0$  de l'intersection.

Pour arriver à l'entrée du cercle de rayon  $r_0$ , à l'instant  $t_p$  (en supposant que l'instant courant est  $t = 0$ ) et à la vitesse de croisière, nous choisissons un profil de vitesse<sup>3</sup> comme illustré par la figure 5 pour un véhicule  $i$  détecté entre  $R$  et  $r_0$ . Quand un véhicule franchit le rayon  $R$ , l'agent de contrôle le perçoit à  $r_i$  avec une vitesse  $v_i$ . En fonction de ses capacités de décélération et d'accélération ( $a_1$  et  $a_2$ ), il doit adapter sa vitesse pour atteindre  $v_x$  puis passer par  $r_0$  à vitesse de croisière. Le profil se décompose ainsi en trois types de zones : (i) deux zones de décélération et d'accélération de durées  $\delta t_{a_1}$  et  $\delta t_{a_2}$ , (ii) un « plateau » de distance  $d$  à vitesse constante, et (iii) deux durées additionnelles fixes de sécurité,  $\delta t_1$  et  $\delta t_2$ , fonctions de la distance ( $l + \frac{\lambda}{2}$ ), qui permettent de stabiliser les vitesses à l'avance.

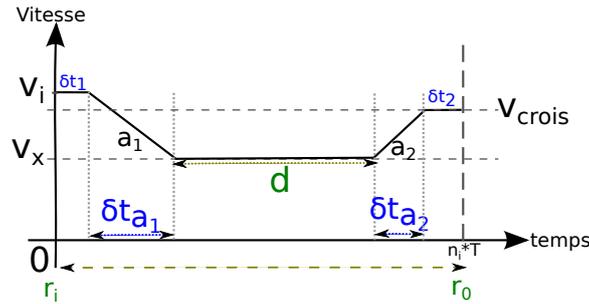


Figure 5. Profil de vitesse

Nous allons maintenant calculer  $v_x$  pour obtenir  $\delta t_{a_1}$  et  $\delta t_{a_2}$ . D'après la figure 5, le temps de parcours de  $r_i$  à  $r_0$  est donné par :

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

$$\text{où } \delta t_{a_1} = \frac{v_x - v_i}{a_1}, \quad \delta t_{a_2} = \frac{v_{crois} - v_x}{a_2}, \quad \delta t_1 = \frac{l + \frac{\lambda}{2}}{v_{crois}}, \quad \text{et } \delta t_2 = \frac{l + \frac{\lambda}{2}}{v_i},$$

et la distance parcourue de  $r_i$  à  $r_0$  est :

$$r_p = 2l + \lambda + \int_0^{\delta t_{a_1}} (v_i + a_1 t) dt + \int_0^{\delta t_{a_2}} (v_{crois} - a_2 t) dt + d. \quad (3)$$

3. Nous raisonnons sur un profil de vitesse, le véhicule recevant le profil d'accélération correspondant.

En utilisant les deux formules 2 et 3 et le paramètre  $d$ , on obtient une équation du second degré :

$$v_x^2 \left( \frac{a_1 - a_2}{2a_1 a_2} \right) + v_x \left( t_p - \delta t_1 - \delta t_2 + \frac{a_2 v_i - a_1 v_{crois}}{a_1 a_2} \right) - \left( r_p - 2l - \lambda + \frac{a_2 v_i^2 - a_1 v_{crois}^2}{2a_1 a_2} \right) = 0. \quad (4)$$

Si le discriminant  $\Delta$  est positif, l'équation admet deux solutions :  $v_x = \frac{-b \pm \sqrt{\Delta}}{2a}$ . L'agent en déduit le profil d'accélération et l'envoie au véhicule. Par la suite, le véhicule appliquera la vitesse de croisière  $v_{crois}$  à l'intérieur de l'intersection (après  $r_0$ ).

Si  $\Delta < 0$ , l'équation n'admet pas de solution réelle, ce qui peut être dû à une distance entre  $r_0$  et  $r_i$  insuffisante pour satisfaire cette adaptation de vitesse. Pour cette raison, la section suivante va étudier les tailles minimales de nos deux rayons.

#### Rayons de la zone de contrôle

Les rayons  $r_0$  et  $R$  (voir figure 4) de chaque intersection ne doivent pas être choisis arbitrairement. Ils doivent d'abord vérifier  $0 < r_0 < R$ , et  $R$  ne doit pas atteindre les intersections voisines. Ils dépendent aussi fortement de l'angle  $\theta$  résultant du croisement des routes de chaque intersection.

– Le rayon minimum  $r_0$  doit être tel que, lorsque le *centre* du véhicule  $i$  est à la distance  $r_0$  du centre de l'intersection, alors le nez de  $i$  n'a pas encore pénétré dans la zone de passage. Cette condition implique :

$$r_0 \geq \frac{L}{2} + \frac{l \max(\cos \frac{\theta}{2}, \cos \frac{\pi - \theta}{2})}{\sin \theta}. \quad (5)$$

– Le rayon  $R$  doit être tel que l'équation (4) peut toujours être résolue ( $\Delta > 0$ ).

Pour un véhicule, le cas limite est quand il devrait attendre presque une période entière avant de passer, c'est à dire que la durée du profil d'accélération est  $t_p - t = \frac{R - r_0}{v} + T + \epsilon$  au lieu de  $\frac{R - r_0}{v} + \epsilon$  (en remplaçant  $r_i$  par  $R$ ).

Ce problème devient plus complexe quand plusieurs véhicules se suivent et forment des files d'attente. En effet, il n'y a pas de solution idéale pour  $R$  car il peut toujours y avoir des véhicules qui arrivent en grand nombre et saturent le segment entre  $R$  et  $r_0$ . Dans la pratique, nous veillons à ce que le rayon  $R$  (qui sera le même pour toutes les intersections) soit assez grand pour qu'il existe ici toujours un profil d'accélération applicable, tout en gardant ce rayon raisonnable.

#### 3.4. Stratégie premier arrivé premier servi

A des fins de comparaisons, nous proposons une seconde approche, laquelle servira de référence, et consiste à faire passer les véhicules dans leur ordre d'arrivée,

d'où l'appellation FCFS (*First Come First Served*). Cette stratégie opportuniste s'inspire de la stratégie de réservation de Dresner et Stone (2008), mais (1) appliquée sur des intersections plus simples et (2) dont on considérera l'effet non seulement localement, mais aussi entre intersections voisines. Par rapport à Alt, on ne réserve plus de créneaux de passage d'une demi-période pour chaque route du réseau mais, dès qu'un véhicule apparaît, on lui affecte la première demi-période disponible.

La figure 6 illustre le principe de l'approche FCFS avec un croisement de deux routes *A* et *B* à deux voies (*A1*, *A2*, *B1* et *B2*). A la première demi-période, la première voie de la route *A* fait passer un véhicule. Ensuite, à la deuxième demi-période (c'est à dire à  $T$ ), les deux voies de la route *A* font passer un véhicule chacune. Puis, à la troisième demi-période, c'est au tour de la route *B*, et ainsi de suite.

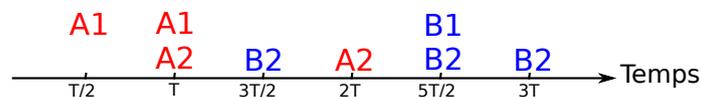


Figure 6. Illustration du fonctionnement de l'approche First Come First Served

### 3.5. Résultats expérimentaux

Afin d'évaluer les performances de nos algorithmes, nous avons décidé de développer notre propre outil de simulation. A notre connaissance il n'existe pas de simulateur microscopique adapté à nos besoins. Les plus fins, tels que ARCHISIM (Espié, 1995), MOVSIM (Treiber, Kesting, 2010), VIVUS (Gechter *et al.*, 2012), TrafficGen (Bonhomme *et al.*, 2014) ou AIM (Dresner, Stone, 2008), sont coûteux en temps de calcul, difficiles à modifier (quand le code source est disponible), et ne satisfont pas les mêmes hypothèses géométriques et cinématiques (angles des routes, propriétés des véhicules).

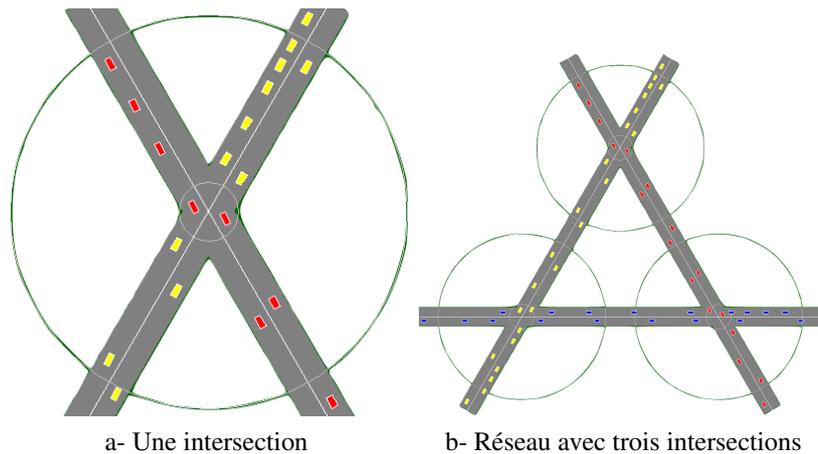


Figure 7. Illustration du simulateur

Le simulateur développé (Tlig, 2015, chap. 5) est à espace continu et à temps discret (captures d'écran fig. 7-a et b). Les paramètres de simulation utilisés, ici avec le réseau de la fig. 7-b, sont :

- chaque route (respectivement chaque segment) est de longueur 1 000 m (resp. 500 m) et de largeur 5 m,  $\lambda = 5$  m, et l'angle de croisement est  $\frac{\pi}{3}$  ;
- le rayon d'action  $R$  de l'agent de contrôle est fixé à 200 m, tandis que  $r_0 = 30$  m (déterminé à partir de l'équation 5) ;
- les véhicules ont les dimensions  $L = 5$  m,  $l = 2,5$  m, ainsi qu'une marge de sécurité d'inter-distance  $\epsilon = 1,5$  m ; les valeurs maximales de vitesse, accélération et décélération sont respectivement 10 m/s, 1,5 m/s<sup>2</sup>, et -1,5 m/s<sup>2</sup> ;
- le pas de simulation est de 0,1 s.

Pour cette configuration, la période minimale est  $T = 3,2$  s, et chaque source injecte des véhicules suivant un processus de Bernoulli de paramètre  $\frac{1}{D}$  (où  $D$  est la durée moyenne, en secondes, entre deux injections consécutives). Une vidéo du simulateur est accessible à l'adresse suivante : [http://www.loria.fr/~7emtlig/videos/5\\_intersections.avi](http://www.loria.fr/~7emtlig/videos/5_intersections.avi).

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

### 3.5.1. Comparaison des stratégies

L'objectif est de comparer les approches Alt et FCFS avec les feux de circulation en observant les retards en résultant. Le retard est ici la différence entre un temps de passage théorique (déterminé à partir de la distance à parcourir pour traverser une intersection) et le temps de passage réel. Pour chaque approche, nous avons calculé la moyenne et l'écart-type des retards pour 500 véhicules qui traversent le réseau. La fréquence d'injection utilisée en chaque source était élevée ( $D = 5s$ ), ce qui donne un trafic dense.

La figure 8 présente les résultats de simulation de Alt et FCFS, et de l'approche par feux de circulation avec différentes durées de phases (10s, 20s, 30s). Les axes indiquent en abscisse le nombre de véhicules ayant quitté le réseau dans l'ordre de sortie et en ordonnée le retard moyen (plus l'écart-type) des véhicules en secondes. Les courbes tracées sont des moyennes sur 100 simulations. Nous arrêtons les simulations dès que 500 véhicules sont sortis, mais sans avoir arrêté les injections.

L'observation des courbes montre que Alt est clairement plus performante sur cette expérimentation que FCFS ou les feux de circulation (en particulier pour les durées de feux de 20 s et 30 s). L'approche par feux de circulation devrait se dégrader pour de trop grandes durées de phases. Toutefois, selon les durées, un plus ou moins grand nombre de véhicules a l'opportunité de ne pas s'arrêter entre deux feux consécutifs,

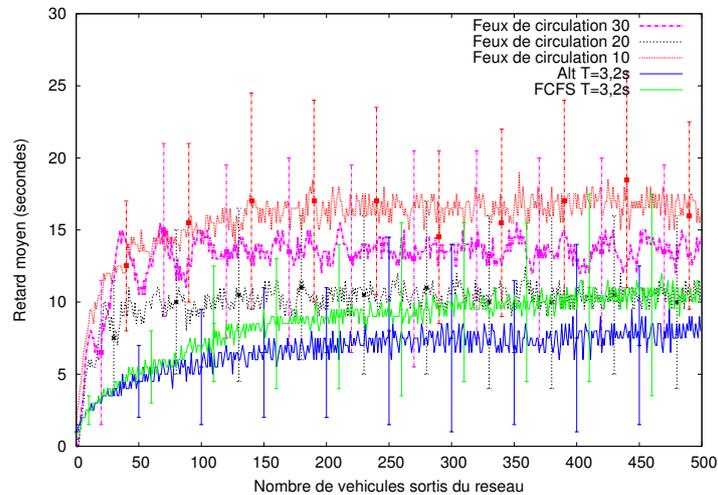


Figure 8. Alt vs. FCFS vs. Feux de circulation à une intersection

d'où ici le fait que les pires retards ne correspondent pas à la durée de 30 s.<sup>4</sup> La moyenne générale des retards est de l'ordre de 8 s pour Alt alors que, pour FCFS et pour la meilleure courbe des feux de circulation, elle est de l'ordre de 11 s, et va jusqu'à 16,5 s pour la pire courbe. Sur 100 simulations, le total moyenne+écart-type (approximation du pire retard) pour l'approche par Alt est de l'ordre de 13 s alors que, pour les autres approches, il est entre 16 s et 26 s. Nous avons aussi testé l'approche par feux de circulation avec des durées de phases inférieures à 10 s et avons observé que les retards augmentent significativement. C'est dû principalement aux véhicules qui s'arrêtent et requièrent ainsi plus de temps pour traverser les intersections, ce qui conduit même à la formation de files d'attente et à des collisions quand la durée des phases est égale ou inférieure à 5 s.

### 3.5.2. Effet de variation du paramètre $R$ (rayon d'action)

Nous mesurons (pour Alt, mais les résultats sont comparables pour FCFS) l'effet de la variation du rayon d'action  $R$  de l'agent de contrôle afin d'évaluer s'il influe sur la qualité de notre solution (sans modifier  $r_0$ ). Le tableau 2 donne les moyennes et les écarts-types de la consommation d'énergie cinétique seulement à l'accélération lorsque  $R = 50$  m,  $R = 100$  m et  $R = 200$  m. Elle est mesurée pendant la traversée de 1 000 véhicules. Nous observons que les retards ne changent pas de manière significative, mais il y a une différence en termes de consommation d'énergie lorsque nous réduisons le rayon. Elle augmente quand  $R$  est réduit. En effet, la faible dis-

4. On verra dans la section suivante que la synchronisation des commandes périodiques est un moyen d'optimiser le réseau.

tance oblige l'agent de contrôle à ralentir brutalement les véhicules (pour corriger leur avance) et donc à générer ensuite une accélération consommatrice d'énergie.

Tableau 2. Moyennes et écarts types de l'énergie consommée en fonction du rayon d'action  $R = 50$  m,  $R = 100$  m et  $R = 200$  m

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

#### 4. Coordination et optimisation distribuée d'un réseau d'intersections

Nous passons maintenant à l'échelle du réseau, avec pour but de rendre le trafic plus régulier et performant. Nous posons le problème (sec. 4.1), discutons les critères (sec. 4.2), puis présentons les algorithmes d'optimisation proposés (sec. 4.3).

##### 4.1. Problème posé

Notre modèle du réseau est le suivant :

- Le réseau est constitué de routes – chacune avec deux voies opposées – et de leurs intersections. Une intersection  $i$  d'angle  $\theta_i$  permet à un véhicule de croiser une route, mais pas de tourner.
- Chaque intersection  $i$  est liée à 4 intersections voisines.
- Chaque voie (*lane*)  $l$  a son propre débit  $W_l$ .
- Le réseau peut donc être vu comme un graphe  $(\mathcal{V}, \mathcal{E})$  dont les sommets dans  $\mathcal{V}$  sont les intersections et les arêtes dans  $\mathcal{E}$  sont les segments de voie  $i \rightarrow j$ .

On notera (1) que le débit d'une voie est limité par le plus faible débit des intersections qu'elle parcourt, et (2) que, si deux intersections successives ont des périodes différentes, alors les durées de parcours des segments sont variables, donc incontrôlables. On choisit donc d'imposer à toutes les intersections la plus petite des périodes compatibles avec chaque intersection. En notant  $T_{min}^j$  la plus petite période possible pour une intersection  $j$  (sec. 3.1), la période d'une l'intersection  $i$  est ainsi  $T^i = \max_j T_{min}^j$ . En pratique, les capacités du réseau sont peu influencées par ce choix puisque les angles, donc les périodes minimales, sont souvent proches.

Compte tenu de notre approche par alternance avec période commune, la seule variable de contrôle est la phase  $\phi_i \in [0, T[$  du signal périodique de chaque intersection. Dans l'idéal, la phase  $\phi_i$  d'une intersection  $i$  sera réglée de telle sorte qu'un véhicule venant d'une intersection voisine  $j$  (ou allant vers celle-ci) n'ait pas à ralentir ou accélérer entre les deux intersections pour les traverser sans s'arrêter. Sauf cas particulier, une telle situation ne peut être atteinte. Notre objectif est de régler au mieux les phases des différentes intersections pour tendre vers cette situation idéale.

#### 4.2. Critères d'optimisation séparables

Nous considérons ici deux critères choisis pour leur importance, les critères temps et énergie, lesquels visent, en notant  $\vec{\phi} = (\phi_1, \phi_2, \dots, \phi_n)$  le vecteur des phases des intersections :

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

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

2. à minimiser l'énergie totale consommée, c'est-à-dire la somme de l'énergie cinétique  $E_{i,j}$  sur chaque segment orienté  $i \rightarrow j$ , pondérée par les débits  $W_{i,j}$  :

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

Les véhicules traversant les intersections à la vitesse par défaut, le temps de traversée  $t_{i,j}$  et la consommation d'énergie  $E_{i,j}$  ne dépendent que de  $\phi_i$  et  $\phi_j$ . En outre, ces deux critères sont additivement séparables (au contraire, par exemple, du pire temps de traversée du réseau), c'est-à-dire qu'ils peuvent s'écrire :

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

où  $f_{i,j}$  représente soit l'énergie cinétique, soit le temps de traversée. Un premier avantage de ces critères séparables est qu'ils peuvent être calculés de manière efficace en parallèle. Un deuxième avantage est qu'à chaque intersection  $i$  sont naturellement associés une variable de contrôle  $\phi_i$  et un critère local à optimiser :

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

où  $\mathcal{E}_i$  est l'ensemble des voisins de  $i$ . En conséquence, nous allons naturellement obtenir des algorithmes d'optimisation décentralisés. Cette séparabilité n'empêche toutefois pas de tomber dans un optimum local.

Pour toute paire d'intersections voisines  $i$  et  $j$ , les valeurs  $f_{i,j}$  se calculent facilement à l'aide des phases  $\phi_i$  et  $\phi_j$ . Il suffit de déterminer le profil d'accélération d'un véhicule allant de  $i$  à  $j$ , et d'en déduire le temps de parcours de  $i$  à  $j$  ou l'énergie cinétique consommée. Par la suite, on appellera  $\text{Évaluation}(\phi_i^t, \vec{\phi}_{voisins(i)}^t)$  la fonction calculant la mesure de performance associée à l'intersection  $i$  et à l'instant  $t$ .

#### 4.3. Optimisation des phases des intersections

Une approche simple d'optimisation des phases est présentée, puis améliorée.

#### 4.3.1. Approche Hill-Climbing

Pour optimiser le réseau, nous partons d'un des algorithmes de recherche locale les plus simples, le Hill-Climbing (Russell, Norvig, 1995), et donc soulevant le moins de difficultés pour une adaptation à notre cadre de travail. D'autres algorithmes de recherche locale demanderaient soit de pouvoir calculer le gradient de la fonction d'évaluation (pour la descente de gradient), soit d'adapter des mécanismes particuliers au cas d'un algorithme décentralisé (la gestion d'une liste tabou par exemple).

Le Hill-Climbing est une technique d'optimisation mathématique itérative commençant avec une solution arbitraire à un problème, puis tentant à l'améliorer en changeant progressivement un seul élément de cette solution à la fois. A chaque itération  $t$ , en chaque intersection  $i$ , l'algorithme cherche localement la meilleure phase tout en tenant compte des phases (connues et supposées fixes) des intersections voisines  $j \in \mathcal{E}_i$ , puis assigne cette valeur à l'intersection  $i$  :  $\phi_{t+1}(i)$ . Les détails sont présentés dans l'algorithme 2, où  $\Delta\phi_t$  est l'intervalle de recherche qui devrait diminuer lentement pour assurer la convergence, et  $\delta\phi_t/\Delta\phi_t$  un entier constant. Ce processus est répété pendant 100 itérations. On notera que, pour une recherche locale, on choisira *a priori* une valeur  $\Delta\phi$  petite devant la période  $T$ .

---

#### Algorithme 2. Hill-Climbing-iteration( $\phi_i^t, \vec{\phi}_{voisins(i)}^t, \Delta\phi$ )

---

```

1 ( $V^{min}, \phi^{min}$ )  $\leftarrow$  (ÉVALUATION( $\phi_i^t, \vec{\phi}_{voisins(i)}^t, \phi_i^t$ )
2 pour  $\phi_{tmp} \leftarrow \phi_i^t - \Delta\phi$  à  $\phi_i^t + \Delta\phi$  pas  $\delta\phi$  faire
3   [ $(V^{min}, \phi^{min}) \leftarrow$  TEST( $\phi_{tmp}, \vec{\phi}_{voisins(i)}^t, V^{min}, \phi^{min}$ )
4    $\phi_i^{t+1} \leftarrow \phi^{min}$ 
5 retourner  $\phi_i^{t+1}$ 

```

---

```

6 TEST( $\phi_{tmp}, \vec{\phi}_{voisins}, V^{min}, \phi^{min}$ )
7 début
8   [ $V^{tmp} \leftarrow$  ÉVALUATION( $\phi_{tmp}, \vec{\phi}_{voisins}$ )
9   si  $V^{tmp} < V^{min}$  alors ( $V^{min}, \phi^{min}$ )  $\leftarrow$  ( $V^{tmp}, \phi_{tmp}$ )
10  retourner ( $V^{min}, \phi^{min}$ )

```

---

#### 4.3.2. Paysages des fonctions à optimiser

En vue d'améliorer l'algorithme utilisé, nous étudions les critères à optimiser, à savoir le temps de traversée et l'énergie consommée par les véhicules. Pour une intersection  $i$  du réseau, nous fixons les phases de ses voisines et faisons varier sa phase dans l'intervalle  $[0, 3T]$ . Les figures 9 *a* et *b* illustrent l'évolution du critère énergie en fonction de la phase pour deux configurations différentes (géométrie et phases voisines), et les figures 9 *c* et *d* illustrent l'évolution du critère temps en fonction de la phase pour les mêmes configurations.

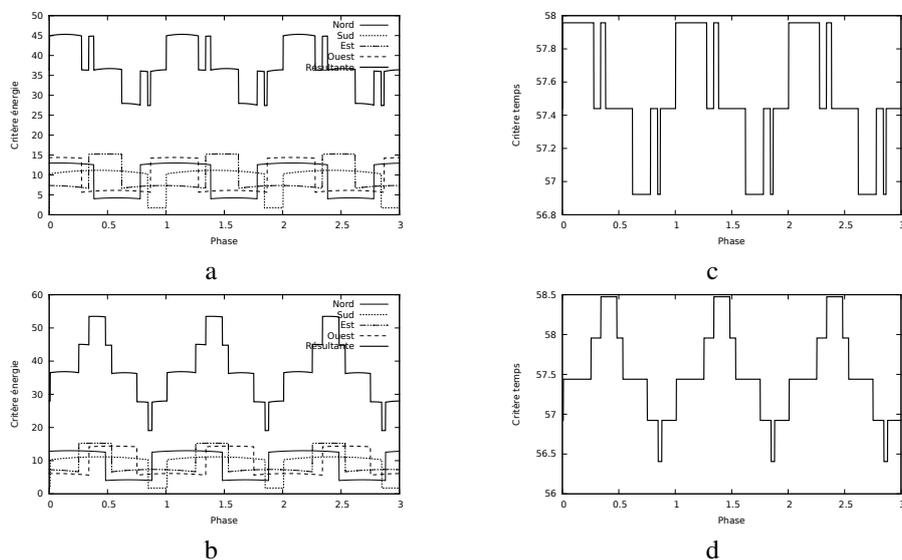


Figure 9. Comparaison entre les courbes d'optimisation obtenues pour le critère énergie (a, b) et pour le critère temps (c, d) dans les mêmes configurations

On peut voir d'abord, que le critère temps forme une fonction en escalier, ce qui est attendu parce que le temps de parcours d'un segment  $i \rightarrow j$  est une fonction linéaire des phases, et évolue de manière opposée au segment  $j \rightarrow i$ . Par ailleurs, dans notre cas, les critères temps et énergie ont la même forme générale, quasiment identique à une transformation affine près (et un arrondi). Ceci est dû au fait que les véhicules ne peuvent que décélérer, donc consomment du temps comme de l'énergie dans chaque parcours de segment. Si, par contre, la vitesse de croisière était inférieure à la vitesse maximale, alors les véhicules auraient le choix entre ralentir et accélérer, et nous pourrions donc avoir des formes de courbe différentes.

Dans les deux cas, les fonctions obtenues contiennent des régions plutôt plates (des plateaux dans le cas du temps) et plusieurs optima locaux, ce qui rend l'optimisation difficile en chaque intersection. Ces paysages locaux, plus la forte dépendance entre les intersections voisines, sont susceptibles de rendre l'optimisation au niveau du réseau très difficile.

#### 4.3.3. Algorithme stochastique distribué

Comme déjà mentionné, la phase d'une intersection influence fortement les phases (optimales) des intersections voisines. L'approche de Hill-Climbing distribué décrite précédemment – où les intersections « font de leur mieux » sans tenir compte des évolutions de leurs voisines – peut conduire à des comportements oscillants appelés « deadlocks attractors » dans le cas des CSP distribués (Fabiunke, 1999).

Pour casser cette dépendance et désynchroniser les intersections, une solution est d'employer des algorithmes distribués stochastiques (DSA pour *distributed stochastic algorithms*) (Fitzpatrick, Meertens, 2001 ; Weixiong *et al.*, 2005). Ces algorithmes proposent d'introduire un processus stochastique dans la phase d'optimisation. A chaque pas de temps, chaque intersection  $i$  choisit de garder sa phase courante ( $\phi_i$ ) avec une probabilité  $1 - \rho$ , ou de l'optimiser avec probabilité  $\rho$ . Ce processus stochastique limite les variations perpétuelles des phases, et donc les dépendances.

#### 4.3.4. Intervalle de recherche local révisé

Etant donné le paysage du critère périodique à chaque intersection  $i$ , chercher dans un intervalle large – jusqu'à  $\Delta\phi = \frac{T}{2}$  – semblerait meilleur pour éviter plateaux et optima locaux que de faire une recherche restreinte autour de  $\phi_i^t$ . Cet élargissement est la seconde proposition d'amélioration (après l'emploi de DSA) et, parmi les trois propositions qui seront faites, elle est la seule à n'impliquer qu'un paramètre.

Aussi, alors que le Hill-Climbing proposé en section 4.3.1 (que l'on dénotera HC par commodité) utilise des points distribués uniformément dans ses intervalles de recherche, une troisième proposition d'amélioration consiste en une distribution non-uniforme en des points  $\phi_i^t \pm \frac{\Delta\phi}{k}$ , où  $k$  est un entier dans  $[1, k_{max}]$  comme illustré figure 10.  $k_{max}$  est un paramètre secondaire permettant, comme  $\delta\phi$  pour une distribution uniforme, de contrôler la discrétisation de l'intervalle de recherche.

Pour un certain jeu de paramètres  $\rho \in ]0, 1]$ ,  $\Delta\phi \in ]0, \frac{T}{2}]$  et  $dist \in \{unif, non-unif\}$ , on notera l'algorithme  $DSA(\rho, \Delta\phi, dist)$ .<sup>5</sup> Ainsi, l'approche précédente, HC, pour une valeur de  $\Delta\phi$  donnée, correspond au cas  $DSA(1, \Delta\phi, unif)$ . L'algorithme 3 présente une itération de  $DSA$  dans le cas particulier où  $dist = non-unif$ , pour une intersection  $i$ , donc en précisant en plus  $\phi_i^t$  et  $\bar{\phi}_{voisins(i)}^t$ .

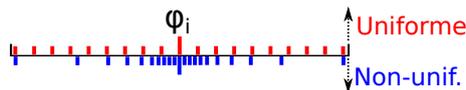


Figure 10. Intervalles de recherche uniforme (utilisé par HC) et non-uniforme

#### 4.3.5. De l'optimisation hors ligne à l'optimisation en ligne

L'algorithme qui vient d'être proposé peut non seulement être exécuté de manière distribuée avec un agent localisé à chaque intersection, mais aussi fonctionner « en ligne », moyennant quelques modifications, pour s'adapter aux conditions changeantes du trafic. Nous détaillons ci-après les deux principales modifications nécessaires, lesquelles visent (i) à obtenir une estimation temps-réel des conditions du trafic, et (ii) à permettre l'exécution et l'optimisation simultanés.

5. Pour des raisons de lisibilité, cette notation n'indique pas les paramètres secondaires  $\delta\phi$  et  $k_{max}$ . Mais la méthode pour choisir ceux-ci sera décrite avec le cadre expérimental.

**Algorithme 3.** DSA-iteration( $\phi_i^t, \vec{\phi}_{voisins(i)}^t, \rho, \Delta\phi$ )

---

```

1  $\beta \leftarrow$  valeur échantillonnée uniformément dans  $[0, 1)$ 
2  $(V^{min}, \phi^{min}) \leftarrow$  (ÉVALUATION( $\phi_i^t, \vec{\phi}_{voisins(i)}^t$ ),  $\phi_i^t$ )
3 si ( $\beta \leq \rho$ ) alors
4   pour  $k \leftarrow 1$  à  $k_{max}$  faire
5      $(V^{min}, \phi^{min}) \leftarrow$  TEST( $\phi_i^t - \frac{\Delta\phi}{k}, \vec{\phi}_{voisins(i)}^t, V^{min}, \phi^{min}$ )
6      $(V^{min}, \phi^{min}) \leftarrow$  TEST( $\phi_i^t + \frac{\Delta\phi}{k}, \vec{\phi}_{voisins(i)}^t, V^{min}, \phi^{min}$ )
7    $\phi_i^{t+1} \leftarrow \phi^{min}$ 
8 sinon  $\phi_i^{t+1} \leftarrow \phi_i^t$ 
9 retourner  $\phi_i^{t+1}$ 

```

---

## 4.3.5.1. Estimation des conditions du trafic

En ligne, une première difficulté est que les conditions du trafic ne sont pas connues, mais doivent être estimées. Ici, chaque agent de contrôle  $i$  estime simplement le débit sur chacune de ses quatre voies. Pour la voie  $X$ , l'agent  $i$  compte les véhicules traversant son intersection et estime ensuite le débit sur cette voie par une moyenne glissante exponentielle de paramètre  $\alpha > 0$  (lequel permet de s'adapter à des conditions plus ou moins changeantes).

## 4.3.5.2. Exécution en cours d'optimisation

Une seconde difficulté de l'optimisation en-ligne est que l'approche Alt n'est pas faite pour fonctionner avec des phases évoluant constamment. Pour résoudre cela, deux mécanismes sont mis en œuvre :

1. une fois calculé, il n'est pas permis de changer le temps de passage d'un véhicule à sa prochaine intersection ; et
2. on n'appliquera pas directement les phases  $\phi_i^t$  obtenues par DSA (lesquelles peuvent être vues comme des objectifs à atteindre), mais des phases *appliquées*  $\tilde{\phi}_i^t$  obtenues en suivant lentement et continûment les phases objectif  $\phi_i^t$  sur  $[O; T[$ .

On notera que, pour pouvoir modifier, même légèrement, une phase appliquée  $\tilde{\phi}_i^t$ , il faut une petite marge de sécurité pour éviter des collisions (ce qui nécessite une augmentation de la période du réseau). Il y a donc là un compromis à régler entre le débit maximum du réseau et la vitesse d'adaptation à des conditions changeantes.

## 5. Résultats expérimentaux

Cette section évalue expérimentalement les bénéfices de l'optimisation des phases pour la stratégie Alt, et compare cette stratégie à la plus opportuniste FCFS.

Toutes les expérimentations sont conduites sur un réseau composé de 6 routes à double sens et 12 intersections (photo d'écran en fig. 11). Deux groupes de 6 voies sont définis (une voie par route), et des conditions de trafic différentes sont créées en affectant simplement un débit faible, moyen ou fort ( $5/10/15$  véhicules.minute<sup>-1</sup>) à toutes les voies d'un même groupe. Ces conditions de trafic sont notées  $X-Y$ , où  $X$  et  $Y \in \{5, 10, 15\}$  sont les débits de chacun des deux groupes.

Notons que nous mesurons le critère optimisé seulement à l'intérieur du réseau (une fois que les véhicules sont passés à travers une première intersection). Cela permet de réduire notablement la variance de nos estimations – sans dégrader la mesure de performance – puisque le système ne contrôle pas ce qui se passe à l'extérieur.

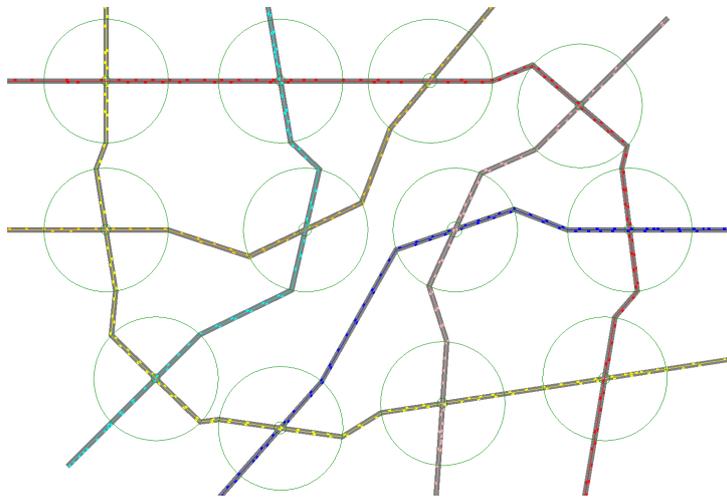


Figure 11. Réseau à 12 intersections

### 5.1. Cadre de simulation

Nous employons le même simulateur qu'en section 3.5, mais avec un réseau à 6 routes (paramètres choisis de manière à proposer un scénario relativement réaliste) dont les paramètres suivants sont modifiés :

- le rayon d'action  $R$  de l'agent de contrôle est fixé à 125 m ;
- la distance entre deux intersections adjacentes est de  $\sim 450$  m ;
- hors des zones de contrôlé un contrôle longitudinal de proche en proche (Scheuer *et al.*, 2009) évite les collisions entre véhicules d'une même voie ;
- le pas de temps de simulation est fixé à 0,05 s.

Pour ce réseau (voir fig. 11 et vidéo disponible sur : [http://www.loria.fr/~emtlig/videos/12\\_intersections.avi](http://www.loria.fr/~emtlig/videos/12_intersections.avi)), la période commune (la pire) est  $T_{max} = 3,5$  s. Pour les optimisations, on prend des distributions de 16 points en utilisant  $\delta\phi = \frac{\Delta\Phi}{8}$  et  $k_{max} = 8$  (cf fig.10).

## 5.2. Résultats

### 5.2.1. Analyse des paramètres

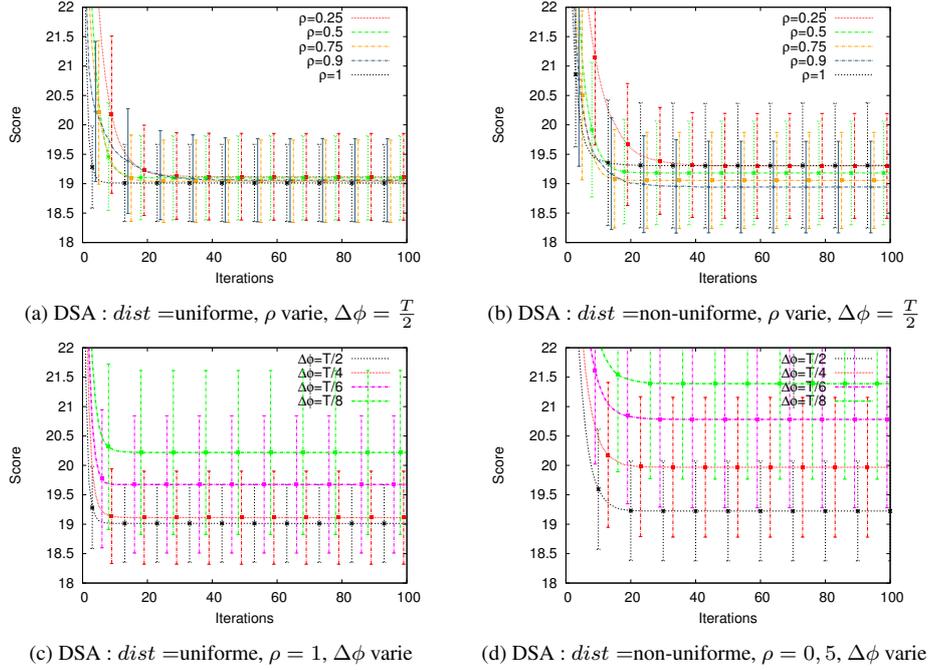


Figure 12. Convergence de l'algorithme d'optimisation du réseau pour différents jeux de paramètres

Dans cette section nous étudions l'impact de chacune des trois améliorations candidates : (i) l'intervalle de recherche  $\Delta\phi$ , (ii) la probabilité de mise à jour de DSA  $\rho$ , et (iii) la distribution uniforme ou non-uniforme dans l'intervalle de recherche. Nous utilisons le réseau de la figure 11, fixons les débits équitablement sur toutes les sources (10–10), et considérons le critère énergie dans le cas hors-ligne.

La figure 12 a–d montre quatre courbes qui correspondent à DSA avec :

- a :  $dist = \text{uniforme}$ ,  $\rho \in \{0,25, 0,5, 0,75, 0,9, 1\}$ ,  $\Delta\phi = \frac{T}{2}$ ,
- b :  $dist = \text{non-unif.}$ ,  $\rho \in \{0,25, 0,5, 0,75, 0,9, 1\}$ ,  $\Delta\phi = \frac{T}{2}$ ,
- c :  $dist = \text{uniforme}$ ,  $\rho = 1$ ,  $\Delta\phi \in \{\frac{T}{2}, \frac{T}{4}, \frac{T}{6}, \frac{T}{8}\}$ , et
- d :  $dist = \text{non-unif.}$ ,  $\rho = 0,5$ ,  $\Delta\phi \in \{\frac{T}{2}, \frac{T}{4}, \frac{T}{6}, \frac{T}{8}\}$ .

Chaque courbe montre les moyennes et écarts-types de la consommation d'énergie (pour le réseau complet) sur 1 000 simulations des 100 premières itérations.

Comme le montre la figure 12-a, avec la distribution uniforme, la vitesse de convergence pendant les premières itérations dépend de  $\rho$  (plus l'algorithme est déterministe,

plus il est rapide), mais la valeur limite (entre 19 et 19, 2), elle, n'en dépend pas. Avec la distribution non-uniforme (fig. 12-b), la vitesse de convergence est plus faible pendant les premières itérations et ne croît plus avec  $\rho$  de manière aussi évidente. Aussi, les valeurs limites sont plus étalées, les meilleurs résultats étant obtenus avec  $\rho = 0, 9$  (18, 9). Ici, un faible aléa améliore à la fois la vitesse de convergence et la valeur limite ; par contre, un aléa plus important dégrade les performances. Au vu des deux courbes, la distribution uniforme apparaît ici comme plus robuste au choix de  $\rho$ .

Les figures 12-c et d montrent qu'accroître  $\Delta\phi$  significativement améliore la convergence dans deux cas distincts (deux distributions avec des comportements stochastiques différents). L'écart-type décroît aussi, mais en restant non-négligeable (comme dans les figures a et b), ce qui suggère que l'algorithme converge souvent vers un optimum local.

Notons que nous ne représentons pas les performances de HC (approche de la section 4.3.1), lesquelles sortent des graphes avec un score autour de 23. Par ailleurs, toutes ces mesures dépendent de la configuration du réseau et des débits d'injection (10-10 dans le cas présent), de sorte qu'employer ces mêmes paramètres dans d'autres configurations ne garantit pas les mêmes résultats.

### 5.2.2. Comparaison de Alt et FCFS hors ligne

L'utilisation de DSA avec  $\rho = 0, 9$ ,  $\Delta\phi = \frac{T}{2}$  et une distribution non-uniforme ayant fourni de bons résultats dans les expérimentations précédentes, ce paramétrage sera désormais appliqué ; l'algorithme résultant sera simplement noté « DSA », et opposé à HC.

En considérant toujours le cas hors-ligne, nous comparons ici Alt et FCFS soit (cas asynchrone) avec des périodes locales  $T_{min}(i)$  et des phases aléatoires, soit (cas synchrone) avec la période commune  $T$  et des phases aléatoires, soit (cas « optimisé » HC/DSA) avec la période commune  $T$  et des phases optimisées pour Alt. Chacune des 4 stratégies résultantes est utilisée dans 4 conditions de trafic pendant 10 expérimentations durant 1 heure chacune (temps dans la simulation). Les résultats sont présentés dans le tableau 3, où nous mesurons l'énergie moyenne consommée par véhicule (en ne tenant compte que de l'énergie cinétique dépensée pour accélérer).

Observons d'abord que la synchronisation du réseau (avec la période commune  $T$ ) améliore déjà Alt, sauf en cas de faible densité (5-5). Par contre, elle dégrade toujours la qualité du trafic pour FCFS (qui profite de son opportunisme dans le cas asynchrone). Dans les cas des intersections synchrones, les phases optimisées pour Alt lui permettent de réduire la consommation d'énergie, et ce d'autant mieux que l'on emploie DSA (lequel évite plus d'optima locaux) plutôt que HC. Elles n'ont toutefois pas d'effet sur FCFS.

Par contre, FCFS est opportuniste puisqu'il économise de l'énergie en cas de faible densité de trafic (5-5). Toutefois, plus le trafic est chargé, plus des conflits ont lieu aux intersections, et donc moins d'opportunités peuvent être exploitées par FCFS, ce qui signifie que les véhicules doivent souvent ralentir. Dans les mêmes conditions

Tableau 3. *Energie moyenne consommée par véhicule en utilisant Alt et FCFS avec soit des intersections asynchrones et des phases aléatoires, soit des intersections synchrones et des phases optimisées hors-ligne pour Alt ; les injections sont mesurées en nombre de véhicules par minute*

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

de trafic chargé, Alt optimisé est plus avantageux ; il maintient une consommation énergétique plus faible. Notons que les mesures avec le critère temps amènent aux mêmes conclusions (voir (Tlig, 2015)).

### 5.2.3. Comparaison de Alt et FCFS en ligne

Nous regardons maintenant le cas en-ligne. Pour cela, nous considérons une période de 4 heures durant laquelle les quatre conditions de trafic utilisées précédemment sont appliquées 1 heure chacune, et comparons :

- FCFS avec des périodes locales et des phases aléatoires,
- Alt avec des phases optimisées hors ligne pour les débits moyens sur les 4 heures : 10-10, et
- Alt avec des phases optimisées en ligne avec différents taux d'apprentissage  $\alpha$ .

Comme détaillé précédemment, chaque agent intersection  $i$  estime seul les débits sur ses voies, optimise sa propre phase  $\phi_i$ , et la communique uniquement à ses voisins. Le tableau 4 présente des estimations empiriques de la consommation d'énergie et du temps de traversée, pour chaque véhicule, dans chaque cas.

Tableau 4. *Energie moyenne consommée et temps de traversée, pour un véhicule, en utilisant Alt avec des phases optimisées en ligne, Alt avec des phases optimisées hors ligne, et FCFS avec des phases aléatoires*

Alg. ( $\alpha$ ).10 <sup>-3</sup>	(0,390 625)	Alt en ligne (0,781 25)	(1,562 5)	Alt hors ligne	FCFS
HC énergie	81,3 ± 8	<b>78</b> ± 3,8	78,3 ± 7,3	80,1 ± 6,9	101,2
DSA énergie	70,4 ± 7,5	<b>68,7</b> ± 5,4	69,3 ± 4,9	70,4 ± 7,6	101,2
HC temps	138 ± 0,5	<b>137,8</b> ± 0,2	137,9 ± 0,5	138,2 ± 0,4	139,9
DSA temps	137,5 ± 0,5	<b>137,4</b> ± 0,3	137,4 ± 0,2	137,6 ± 0,3	139,9

La version en ligne de la stratégie Alt améliore les résultats de la version hors-ligne et est plus efficace que FCFS. Cela démontre clairement que l'estimation en-ligne de débits et l'optimisation des phases sont efficaces sur ce réseau, *offrant une approche qui s'adapte aux conditions de trafic*. Les meilleurs résultats sont obtenus

quand  $\alpha = 0,78125 \cdot 10^{-3}$ , même si la différence avec d'autres valeurs n'est pas significative.

## 6. Conclusion

Dans cet article, nous avons proposé un mécanisme de croisement sans arrêt pour des flux de véhicules autonomes. Celui-ci repose sur l'établissement d'une période minimale permettant le passage alterné des véhicules provenant des flux se croisant (approche Alt). La mise en œuvre suppose que l'infrastructure des intersections est capable de communiquer avec les véhicules pour la transmission des profils d'accélération. Nous avons mesuré, sur un réseau de trois routes et trois intersections, que l'approche réduit considérablement les retards par rapport à un système de feux de circulation, mais aussi par rapport à FCFS, une approche par réservation inspirée de (Dresner, Stone, 2008).

Dans un deuxième temps, nous avons considéré la possibilité de coordonner les intersections entre elles, afin d'optimiser le trafic global d'un réseau de routes se croisant. Pour cela, nous avons proposé une étude des paysages des critères de performance. Celle-ci nous a conduit à proposer l'emploi d'un algorithme d'optimisation distribué générique (DSA) dans le cadre d'intersections autonomes, et ce, en utilisant différents intervalles de recherche locale.

Des expérimentations ont montré que Alt avec des phases optimisées avec DSA fait significativement mieux que les autres solutions (phases non optimisées, ou optimisées avec HC), excepté pour des conditions de trafic faibles, où l'approche opportuniste FCFS est plus appropriée. Ces résultats restent valables dans le cadre en-ligne, où s'adapter à des conditions de trafic changeantes est meilleur qu'une configuration statique précalculée.

### *Travaux futurs*

L'algorithme d'optimisation proposé pourrait être facilement amélioré, par exemple en utilisant des redémarrages ou en réduisant progressivement les intervalles de recherche. En particulier, plutôt que de sélectionner des points selon une distribution prédéfinie dans  $[\phi_i - \Delta\phi, \phi_i + \Delta\phi]$ , une autre perspective est de calculer analytiquement où se trouvent les « sauts » (les discontinuités) du paysage des critères énergie ou temps. En effet, au moins l'un d'entre eux correspond à une valeur optimale de  $\phi_i$  (étant données les phases voisines fixes).

Par ailleurs, d'autres expérimentations devraient être effectuées pour obtenir une meilleure compréhension des effets de l'approche proposée, en particulier en considérant des réseaux différents et des scénarios différents.

Enfin, une question ouverte importante est de savoir quand utiliser les stratégies Alt et FCFS. Peut-on passer automatiquement de l'une à l'autre, ou éventuellement employer les deux stratégies au sein d'un même réseau routier ?

Nous comptons aussi étendre notre modèle pour permettre aux véhicules de tourner aux intersections. C'est possible en utilisant, à chaque intersection, des cycles constitués de quatre phases au lieu de deux : 2 phases pour permettre aux véhicules de chaque route d'aller tout droit ou de tourner à droite, plus 2 phases pour permettre aux véhicules de chaque route de tourner à gauche. Les voies auront besoin d'être dédoublées avant les intersections pour éviter les conflits entre véhicules de la même voie mais allant dans des directions différentes.

La méthode d'optimisation distribuée proposée pourrait être mise en œuvre sur des réseaux de feux de signalisation en modifiant simplement la fonction à optimiser. Elle pourrait aussi être employée dans d'autres domaines, par exemple les réseaux de capteurs (une des premières applications de l'algorithme stochastique distribué DSA), à condition qu'ils impliquent des critères de performance séparables et périodiques.

## Bibliographie

- Balan G., Luke S. (2006). History-based traffic control. In *Proceedings of the fifth International Conference on Autonomous Agents and Multiagent Systems*, p. 616–621.
- Bazzan A. (2005). A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multi-Agent Systems*, vol. 10, n° 1, p. 131–164.
- Bazzan A. (2009). Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, vol. 18, n° 3, p. 342–375.
- Bhouri N., Balbo F., Pinson S., Tlig M. (2011, aug.). Collaborative agents for modeling traffic regulation systems. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, vol. 2, p. 7–13.
- Bonhomme A., Mathieu P., Picault S. (2014). A versatile description framework for modeling behaviors in traffic simulations. In *International Conference on Tools with Artificial Intelligence, (ICTAI'14)*, p. 937–944.
- Brockfeld E., Barlovic R., Schadschneider A., Schreckenberg M. (2001). Optimizing traffic lights in a cellular automaton model for city traffic. *Physical Review E*, vol. 64, n° 5.
- Bull L., Sha'Aban J., Tomlinson A., Addison J., Heydecker B. (2004). Towards distributed adaptive control for road traffic junction signals using learning classifier systems. In *Applications of Learning Classifier Systems*, p. 276–299. Springer Berlin Heidelberg.
- Camponogara E., Kraus Jr W. (2003). Distributed learning agents in urban traffic control. In *Progress in Artificial Intelligence*, p. 324–335. Springer.
- Carlino D., Boyles S., Stone P. (2013). Auction-based autonomous intersection management. In *Intelligent Transportation Systems - (ITSC), 16th International IEEE Conference on*.
- Da Silva B., Basso E., Bazzan A., Engel P. (2006). Dealing with non-stationary environments using context detection. In *Proceedings of the 23rd international conference on Machine learning*, p. 217–224.
- De Oliveira D., Bazzan A. (2006). Traffic lights control with adaptive group formation based on swarm intelligence. In *Ant Colony Optimization and Swarm Intelligence*, p. 520–521.

- De Oliveira D., Bazzan A. (2007). Swarm intelligence applied to traffic lights group formation. *Anais do VI Encontro Nacional de Inteligência Artificial (ENIA)*, p. 1003–1112.
- Doniec A., Espié S., Mandiau R., Piechowiak S. (2005). Dealing with multi-agent coordination by anticipation : Application to the traffic simulation at junctions. In *Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS)*.
- Dresner K., Stone P. (2004). Multiagent traffic management : A reservation-based intersection control mechanism. In *Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- Dresner K., Stone P. (2005). Multiagent traffic management : An improved intersection control mechanism. In *Proceedings of the Fourth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- Dresner K., Stone P. (2008). A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, p. 591–656.
- Espié S. (1995). Archisim, multi-actor parallel architecture for traffic simulation. In *Proceedings of the Second World Congress on Intelligent Transport Systems*, vol. 4.
- Fabiunke M. (1999). Parallel distributed constraint satisfaction. In *Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*.
- Ferreira E., Khosla P. (2000). Multi agent collaboration using distributed value functions. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, p. 404–409.
- Fitzpatrick S., Meertens L. (2001). An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs. In *Proc. 1st Symp. on Stochastic Algorithms : Foundations and Applications*.
- Fok C., Hanna M., Gee S., Au T., Stone P., Julien C. *et al.* (2012). A platform for evaluating autonomous intersection management policies. In *Third International Conference on Cyber-Physical Systems (ICCPS)*, p. 87–96.
- France J., Ghorbani A. (2003). A multiagent system for optimizing urban traffic. In *Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on*, p. 411–414.
- Gechter F., Contet J., Lamotte O., Galland S., Koukam A. (2012, mai). Virtual intelligent vehicle urban simulator : Application to vehicle platoon evaluation. *Simulation Modelling Practice and Theory (SIMPAT)*, vol. 24, p. 103-114.
- Gershenson C. (2004). Self-organizing traffic lights. *complex Systems*, vol. 16, n° 1, p. 29–53.
- Guler S. I., Menendez M., Meier L. (2014). Using connected vehicle technology to improve the efficiency of intersections. *Transportation Research Part C : Emerging Technologies*, vol. 46, p. 121–131.
- Henry J., Farges J., Tuffal J. (1984). The PROLYN real time traffic algorithm. In *4th IFAC/IFIP/IFORS Conference On Control*.
- Hounsell N., Shrestha B. (2012). A new approach for co-operative bus priority at traffic signals. *Intelligent Transportation Systems*, vol. 13, n° 1, p. 6 -14.
- Hunt P. B., Robertson D. I., Bretherton R. D., Winton R. I. (1981). *SCOOT - A Traffic Responsive Method of Coordinating Signals*. Rapport technique. TRRL.
- Köhler E., Möhring R., Wunsch G. (2005). Minimizing total delay in fixed-time controlled traffic networks. In *Operations Research Proceedings 2004*, p. 192–199.

- Kosonen I. (2003). Multi-agent fuzzy signal control based on real-time simulation. *Transportation Research Part C : Emerging Technologies*, vol. 11, n° 5, p. 389–403.
- Lowrie P. (1982). The sydney coordinated adaptive traffic system-principles, methodology, algorithms. In *International Conference on Road Traffic Signalling, 1982, London, UK*.
- Mohring R., Nokel K., Wunsch G. (2006). A model and fast optimization method for signal coordination in a network. In *Control in Transportation Systems*, p. 73–78.
- Naumann R., Rasche R. (1997). *Intersection collision avoidance by means of decentralized security and communication management of autonomous vehicles*. Univ.-GH, SFB 376.
- Naumann R., Rasche R., Tacke J., Tahedi C. (1997). Validation and simulation of a decentralized intersection collision avoidance algorithm. In *Intelligent Transportation System (ITSC'97), IEEE*, p. 818–823.
- Nunes L., Oliveira E. (2004). Learning from multiple sources. In *Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems*, p. 1106–1113.
- Richter S., Aberdeen D., Yu J. (2007). Natural actor-critic for road traffic optimization. In B. Schölkopf, J. Platt, T. Hofmann (Eds.), *Advances in Neural Information Processing Systems 19*. Cambridge, MA, MIT Press.
- Robertson D. (1969). *TRANSYT : A Traffic Network Study Tool*. Road Research Laboratory.
- Rochner F., Prothmann H., Branke J., Müller-Schloer C., Schmeck H. (2006). An organic architecture for traffic light controllers. In *Proceedings of the Informatik 2006 –Informatik fürMenschen–I Jahrestagung (1)*, p. 120–127.
- Russell S., Norvig P. (1995). *Artificial Intelligence : A Modern Approach*. Englewood Cliffs, NJ : prentice Hall.
- Sánchez J., Aguirre J. (2007). Traffic light control through agent-based coordination. In *Artificial Intelligence and Applications*, p. 163–168.
- Scheuer A., Simonin O., Charpillat F. (2009). Safe longitudinal platoons of vehicles without communication. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, p. 2835–2840.
- Steingrover M., Schouten R., Peelen S., Nijhuis E., Bakker B. (2005). Reinforcement learning of traffic light controllers adapting to traffic congestion. In *Proceedings of the Belgium-Netherlands Artificial Intelligence Conference (BNAIC)*, p. 216–223.
- Tlig M. (2015). *Coordination locale et optimisation distribuée du trafic de véhicules autonomes dans un réseau routier*. Thèse, Université de Lorraine. (<https://hal.inria.fr/tel-01251168>)
- Treiber M., Kesting A. (2010). An open-source microscopic traffic simulator. *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, n° 3, p. 6–13.
- Weixiong Z., Guandong W., Zhao X., Lars W. (2005). Distributed stochastic search and distributed breakout : properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, vol. 161, n° 1–2, p. 55–87.
- Wiering M. (2000). Multi-agent reinforcement learning for traffic light control. In *Proceedings of the International Conference on Machine Learning*, p. 1151–1158.
- Yin B., Dridi M., El Moudni A. (2015). Adaptive traffic signal control for multi-intersection based on microscopic model. In *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*, p. 49–55.

TABLE 5: Présentation de toutes les solutions citées selon les différentes dimensions identifiées. ?="non identifié". Temps="Temps de traversée des véhicules". Prog. Dyn.="programmation dynamique". MIP="Mixed Integer Programming". AE="algorithmes évolutionnaires". QL="Q-learning". QL DC="App. par renforcement avec détection de contexte. NAC="Natural Actor-Critic".

Approches	Perceptions	Adaptation	Var de contrôle	Espace	Critère	(Dé) Centralisation	Algo d'optim
TRANSYT (Robertson, 1969)	nbre de véhicules	moyenne	plan des feux	?	temps	centralisé	Hill-Climbing
SCOOT (Hunt <i>et al.</i> , 1981)	nbre de véhicules	moyenne	plan des feux	?	temps	décentralisé	Hill-Climbing
SCATS (Lowrie, 1982)	nbre de véhicules	moyenne	plan des feux	?	temps	décentralisé	Hill-Climbing
PRODYN (Henry <i>et al.</i> , 1984)	nbre de véhicules	instantanée	plan des feux	?	temps	décentralisé	Prog dyn
(Brockfeld <i>et al.</i> , 2001)	densité des flux	moyenne	phases	discret	retard	centralisé	mathématique
(Gershenson, 2004)	véhicules en attente	moyenne	plan des feux	discret	retard	décentralisé	à base de règles
(Köhler <i>et al.</i> , 2005)	temps perdu	moyenne	phases	continu	retard	centralisé	MIP
(Mohring <i>et al.</i> , 2006)	temps perdu	moyenne	phases	continu	retard	centralisé	MIP
(De Oliveira, Bazzan, 2006)	phénomènes	instantanée	plan des feux	discret	temps	décentralisé	à base de règles
(France, Ghorbani, 2003)	densité des flux	moyenne	plan des feux	discret	flux	décentralisé	à base de règle
(Roehner <i>et al.</i> , 2006)	nbre de véhicules	instantanée	plan des feux	indépendant	retard	centralisé	AE
(Kosonen, 2003)	temps perdu	instantanée	plan des feux	continu	retard	décentralisé	logique floue
(Sánchez, Aguirre, 2007)	files d'attente	instantanée	plan des feux	continu	retard	décentralisé	négociation
(Bhourri <i>et al.</i> , 2011)	files d'attente	instantanée	plan des feux	discret	temps	décentralisé	à base de règles
(Hounsell, Shrestha, 2012)	files d'attente	instantanée	plan des feux	discret	temps	décentralisé	à base de règles
(Ferreira, Khosla, 2000)	nbre de véhicules	instantanée	plan des feux	continu	temps	décentralisé	QL
(Camponogara, Kraus Jr, 2003)	files d'attente	instantanée	plan des feux	discret	temps	décentralisé	QL
(Bazzan, 2005)	files d'attente	moyenne	phases	continu	temps	décentralisé	théorie des jeux
(Doniec <i>et al.</i> , 2005)	véhicules	instantanée	priorités	continu	matrice de gain	décentralisé	théorie des jeux
(Wiering, 2000)	temps d'attente	instantanée	plan des feux	discret	temps	dé. inters. + véhic.	QL
(Steingrover <i>et al.</i> , 2005)	temps d'attente	instantanée	plan des feux	discret	temps	décentralisé	QL et Prog dyn
(Da Silva <i>et al.</i> , 2006)	nbre de véhicules	instantanée	plan des feux	discret	nbre de véhicules	décentralisé	QL DC
(Nunes, Oliveira, 2004)	densité des flux	instantanée	plan des feux	continu	taux d'occupation	centralisé	QL+AE
(Richter <i>et al.</i> , 2007)	nbre de véhicules	instantanée	plan des feux	discret	nbre de véhicules	centralisé	NAC
(Yin <i>et al.</i> , 2015)	files d'attente	instantanée	plan des feux	discret	nbre de véhicules	décentralisé	Prog dyn
(Guler <i>et al.</i> , 2014)	véhicules	instantanée	plan des feux	continu ?	temps / nbr arrêts	décentralisé	mathématique
(Naumann, Rasche, 1997)	véhicules	instantanée	véhicules	indépendant	temps	déc. sur les véhicules	négociation
(Carlino <i>et al.</i> , 2013)	véhicules	instantanée	véhicules	continu	temps	décentralisé	à base de règles
(Balan, Luke, 2006)	historique véhicule	instantanée	plan des feux	indépendant	retard	décentralisé	à base de règles
(Dresner, Stone, 2008)	véhicules	instantanée	véhicules	continu	temps	décentralisé	à base de règles
Notre approche : intersection	véhicules	instantanée	véhicules	continu	temps / énergie	décentralisé	à base de règles
Notre approche : réseau	densité des flux	moyenne	déphasages	continu	temps / énergie	décentralisé	Hill-Climbing

App. Avancées

IA