

Un algorithme distribué de contrôle des feux de circulation sur plusieurs intersections par un réseau de capteurs sans fil

Sébastien Faye*, Claude Chaudet*, Isabelle Demeure*

*Institut Mines-Telecom, Telecom ParisTech, CNRS LTCI UMR 5141, Paris, France
prenom.nom@telecom-paristech.fr,
<http://perso.telecom-paristech.fr/~nom>

Résumé. Dans cet article, nous étudions le scénario consistant à utiliser un réseau de capteurs sans fil afin de contrôler les feux de circulation d'un réseau de transport urbain et d'optimiser son fonctionnement en réduisant le temps moyen d'attente des usagers. Le faible coût des capteurs autorisant un nombre de points de mesure important, il est possible de résoudre les problèmes de congestion localement, sans passer par un processus de décision centralisé généralement coûteux. Nous nous intéressons ici à un réseau couvrant plusieurs intersections et nous proposons un algorithme distribué permettant aux capteurs de coopérer et d'adapter constamment la politique de feux aux conditions de trafic. Nos simulations montrent que notre algorithme, appliqué à plusieurs intersections, permet d'améliorer considérablement le temps moyen d'attente des usagers en s'adaptant mieux aux variations de trafic, en comparaison à des méthodes plus traditionnelles.

1 Introduction

Nous nous intéressons à la gestion des feux de circulation dans un réseau routier urbain afin de réduire les embouteillages et les ralentissements. Traditionnellement, les intersections à feux de circulation sont gérées par des contrôleurs qui définissent et mettent en œuvre une séquence (appelée plan de feux) prédéterminée, alternant feux verts et rouges sans prendre en compte les variations de trafic.

Dans certaines zones urbaines, des détecteurs comptabilisent les véhicules présents sur chacune des voies d'une intersection. Ces dispositifs permettent de mettre en place une gestion dynamique des plans de feux, utilisant des méthodes dites adaptatives, réagissant aux conditions de trafic. En effet, la connaissance de la répartition du nombre de véhicules par voie permet d'estimer les valeurs les plus adaptées à un instant donné de variables telles que le temps de feu vert ou encore la séquence de feux permettant de réduire, par exemple, le temps moyen d'attente.

Les architectures actuelles de ces systèmes s'organisent généralement autour d'un contrôleur de feux qui peut prendre une décision locale ou communiquer ses informations à un centre de contrôle (appelé poste de circulation) capable de lui retransmettre une politique en retour. Ce mode de fonctionnement est celui de nombreux réseaux de capteurs opérés. Dans cet article,

nous étudions comment décentraliser cette gestion tout en prenant en compte les interactions entre carrefours proches en utilisant un réseau de capteurs sans fil. Le réseau de capteurs considéré ici n'est composé que de capteurs fixes, afin de s'affranchir des problèmes liés à la masse critique d'utilisateurs à atteindre avant d'aboutir à une application fonctionnelle. Il est toutefois envisageable d'intégrer les véhicules à l'architecture dans un second temps. Dans le cas des systèmes de transport intelligents (STI), l'utilisation de nombreux capteurs de faible coût représente une alternative aux boucles électromagnétiques. La densité de déploiement qu'il est possible d'atteindre permet de détecter et de traiter les problèmes de congestion localement et d'aboutir à un STI distribué et plus réactif ne reposant pas sur une seule entité de décision.

Dans cet article, extension de Faye et al. (2012b), après une présentation des travaux en relation avec ce travail en section 2, nous présentons une architecture pour le réseau de capteurs déployé à une intersection et présentons deux algorithmes permettant de sélectionner les séquences de mouvements et les durées de feux verts permettant de réduire le temps moyen d'attente à une intersection. Le premier algorithme, évoqué en section 3, s'applique au cas d'une intersection seule et isolée et se distingue de l'état de l'art par son caractère adaptatif. Il ne se base pas sur des séquences prédéterminées de mouvements mais sélectionne, lors de chaque feu vert, les mouvements suivants en fonction des données collectées par le réseau de capteurs, réduisant les temps d'attente sans introduire de famine. L'algorithme TAPIOCA, présenté en section 4, est une généralisation au cas d'intersections multiples permettant de prendre en compte les interactions entre carrefours voisins afin de favoriser le délestage ou de créer des vagues vertes. Nous montrons ensuite par des résultats de simulation présentés en section 5, que ces deux algorithmes permettent de réduire d'environ 26 % le temps d'attente des usagers par rapport à un plan de feux réel tel que celui qui est mis en œuvre dans la ville d'Amiens et de 27 % par rapport à un scénario basé sur la ville de Cologne. En outre, l'algorithme TAPIOCA s'adapte plus rapidement aux évolutions de trafic que des algorithmes locaux ou que des séquences prédéterminées.

2 Travaux liés

2.1 Capteurs électromagnétiques spécifiques aux STI

Les STI adaptatifs qui utilisent des réseaux de capteurs sans fil (Yousef et al., 2010; Tubaishat et al., 2008; Zhou et al., 2010, 2011; Zou et al., 2009) sont généralement basés sur un modèle de file d'attente, qui nécessite la connaissance du nombre de véhicules de chacune des voies d'une intersection, ou celle du processus d'arrivée des véhicules. Si les radars et les boucles électromagnétiques sont largement répandus pour ce type d'acquisition, leur coût et leur temps d'installation les rendent peu accessibles, notamment pour les intersections isolées ou zones urbaines pauvres. Ce type de détecteur est capable d'enregistrer une signature propre à chaque véhicule, en mesurant les perturbations produites sur le champ magnétique terrestre lors de leur passage. Ces dispositifs sont capables de comptabiliser 99% des véhicules les traversant (Cheung et al., 2005). Corredor et al. (2008) montrent qu'ils sont plus efficaces que les boucles électromagnétiques, car ils sont plus réactifs, plus faciles d'installation et augmentent le nombre de points de mesures possibles. Knaian (2000) évoque un coût très bas de fabrication – moins de 30 \$ par unité – et une taille comparable à une pièce de monnaie. Cette technologie est inhabituelle dans le domaine des transports mais est toutefois simple à mettre

en place et pourrait, dans de nombreux cas, être remplacée par des capteurs virtuels (caméras) afin de détecter le passage de véhicules (sous réserve d'avoir de bonnes conditions météo et un bon angle de vue). Collotta et al. (2012) abordent cette possibilité de réseau hybride, formée de caméras et de capteurs sans fil afin d'avoir un meilleur contrôle de l'infrastructure et des usagers.

Comme le précisent Collotta et al. (2012), les réseaux de capteurs sans fil doivent communiquer sur de courtes distances en utilisant un minimum d'énergie. Pour satisfaire cette condition, les capteurs utilisés dans le cadre du trafic routier utilisent généralement le standard *IEEE 802.15.4*, définissant une couverture d'environ 50-100 mètres par capteur, une vitesse de transmission des données de 250 Kbps et une faible consommation d'énergie. Le protocole *IEEE 802.15.4/ZigBee* est généralement utilisé avec de tels réseaux afin d'assurer des communications de faible distance avec une économie d'énergie.

2.2 Architectures de réseaux de contrôle d'un carrefour

Le modèle d'intersection sur lequel s'appuient la plupart des contributions de la littérature est représenté en figure 1. Il comporte quatre directions (N, E, S, O) et chacune de ces directions est, dans cet article, décomposée en deux voies entrantes et deux voies sortantes. La voie entrante de gauche est réservée aux véhicules tournant à gauche et la voie de droite aux véhicules allant tout droit ou tournant à droite. On appelle *cycle* ou *plan de feux* une séquence complète dans laquelle chaque feu sera passé au moins une fois au vert. Un cycle est composé de *phases*, c'est-à-dire des combinaisons temporisées de feux verts et rouges. Les feux verts autorisent les usagers à effectuer un ou plusieurs *mouvements*, définis par une direction de départ et d'arrivée (*exemple : WE sur la figure 1 dénote le mouvement d'ouest en est*). Un contrôleur de feux détermine, pour chaque *phase*, le ou les *mouvements* obtenant le feu vert ainsi que la durée de la phase.

On considère souvent, dans la littérature que chaque voie entrante est équipée de deux capteurs : l'un situé au niveau du feu afin de comptabiliser les départs et l'autre à une distance fixe, afin de comptabiliser les arrivées. La distance entre ces deux capteurs est variable et doit être suffisamment longue pour mesurer l'évolution de la file d'attente lors d'un feu vert sans toutefois être trop importante afin de minimiser les erreurs dues aux changements de file des usagers. La littérature préconise une distance de l'ordre de 5 à 8 véhicules, (Yousef et al., 2010; Tubaishat et al., 2008), ou une distance fonction du temps maximum de feu vert (Zhou et al., 2010). Si ce modèle est couramment utilisé, la présence de deux capteurs par voie n'est pas indispensable. Un seul capteur suffit à estimer le processus d'arrivées, mais fournit un résultat moins précis. Un capteur par direction permet également d'obtenir des résultats. Cependant, le nombre de voies couvertes est fonction de la portée de détection du capteur (Zou et al., 2009).

2.3 Algorithmes de contrôle des feux de circulation

Les travaux évoqués ci-dessus s'insèrent le plus souvent dans un système global administré par un centre de contrôle distant ou d'une station de base chargée de prendre les décisions. Profiter des possibilités de répartition offertes par un réseau de capteurs permet d'améliorer la réactivité générale de détection et d'action, ainsi que la tolérance aux fautes, supprimant la dépendance au point central qui représente un point unique de défaillance. Nous présentons ici

Contrôle des feux de circulation par un réseau de capteurs sans fil

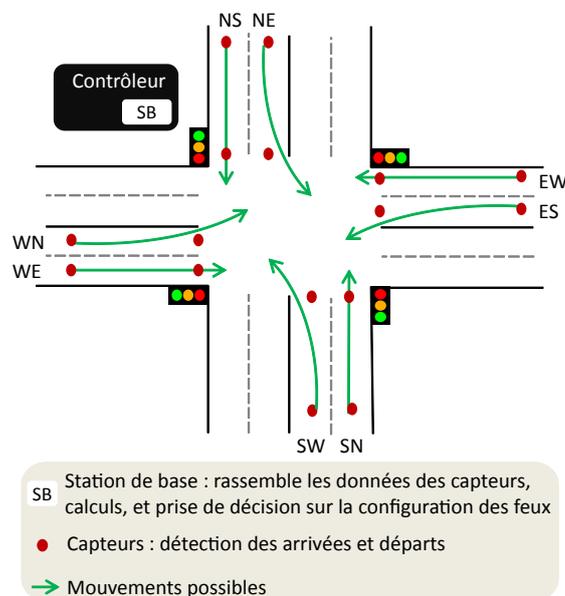


FIG. 1 – *Modèle typique d'intersection à 4 directions.*

les contributions visant à faire fonctionner une intersection de manière autonome. Les notations utilisées ici et dans le reste de l'article sont détaillées dans le tableau 1.

Notation	Description
T_G	Durée de feu vert.
T_s	Temps de démarrage des véhicules lorsqu'un feu passe au vert.
T_H	Temps de traversée d'un feu par un véhicule.
T_{max}	Temps maximum autorisé de feu vert.
N_i, N^y	Nombre de véhicules sur une file d'attente (sur une voie i), ou sur l'ensemble des voies composant le mouvement y .
λ	Fréquence moyenne d'arrivée des véhicules.
μ	Fréquence moyenne à laquelle les véhicules quittent l'intersection lorsqu'un feu devient vert.

TAB. 1 – *Principales notations utilisées dans cet article.*

Yousef et al. (2010) proposent un modèle reposant sur la théorie des files d'attente et se basant sur un modèle d'intersection conforme à la figure 1 : huit mouvements sont possibles (deux par direction) et chaque voie est dotée de deux capteurs. L'idée principale de l'article est de modéliser chaque mouvement y comme une file M/M/1 et de sélectionner les phases les plus adaptées constituées de mouvements n'entrant pas en conflits. Par exemple, sur la figure 1, les mouvements EW et WE peuvent se faire simultanément sans conflit, tout comme WN

et WE, ou encore WN et ES. Les longueurs des files associées à chaque mouvement (N^y) et le temps moyen d'attente ($AWT = N^y/\lambda$) sont déterminés en utilisant la loi de Little. Soit T_G le temps de feu vert et T_R le temps de feu rouge, le nombre de véhicules présents sur la voie i au cycle C est égal à $N_i^C = N_i^{C-1} + \lambda T_G - \mu T_G + \lambda T_R$, où μ représente la fréquence moyenne de départ. λT_G et λT_R véhicules arrivent durant le feu vert et le feu rouge respectivement. μT_G véhicules quittent l'intersection durant le feu vert. Basé sur cette équation, l'algorithme proposé par Yousef *et al.* fonctionne de façon gloutonne, sélectionnant à chaque phase la combinaison de mouvements n'entrant pas en conflit et dont la somme des nombres de véhicules à écouler est la plus grande, de manière à minimiser la longueur de file moyenne sur l'intersection. Le temps de feu vert est déterminé proportionnellement à la longueur des files d'attente. Cet algorithme est conçu pour fonctionner sur une intersection. Une extension au cas d'un réseau maillé d'intersections est présentée dans cet article. Chaque intersection est vue comme un ensemble de nœuds communiquant et possédant une vision locale. Le modèle ainsi que le principe de décision restent les mêmes. Les intersections voisines peuvent communiquer et collaborer afin d'agir sur le temps de feu et créer ainsi des *vagues vertes* (feux verts successifs). En fonction du délai séparant deux intersections et des stops, les intersections sont en mesure de manipuler le temps des feux (rallongement, synchronisation, etc.).

Tubaishat et al. (2008) proposent une solution sur la base de trois ensembles de phases prédéterminées, composés respectivement de 4, 5 et 8 phases. Les cycles sont définis sur l'un de ces ensembles en ordonnant les phases en fonction de la longueur des files. L'évaluation est réalisée avec le simulateur *Green Light District simulator* (Wiering et al., 2004) développé initialement pour les systèmes d'agents mobiles, mais incapable d'adapter le temps de feu dynamiquement. Ces propositions supposent l'utilisation de phases statiques et prédéfinies, réduisant les possibilités d'adaptation au trafic. Enfin, considérer la longueur des files d'attente comme seul critère d'optimisation peut entraîner des situations de famine, une voie de faible importance pouvant être mise de côté longtemps face aux voies principales.

Zhou et al. (2010) proposent un plan de feux en se basant sur les combinaisons de mouvements pouvant intervenir simultanément sans aucun conflit. Sur un carrefour comportant 4 voies, il existe 8 combinaisons distinctes de mouvements sans conflit. Leur algorithme sélectionne ensuite la séquence de phases composant un cycle, selon plusieurs critères : la présence de véhicules prioritaires, la durée des périodes où aucune détection n'est effectuée, les cas de famine, le temps d'attente total et la longueur des files d'attente. Le modèle proposé par Zhou et al. (2010) se base cependant sur des hypothèses irréalistes, imposant aux véhicules d'être de même type et de rouler à la même vitesse. Dans un second article (Zhou et al., 2011), les auteurs étendent ce modèle au cas des multiples intersections communicantes. L'architecture définie est composée d'une intersection centrale, encadrée par 4 intersections : une par direction, en reprenant le modèle d'intersection présenté sur la figure 1. La répartition des capteurs diffère toutefois de leur premier modèle : cette fois-ci, les capteurs sont distribués de part et d'autre du réseau, particulièrement entre les intersections, de manière à mesurer avec précision les différents flux et échanges. Traditionnellement, la littérature se base sur un temps de cycle commun aux intersections. Ici, les auteurs s'intéressent à la définition même des flux de circulation, en souhaitant augmenter le débit et réduire en conséquence le temps moyen d'attente. Chaque intersection exécute son propre algorithme afin de sélectionner les cas de phases les plus adaptées, en se basant sur des critères locaux, ainsi que sur des critères reflétant l'influence

des intersections voisines. Le temps de phase est ajusté afin de prendre en compte les intersections voisines et de créer des vagues vertes : un temps minimal est fixé, permettant de faire passer les véhicules présents sur l'intersection et une extension de ce temps est possible sous réserve de ne pas dépasser une borne T_{max} . Cette rallonge, ainsi que le critère d'influence, sont déterminés en fonction des relations existantes entre deux intersections, des phases actives au moment de l'exécution de l'algorithme. Ce modèle possède toujours de grosses limitations : les voitures doivent notamment circuler à la même vitesse afin de pouvoir calculer le temps de traversée entre deux intersections.

Zou et al. (2009) définissent un temps de feu vert en utilisant des règles de logique floue : chaque durée de feu vert est déterminée en se basant sur un ensemble d'intervalles de charge. Par exemple, si moins de cinq véhicules par minute sont détectés, le feu vert est de 10 secondes.

Outre ces quelques articles, très peu de travaux s'intéressent à la relation entre STI et réseaux de capteurs sans fil. Si le cas d'une intersection est tout de même abordé par quelques auteurs, l'extension à de multiples intersections est rarement imaginée, ou alors de manière complexe.

3 Contrôle des feux de circulation d'une intersection unique

3.1 Architecture du réseau de capteurs

Nous souhaitons parer les difficultés des contributions précédentes en concevant un algorithme de gestion des feux de circulation sur une intersection qui soit le plus réactif et dynamique possible, s'appuyant sur un réseau de capteurs communicants et ne nécessitant pas de station de base. Nous traitons dans cette section le cas d'une simple intersection, que nous détaillons totalement dans Faye et al. (2012a) et qui est conforme au modèle de la figure 1 : sur chaque voie, deux capteurs sont présents, l'un au niveau du feu (comptabilisant les départs), l'autre à une distance fixe du feu (comptabilisant les arrivées). Ce modèle nous permet de mesurer correctement la longueur des files d'attente et de définir, sur la base de cette estimation, un score pour chaque voie. Ce score rend compte de plusieurs objectifs auxquels il est possible de donner un poids relatif. Il est possible de privilégier ou de limiter l'influence d'un critère en fonction d'une politique transmise par le contrôleur ou par des intersections voisines (cf. sec. 4). Dans l'architecture de carrefour que nous proposons, la station de base est réduite à une interface vers le contrôleur de feu et peut toutefois participer, au même titre que les autres nœuds, au processus de décision.

L'architecture du réseau de capteurs local comporte, sur chaque voie de circulation, deux capteurs, à l'avant (*niveau 2*) et à l'arrière (*niveau 1*), mesurant respectivement les départs et les arrivées des véhicules. La distance entre ces deux capteurs dépend de l'infrastructure routière. Le but étant de comptabiliser le nombre de véhicules qu'il est possible de faire passer en un temps de T_{max} secondes, nous proposons de fixer cette distance à $N * L_{veh}$. $N = \frac{T_{max} - T_s}{T_H}$ est le nombre de véhicules passant théoriquement en T_{max} secondes, sachant que chaque véhicule met un temps de T_H secondes à traverser le feu. L_{veh} est la longueur moyenne d'un véhicule. Pour nos simulations, nous fixons dans le reste du document $T_s = 4 s$, $T_H = 2 s$ et $L_{veh} = 6 m$, comme le préconise Gordon et al. (2005). Toutefois, ces valeurs peuvent être ajustées au cours de la vie du réseau. Il est possible, à cet effet, de rendre le capteur

arrière mobile et d'adopter une approche empirique, ce type de capteurs pouvant fonctionner sur batteries.

Les capteurs arrière mesurent les arrivées en permanence alors que les capteurs avant ne fonctionnent qu'au feu vert. Étant moins souvent sollicités et plus proches de la passerelle, ils peuvent être sollicités pour des calculs et c'est *a priori* parmi eux que peuvent être sélectionnés des nœuds d'agrégation des données et de calcul (*niveau 3*) parmi lesquels sera élu dynamiquement un nœud principal (*niveau 4*), en charge de prendre les décisions et de les communiquer au contrôleur. Cette architecture est basée sur plusieurs élections qui peuvent être révisées à intervalles réguliers pour optimiser le fonctionnement en prenant en compte les niveaux de batteries ou les pannes des nœuds.

3.2 Principe général

Notre approche se distingue de la littérature car son architecture distribuée permet de mieux répartir la charge, tolère les modifications du réseau et facilite l'extension du réseau en interconnectant différentes intersections. Nous adoptons par ailleurs une approche plus dynamique qui cherche à calculer le changement des feux phase après phase et non plus cycle après cycle. Ceci permet notamment une réactivité plus nette face à des pics de circulation soudains, mais nécessite une gestion explicite des situations de famine afin de ne pas tomber dans les mêmes travers que les solutions analogues précédentes. Enfin, nous nous reposons sur une matrice des conflits définissant les mouvements qu'il est possible ou non de faire circuler en même temps afin de créer une phase. Par exemple, sur l'intersection de la figure 1, le mouvement SN peut circuler en même temps que le mouvement NS. A l'inverse, le mouvement SN ne peut pas être actif en même temps que le mouvement WE, pour des raisons de sécurité.

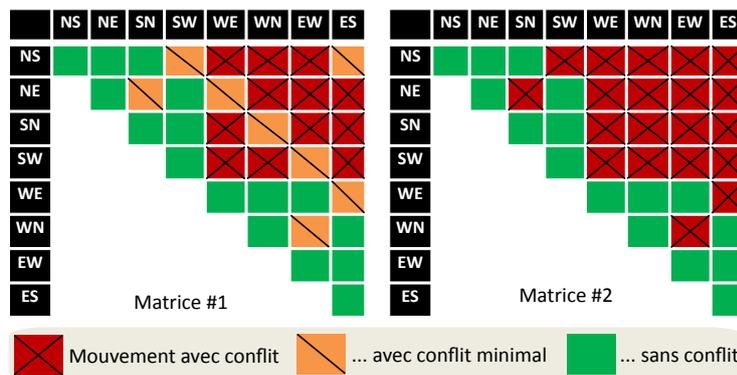


FIG. 2 – Modèle typique d'intersection à 4 directions.

La figure 2 illustre deux matrices utilisées dans Faye et al. (2012a). Tandis que la matrice #2 n'autorise aucun conflit entre les mouvements, la matrice #1 permet d'exécuter des mouvements entrant en conflits légers (par exemple : NS et SW), afin d'augmenter le nombre de mouvements possibles d'une phase. De tels conflits peuvent être mis sous condition (e.g. ne les autoriser uniquement que si le nombre de véhicules touchés est faible). Notons également que cette matrice peut être étendue afin de représenter plusieurs niveaux de gravité des conflits.

Nous montrons dans Faye et al. (2012a) que l'autorisation de certains conflits est plus efficace en terme de temps d'attente qu'une solution plus stricte.

3.3 Algorithme de contrôle

Une fois la répartition des véhicules connue et les données agrégées au sein du nœud de *niveau 4*, chaque mouvement y possible de l'intersection est évalué par un score $SL(y)$ en vue d'une sélection. Deux objectifs sont ici pris en compte : d'une part le nombre de véhicules présents sur les voies du mouvement y , N^y , d'autre part le temps d'attente depuis la dernière sélection du mouvement y , T_F^y . Le calcul du score $SL(y)$ pour un mouvement y parmi les M mouvements possibles s'effectue selon l'algorithme suivant :

- Si $N^y = 0$, le score est nul.
- Sinon, le score de y est calculé à partir de T_F^y et N^y , normalisés et pondérés par des poids W_{T_F} et w_N afin de privilégier un objectif ou l'autre :

$$SL(s, d) = \alpha \cdot \left(\frac{N^{(s,d)}}{\sum_{\{a,b\} \in D} N^{(a,b)}} \right) + \beta \cdot \left(\frac{T_F^{(s,d)}}{\sum_{\{a,b\} \in D} T_F^{(a,b)}} \right)$$

Finalement, les combinaisons possibles de mouvements sont calculés : une combinaison est une union de plusieurs mouvements étant autorisés par la matrice des conflits à circuler les uns avec les autres. La phase sélectionnée est la combinaison pour laquelle la somme des scores des mouvements qui la composent est la plus élevée. Le temps de feu correspondant, pour sa part, est borné à T_{max} et est calculé de manière à laisser, dans un cas idéal, le temps aux véhicules présents sur l'intersection de passer.

T_{max} a une influence sur les performances du réseau. Une valeur trop importante laissera les véhicules s'accumuler sur les voies non sélectionnées alors qu'une valeur trop faible ne permettra pas d'écouler le trafic efficacement. Les premiers résultats obtenus nous ont permis de conclure qu'une valeur située entre 30 s et 45 s donnait les meilleurs résultats sur les simulations effectuées. La pondération des objectifs est variable en fonction de la matrice des conflits en place : si beaucoup de mouvements sont autorisés à circuler en même temps, le critère relatif à la longueur des files d'attente doit primer sur celui reflétant les situations de famine ; à l'inverse, la gestion des situations de famine doit être prépondérante si les mouvements autorisés sont peu nombreux. Nous montrons dans Faye et al. (2012a) que le temps d'attente engendré par cette méthode est nettement inférieur à une méthode reposant sur un cycle prédéterminé.

3.4 Passages piétons

La gestion des piétons sur une intersection peut être réalisée facilement, en intégrant les passages piétons possédant des feux à la matrice des conflits. Pour chaque phase, il suffit en effet de donner le feu vert aux passages piétons n'entrant pas en conflit avec les files de véhicules sélectionnées. De la même manière que pour les mouvements, un passage piéton peut également posséder un ou plusieurs conflits autorisés : des véhicules peuvent interférer avec le passage piéton, mais ne sont pas considérés comme dangereux ou chargés, ils ne sont pas prioritaires.

Dans certains cas, aucun conflit avec les passages piétons n'est désiré et il est impossible de configurer une matrice des conflits parallélisant la distribution des feux verts des passages piétons et des files de véhicules. Un passage piéton peut alors avoir les mêmes propriétés qu'une file de véhicules, en possédant un objectif relatif à la famine et un autre objectif indiquant la présence de piétons, dont l'importance peut être calculée à l'aide de caméras, de capteurs piézo-électriques ou simplement de boutons poussoirs.

4 Contrôle des feux de circulation sur de multiples intersections

Nous proposons dans cette section une généralisation de l'architecture et de l'algorithme évoqués à la section précédente au cas d'intersections multiples et voisines. Nous supposons que chaque intersection est capable de communiquer avec ses voisines par l'intermédiaire du réseau de capteurs lui-même ou par une liaison globale (filaire, cellulaire, etc.). Le choix du mode d'interconnexion dépend essentiellement de la distance séparant deux feux de circulation et dépend de choix liés au génie civil. L'architecture et l'algorithme que nous proposons est insensible au mode d'interconnexion et est facilement généralisables à tout type d'intersection, hormis les impasses.

4.1 Architecture du réseau d'une intersection

Contrairement au cas d'une intersection simple, il est nécessaire, dans le cas de multiples intersections, de connaître la direction empruntée par chacun des véhicules afin de caractériser les processus d'arrivée à un carrefour en fonction des processus de départ des carrefours voisins.

L'architecture du réseau déployé au niveau d'une intersection interconnectée à ses voisines est représentée sur la figure 3. Les nœuds en charge de la comptabilisation des arrivées, notés *CS*, sont toujours placés à une distance fixe du feu signalétique : ils permettent d'identifier la source de chaque mouvement. Les nœuds en charge de la comptabilisation des départs, notés *CD*, sont, pour leur part, situés non plus au niveau des feux, mais à l'entrée de chaque voie de sortie, afin d'identifier les destinations. Ainsi, ce modèle permet de différencier chacun des mouvements possibles d'une intersection, même si certains sont regroupés sur les mêmes voies : en se basant sur la signature électromagnétique laissée par chaque véhicule (ou indirectement une fonction de hachage retournant un identifiant unique), les capteurs sont capables de savoir par où arrive un véhicule et par où il quitte l'intersection (*niveau 1*). L'identification des véhicules permet de comptabiliser avec précision la longueur des files d'attente, notamment dans le cas où plusieurs voies mènent à la même destination : les arrivées sont comptabilisées correctement, mais sans identification, il est impossible de connaître la voie d'origine d'un départ et mettre à jour le nombre de véhicules correspondant. Au-dessus de cette première couche permettant la collecte des données, un nœud est élu pour chaque direction, parmi les capteurs *CD* pour les mêmes raisons que dans le premier modèle. Chaque direction étant *a priori* liée à une intersection voisine différente, chaque nœud *CD* de *niveau 2* est en charge de la communication correspondante, en plus d'agréger les données des mouvements dont la source est sur l'une de ses voies. Enfin, un ultime maître est élu sur l'intersection, de *niveau 3*

Contrôle des feux de circulation par un réseau de capteurs sans fil

et permet d'effectuer les calculs finaux, puis d'envoyer un ordre à l'interface, liée au contrôleur. Le contrôleur peut être rattaché à une ou plusieurs intersections.

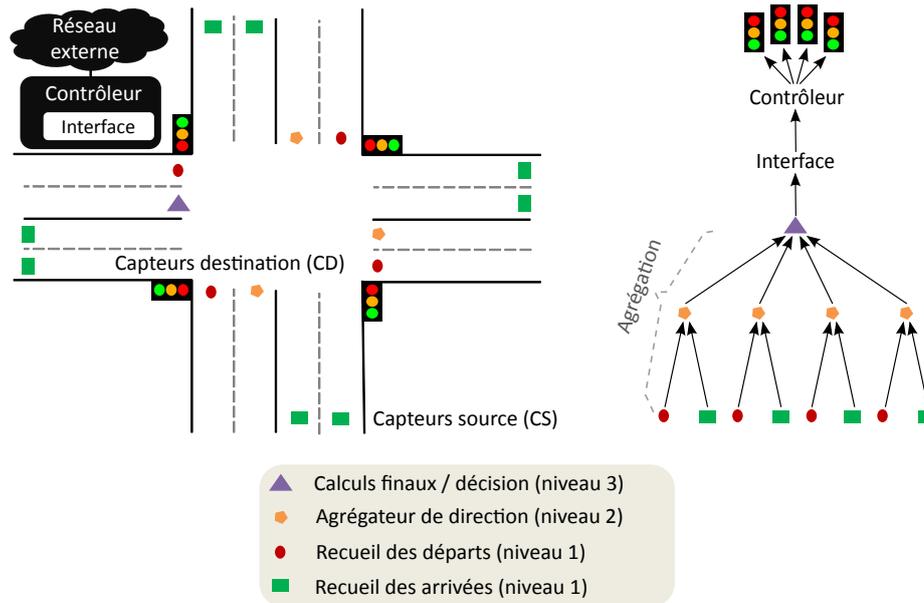


FIG. 3 – *Modèle d'intersection, avec un capteur pour mesurer les sources, un autre pour les destinations.*

Les communications entre intersections voisines utilisent chacun des nœuds du réseau de capteurs sans fil, auquel on peut adjoindre des nœuds additionnels relais ou d'un réseau annexe en cas de distance trop élevée. Le routage d'une information peut être implanté statiquement via des tables de routage, ou de manière plus dynamique via, par exemple, une diffusion géographique. Plusieurs méthodes d'élection distribuée ont été proposées dans la littérature, opérant en temps logarithmique voire inférieur (Jurdziński et al., 2002) et peuvent convenir à cette tâche. Tout comme pour notre première architecture, cette élection pourrait avoir lieu à intervalles réguliers afin de répartir les coûts. Les critères pourraient être l'énergie, l'encombrement moyen des voies dont les nœuds sont en charge etc.

4.2 Objectifs d'optimisation locaux

Afin de simplifier et de rendre efficace notre algorithme, nous avons identifié trois objectifs représentatifs des critères à prendre en compte sur plusieurs intersections, afin d'évaluer chaque mouvement d'une intersection et de sélectionner les plus importants en fonction des véhicules présents. Considérons une intersection dont nous voulons calculer un score $S(y)$ pour chacun de ses mouvements y . Ce score doit être une combinaison de :

1. **Score local.** Ce premier objectif permet de privilégier les véhicules présents sur l'intersection seule et correspond à la méthode utilisée pour une simple intersection, en 3.3,

à savoir le calcul d'un score local $SL(y)$ pour chaque mouvement y . De cette manière, nous sommes certains de prendre en compte les données présentes localement et non uniquement les données des intersections voisines. Dans le cas où $SL(y)$ est nul, aucun véhicule n'est présent : les deux autres objectifs n'ont pas besoin d'être pris en compte, par conséquent $S(y) = 0$.

2. **Gain espéré.** Cet objectif permet de privilégier les mouvements amenant un nombre important de véhicules vers une direction d'une intersection voisine peu chargée. En effet, il est souvent préférable d'envoyer en priorité beaucoup de véhicules sur des voies peu chargées, plutôt que peu de véhicules sur des voies déjà encombrées. Agir de cette manière permet également aux intersections voisines non concernées (trop chargées) de se décharger avant la prochaine phase. Le gain espéré pour un mouvement y est $GE(y) = N^y - NS_d^j$, avec N^y le nombre de véhicules du mouvement y sur l'intersection i , NS_d^j le nombre de véhicules présents sur les voies de la direction d de l'intersection voisine j .
3. **Influence voisine.** Cet objectif permet une meilleure synchronisation entre les intersections, de manière à créer, si possible, des vagues vertes. Le but est ici de privilégier les mouvements menant à des directions dont le ou les feux vont passer au vert au plus tôt. Toutes les intersections associent déjà un score local $SL(s, d)$ à chaque mouvement (s, d) allant de s à d . Une intersection I_1 évalue tout d'abord l'intensité du trafic qui va être envoyé à l'une de ses voisines I_2 en additionnant les scores de chacun des mouvements menant à I_2 . Ces scores, calculés pour chacune des $|D|$ directions de l'intersection peuvent être comparés et ordonnés. Nous notons par $r(d)$ le rang de la direction sortante d : la direction qui obtient le plus grand score a un rang de $|D|$, celle d'après de $|D| - 1$, etc. jusqu'à la dernière qui obtient un rang de 1. Ces rangs sont normalisés ($R(d) = r(d)/|D|$) et transmis à chacune des intersections voisines. Une intersection réceptrice (I_2) associe alors à chaque mouvement (s, d) prenant origine à l'intersection émettrice (I_1) un score $R(s, d)$ égal au rang de la direction allant de I_1 à I_2 .

Finalement, pour chaque mouvement y le score global $S(y)$ s'exprime, sur la base des trois objectifs normalisés sur les M mouvements existants, comme suit :

$$S(s, d) = \gamma \cdot \frac{SL(s, d)}{\sum_{\{a,b\} \in D} SL(a, b)} + \delta \cdot \frac{GE(s, d)}{\sum_{\{a,b\} \in D} GE(a, b)} + \eta \cdot \frac{R(s, d)}{\sum_{\{a,b\} \in D} R(a, b)},$$

γ , δ et η sont trois pondérations qui peuvent être utilisées afin de donner une importance supplémentaire à chacun des objectifs. Par exemple, nous pouvons imaginer le cas d'un embouteillage important, où l'objectif lié à la création de vagues vertes deviendrait moins important que celui relatif au délestage. Enfin, notons qu'une intersection isolée ne recevra jamais les informations nécessaires pour les objectifs 2 et 3 et se basera donc uniquement sur le score local, ce qui revient à appliquer l'algorithme précédent sur une simple intersection.

4.3 Algorithme au niveau d'une intersection

Afin de prendre en compte les trois objectifs identifiés précédemment, nous concevons un algorithme, baptisé TAPIOCA (*distribuTed and AdaPtive IntersectiOns Control Algorithm*),

Contrôle des feux de circulation par un réseau de capteurs sans fil

mis en œuvre par le réseau de capteurs décrit en 4.1. Cet algorithme est divisé en intervalles de temps (ou slots) et est composé de 7 étapes :

1. **Comptabilisation.** Durant la phase P , chaque nœud CS collecte les arrivées en enregistrant chacune des signatures électromagnétiques laissées par les véhicules. De la même manière, chaque nœud CD collecte les départs. A la fin de la phase P , les informations sont corrélées afin de pouvoir calculer la longueur des différentes files d'attente. Les étapes décrites ci-dessous sont illustrées au travers d'un exemple sur la figure 4.
 - (a) Le slot $s1$ commence. Soit CD_n^d un nœud CS de niveau n facultativement rattaché à une direction d . Chaque nœud CS envoie les arrivées comptabilisées aux nœuds CD_2^d dont la direction d est une destination possible. Chaque nœud CD_1^d envoie les départs comptabilisés au même moment à son nœud CD_2^d possédant la même direction d .
 - (b) Le slot $s1$ se termine. Si un nœud CD (respectivement CS) n'a pas envoyé de message pendant $s1$, alors aucun nouveau départ (respectivement nouvelle arrivée) n'a été comptabilisé(e) par ce nœud. S'il existe deux nœuds $CD_2^{d^1}$, $CD_2^{d^2}$, avec $d^1 \neq d^2$, possédant une source commune CS , alors chacun envoie à l'autre les départs comptabilisés durant un slot de temps additionnel $s1b$. Cette opération est nécessaire afin de prendre en compte les cas où une voie possède plusieurs mouvements.
 - (c) Pour chaque mouvement y dont la destination est la direction d , chaque nœud CD_2^d possède ainsi les signatures des arrivées notées S_A^y et des départs notés S_D^y . Chacun de ces nœuds peut calculer le nombre de véhicules présents à la source, sur les voies du mouvement y :

$$N_P^y = N_{P-1}^y + |S_A^y| - |S_A^y \cap S_D^y|.$$

Notons que chaque nœud CD_2^d possède et maintient le temps d'attente depuis le dernier feu vert T_F^y , pour chaque mouvement y . Chaque nœud CD_2^d possède également des informations sur l'intersection voisine j liée à sa direction d : le nombre de véhicules présents sur les voies entrantes de j (deuxième objectif), la *priorité normalisée* de la direction de j liée à d (troisième objectif).

2. **Agrégation.** Chaque nœud CD_2^d calcule le nombre de véhicules présents sur les voies entrantes de sa direction d , N^d . Durant le slot de temps $s2$, chaque nœud CD_2^d envoie ses valeurs de N_P^y , T_F^y au nœud maître CD_3 .
3. **Définition de la prochaine phase.**
 - (a) **Sélection des mouvements** composant la prochaine phase. Comme décrit en 4.2, le nœud sélectionne les phases les plus adaptées, en calculant tout d'abord trois objectifs. Premièrement, le score de chaque mouvement (score "local"). Deuxièmement, le gain espéré pour chaque intersection voisine. Enfin, les priorités des intersections voisines qui sont déjà connues et normalisées. Sur la base de ces trois paramètres, chacun des mouvements est évalué et la meilleure combinaison respectant la matrice des conflits est sélectionnée.

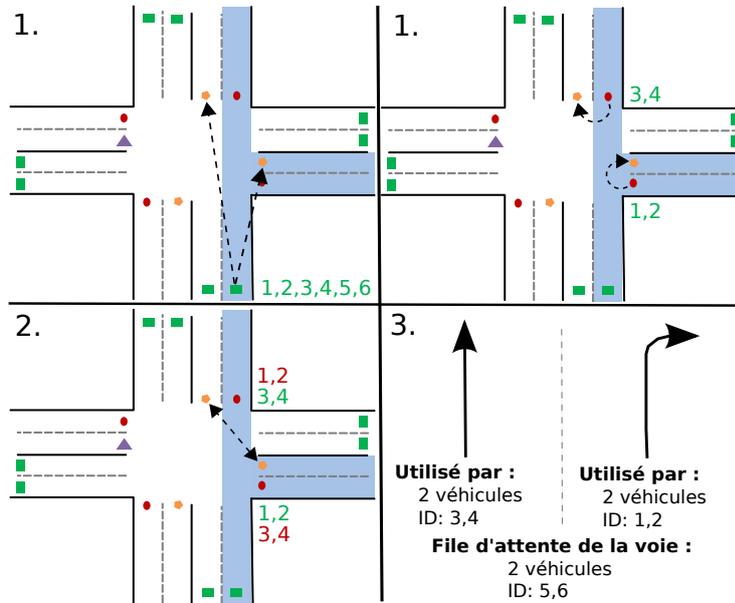


FIG. 4 – Echanges nécessaires afin de réaliser le comptage des véhicules, ici pour le mouvement sud-nord et sud-est.

- (b) **Calcul du Temps de feu.** Borné à T_{max} , ce dernier doit laisser passer un maximum de véhicules de l'intersection et est estimé à $T_P = T_s + N_{max} * T_H$ secondes. N_{max} est le nombre de véhicules pour la voie en ayant le plus grand nombre.
- Influence externe.** Certaines valeurs doivent être transmises aux intersections voisines. Tout d'abord, Le nœud CD_3 se charge de calculer la priorité normalisée pour chacune des directions d de son intersection, puis transmet, pendant le slot de temps s_3 , le résultat à chacun des nœuds CD_2^d ayant $N^d \neq 0$, en plus de la phase qui a été sélectionnée. Ensuite, chaque nœud CD_2^d met à jour T_F^y : si $N^d > 0$ et que le mouvement y est dans la phase sélectionnée, alors $T_F^y = 0$, sinon $T_F^y = T_F^y + T_{P-1}$. Enfin, durant le slot de temps s_4 , chaque nœud CD_2^d ayant $N^d \neq 0$ transmet à l'intersection voisine de sa direction d sa priorité normalisée ainsi que N^d , afin que puissent être calculés le deuxième et troisième objectif.
 - Mise en place de la phase.** Le nœud CD_3 envoie la décision à l'interface reliée au contrôleur durant le slot de temps s_4 , en simple ou multi-sauts s'il est rattaché à l'intersection ou à un ensemble d'intersections, la phase $P+1$ débute.
 - Durant la phase $P+1$,** dès lors qu'un nœud CD_2^d de l'intersection i enregistre pour la première fois un véhicule, il envoie un message à l'intersection voisine, contenant l'identifiant de ce véhicule, le nombre de véhicules qui vont arriver, ainsi qu'un horodatage $T_{timestamp^1}$. Lorsque l'intersection voisine j enregistre l'arrivée du premier véhicule sur l'un des nœuds CS , un deuxième horodatage $T_{timestamp^2}$ est enregistré. La valeur $T_{i,j} = T_{timestamp^2} - T_{timestamp^1}$ correspond au temps de traversée entre l'intersection

i et j , cette dernière servant à ajuster une moyenne du temps de traversée, calculée au cours de la vie du réseau, $T_{i,j}^{AVG}$. Dans le cas où $T_P + T_{i,j}^{AVG} < T_{max}$, l'intersection voisine est en mesure de prolonger la durée du feu vert et faire passer en conséquence un maximum de véhicules. Des contraintes peuvent être ajoutées à cette condition, comme par exemple un nombre minimal de véhicules à attendre.

4.4 Estimation des coûts de communication

Si nous regardons uniquement les coûts sur une intersection, coupée de l'extérieur, un protocole de communication naïf dans lequel chaque capteur ne fait que reporter les arrivées et les départs à un nœud central engendre environ $|arrivées\ totales| + |départs\ totaux|$ messages. Notre solution génère un trafic réseau, par phase, situé entre 1 message (aucun véhicule sur l'intersection) et $|noeuds| + 2 * |directions| + \Delta_{s1b} - 1$ messages (une arrivée comptabilisée pour chaque nœud CS et un départ comptabilisé pour chaque destination possible, sur chaque nœud CD , Δ_{s1b} représentant les communications engendrées par le slot s1b). Les coûts supplémentaires liés à la communication avec les intersections voisines sont faibles : il est nécessaire d'envoyer 2 messages par intersection voisine, en simple ou multi-sauts selon la distance. L'élection des nœuds peut être réalisée en $O(\log(\alpha))$ messages (Jurdziński et al., 2002), où α est le nombre de nœuds parmi lesquels réaliser l'élection.

4.5 Comparaisons

Le tableau 2 ci-dessous compare le modèle utilisé par notre solution à ceux utilisés dans la littérature (étudiée en section 2). Nous pouvons constater que notre modèle utilise une architecture répartissant équitablement un capteur par voie et propose une logique hiérarchique distribuée, tandis que la littérature se contente d'utiliser les capteurs comme simple points de recueil afin de centraliser la gestion des données et des calculs. De plus, notre architecture, de même que notre algorithme, ne supposent pas une infrastructure limitée ou des véhicules de même type et vitesse (Zou et al., 2009; Zhou et al., 2010) et permettent une prise de décision rapide à chaque phase.

Auteurs	Gestion	Décisions	Nombre de capteurs	Remarques
Yousef et al. (2010)	Centralisée	Par cycle	2 par voie entrante	-
Tubaishat et al. (2008)	Centralisée	Par cycle	2 par voir entrante	Hypothèses lourdes
Zhou et al. (2010)	Centralisée	Par cycle	1 par direction	Infrastructure limitée
Zou et al. (2009)	Centralisée	Par cycle	-	-
Faye et al. (2012a)	Distribuée	Par phase	1 par voie entrante / sortante	-

TAB. 2 – Comparaison de notre modèle avec la littérature.

Le tableau 3 compare notre algorithme à ceux adoptés par les mêmes auteurs de la littérature. Nous constatons que notre algorithme calcule les séquences de feux verts et leur durée entièrement en fonction des variations de trafic et ne se base pas, contrairement à certaines contributions, sur des données statiques, ce qui réduit potentiellement les possibilités de déploiement à n'importe quel type d'intersection. Notre algorithme prend en compte le cas d'intersections voisines, afin de synchroniser les feux de circulation dans l'optique de créer des

vagues vertes. Enfin, plusieurs objectifs sont pris en compte, et non seulement la taille des files d'attente, afin d'empêcher, par exemple, les cas de famine.

Auteurs	Paramètres	Séquence de feux	Temps de feux	Multi-intersections
Yousef et al. (2010)	Queue	Adaptatif	Adaptatif	Yousef et al. (2010)
Tubaishat et al. (2008)	Queue	Semi-adaptatif	-	-
Zhou et al. (2010)	Multiple	Semi-adaptatif	Adaptatif	Zhou et al. (2011)
Zou et al. (2009)	Queue	-	Logique floue	-
Faye et al. (2012a)	Multiple	Adaptatif	Adaptatif	<i>Ici</i>

TAB. 3 – Comparaison de notre algorithme avec la littérature.

5 Simulations

Nous évaluons l'algorithme TAPIOCA en utilisant le simulateur de réseau routier SUMO (Behrisch et al., 2011), au travers de deux scénarios.

5.1 Scénario réel, ville d'Amiens

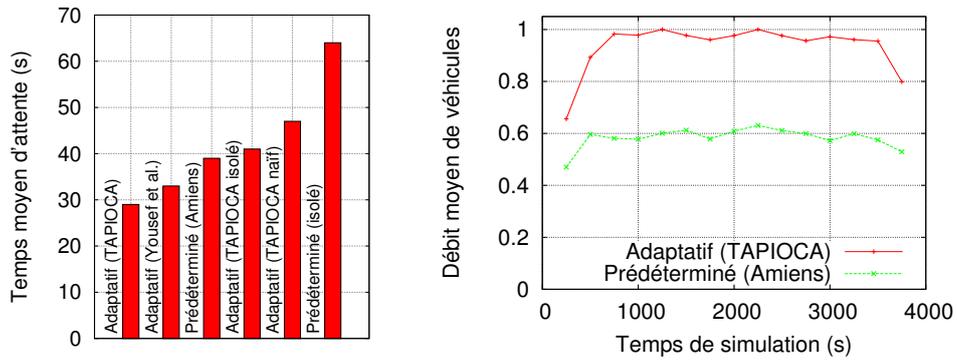
Dans un premier temps, nous nous reposons sur cinq intersections voisines issues de la ville d'Amiens (Somme, France), sur lesquelles nous possédons des statistiques de comptage et les plans de feux en place à une heure de pointe, entre 8 h et 9 h du matin. La carte simulée consiste en une intersection centrale encadrée par quatre autres (au nord, est, ouest et sud).

Sur la figure 5(a), nous comparons le temps moyen d'attente cumulé sur ces intersections, pour une charge λ et une répartition de véhicules conformes aux données possédées. Nous comparons TAPIOCA avec cinq méthodes différentes. Tout d'abord, l'algorithme adaptatif proposé par Yousef *et al.*. Ensuite, les plans de feux prédéterminés utilisés sur Amiens pour ces cinq intersections, conçus pour répondre à la demande simulée et synchronisés afin de créer des vagues vertes. Ensuite, l'algorithme TAPIOCA uniquement local (isolé), c'est-à-dire sans lien entre intersections voisines (objectifs 2 et 3 supprimés). Puis, l'algorithme TAPIOCA dans une version naïve, ne prenant en compte que la longueur des files comme critère de décision. Enfin, nous isolons la gestion des deux intersections en proposant deux plans de feux prédéterminés, sélectionnés de manière indépendante pour leurs performances, parmi 5 plans de feux équitables et quelconques. Les résultats sont indiqués sur la figure 5.

Le temps moyen d'attente global général par l'algorithme TAPIOCA est reporté sur la figure 5(a) et est de 12 % plus court que Yousef *et al.*, de 26 % comparé aux plans de feux Amiénois et reste moins important qu'un algorithme similaire mais dont les intersections ne sont pas synchronisées. La figure 5(b) montre que TAPIOCA est capable de traiter davantage de véhicules qu'une méthode prédéterminée.

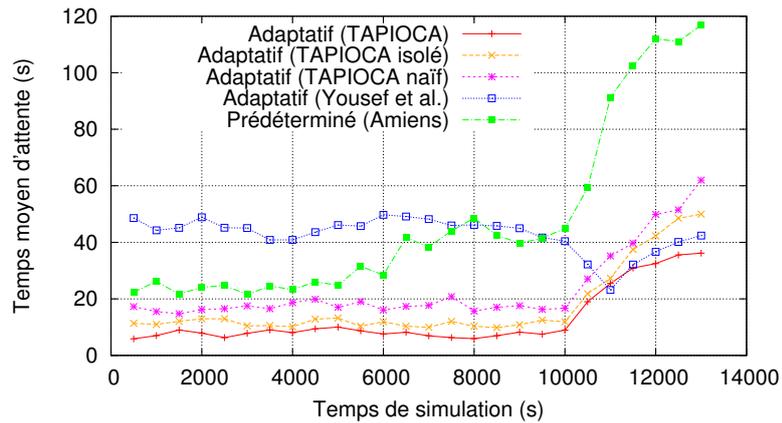
Dans les deux précédents cas, chaque simulation est lancée durant 3 600 s et estime le temps moyen d'attente de 3 000 véhicules entrant sur le réseau depuis chacune des intersections extérieures, en suivant un processus de Poisson λ d'intensité constante. La figure 5(c) représente l'évolution du temps moyen d'attente des véhicules empruntant un chemin dont le trafic triple à 10 000 s de simulation, sur un total de 13 000 s. Il est montré que TAPIOCA offre une meilleure réactivité qu'une méthode prédéterminée, qui est conçue pour une charge

Contrôle des feux de circulation par un réseau de capteurs sans fil

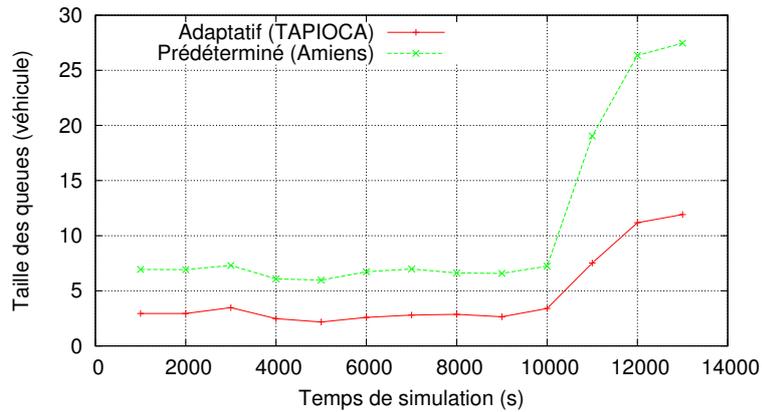


(a) Temps moyen d'attente global.

(b) Débit moyen de véhicules.



(c) Temps moyen d'attente face à un changement soudain de trafic.



(d) Taille moyenne des files d'attente.

FIG. 5 – Comparaison de TAPIOCA avec d'autres stratégies - Amiens.

moyenne particulière. Le temps moyen d'attente généré par l'algorithme de Yousef *et al.* diminue, ceci étant dû aux paramètres, dont la configuration est faite pour accueillir une charge plus importante.

5.2 Scénario TAPASCologne

Dans un second temps, nous nous reposons sur le scénario *TAPASCologne (0.0.3)* (Behrisch et al., 2011), représentant le trafic de la ville de Cologne (Allemagne) pour deux heures. Ce scénario volumineux est l'un des plus grands - si ce n'est le plus grand - jeu de données de simulation de trafic existant actuellement. Le réseau routier et les données de trafic sont de bonne qualité et ont été générés par l'équipe de SUMO sur la base d'informations statistiques et cartographiques. Nous noterons que le scénario est toujours en cours de développement, certains aspects s'éloignent hélas de la réalité : les plans de feux en place sont, par exemple, générés par SUMO. Ce scénario constitue tout de même une bonne base d'évaluation de nos algorithmes.

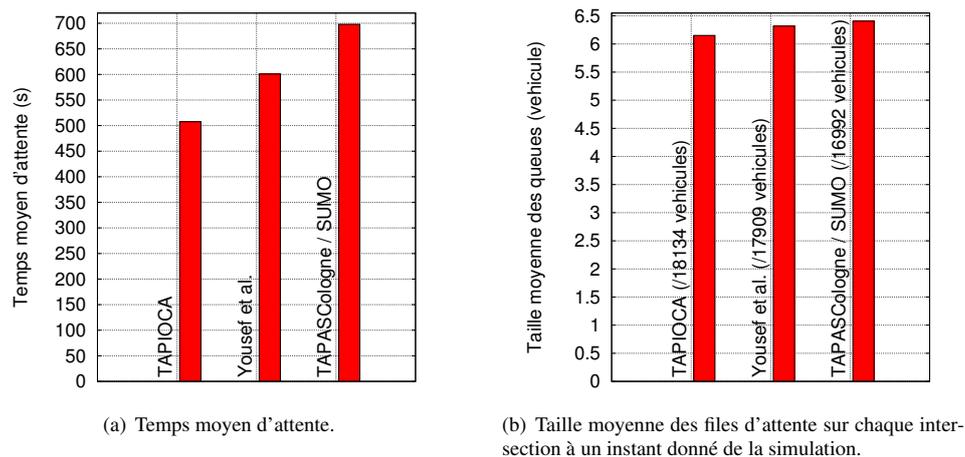


FIG. 6 – Comparaison de TAPIOCA avec d'autres stratégies - TAPASCologne.

Les résultats présentés sur la figure 6 montrent que TAPIOCA améliore le temps d'attente des véhicules parcourant ce réseau ainsi que la taille moyenne des queues par rapport à deux méthodes : la première utilise les plans de feux pré-déterminés par SUMO, la seconde est inspirée de l'algorithme adaptatif de Yousef et al. (2010). Ce dernier algorithme, ainsi que TAPIOCA, ont été adaptés aux besoins du scénario. Ainsi, pour chaque plan de feux à rendre adaptatif :

- les deux algorithmes se basent sur une matrice des conflits générée à partir des phases statiques définies,
- TAPIOCA ajuste automatiquement le temps maximum de feu vert et les pondérations des objectifs à partir des données statiques issues du plan de feux.

La figure 6(a) montre que le temps moyen d'attente généré par TAPIOCA est de 27 % plus court que le temps d'attente généré par les plans de feux en place sur le scénario et de 15 %

plus court que celui généré par l'algorithme adaptatif issu de Yousef et al. (2010). La figure 6(b) montre que notre algorithme permet de faire entrer davantage de véhicules dans le réseau que les méthodes concurrentes, jusqu'à 1 142 véhicules supplémentaires en comparaison à la méthode initiale. Malgré cette différence de véhicules dans le réseau, la taille moyenne des files d'attente reste plus faible, de 4 % que la méthode initiale et de 2,6 % que l'algorithme issu de Yousef et al. (2010). Des résultats supplémentaires sont disponibles en ligne et seront complétés au fil de nos études ¹.

6 Conclusion

L'algorithme de contrôle des feux de circulation proposé dans cet article combine, pour décider des mouvements qui composeront la prochaine phase et de la durée de cette phase, des objectifs locaux de réduction de la longueur des files d'attente et de prévention des situations de famine, avec des objectifs plus généraux, favorisant le délestage ou cherchant à fluidifier le trafic en créant des vagues vertes.

Les résultats de simulation obtenus sur un modèle de trafic réel issu de la ville d'Amiens montrent le potentiel d'un tel algorithme dynamique de contrôle des feux de circulation en termes de réduction du temps d'attente moyen. L'algorithme présenté ici permet de s'adapter aux variations instantanées du trafic plus efficacement que des stratégies locales ou qu'un plan de feux pré-calculé. Les résultats de simulation sur un modèle de trafic à grande échelle issu de TAPASCologne nous confortent également dans cette idée.

L'estimation rapide du coût de communication réalisée dans ce travail montre qu'il est tout à fait possible, en termes de débit, de déployer une telle stratégie en utilisant des solutions sans-fil standard. Toutefois, nous n'avons pas pris en compte les phénomènes d'obstructions, de collisions ou d'interférences et nous travaillons actuellement au développement de notre algorithme dans un co-simulateur composé de SUMO et d'OMNeT++ pour la partie communications.

En outre, si l'algorithme présenté se borne pour l'instant à réduire les temps d'attente aux différentes intersections, il est parfaitement adaptable pour intégrer, par exemple, des objectifs liés à la sécurité des usagers ou à une application incitative des règles de circulation. Si un flux de véhicule roule systématiquement à une vitesse moyenne trop élevée sur le même chemin, il pourra être détecté et des malus pourraient être ajoutés au calcul du score. Enfin, nous pouvons imaginer des extensions telles que la détection de véhicules d'urgence et plus globalement l'adaptation de l'algorithme – qui est ici très général – aux normes en vigueur dans les différents pays.

Remerciements

Les auteurs souhaitent remercier Thierry Delaporte, responsable du poste de régulation du trafic d'Amiens, qui nous a fourni les données nécessaires à nos simulations.

1. <http://perso.telecom-paristech.fr/~faye/TAPASCologne/>

Références

- Behrisch, M., L. Bieker, J. Erdmann, et D. Krajzewicz (2011). Sumo - simulation of urban mobility : An overview. In *The Third International Conference on Advances in System Simulation (SIMUL 2011)*, Barcelona, Spain, pp. 63–68.
- Cheung, S., S. Coleri, B. Dundar, S. Ganesh, C. Tan, et P. Varaiya (2005). Traffic measurement and vehicle classification with single magnetic sensor. *Transportation Research Record : Journal of the Transportation Research Board 1917(-1)*, 173–181.
- Collotta, M., G. Pau, V. Salerno, et G. Scatá (2012). Wireless sensor networks to improve road monitoring.
- Corredor, I., A. García, J. Martínez, et P. López (2008). Wireless sensor network-based system for measuring and monitoring road traffic. In *6th Collaborative Electronic Communications and eCommerce Technology and Research (COLLECTeR 2008)*, Madrid, Spain.
- Faye, S., C. Chaudet, et I. Demeure (2012a). A distributed algorithm for adaptive traffic lights control. In *15th IEEE Intelligent Transportation Systems Conference (ITSC 2012)*, Anchorage, USA.
- Faye, S., C. Chaudet, et I. Demeure (2012b). Un algorithme distribué de contrôle des feux de circulation sur plusieurs intersections par un réseau de capteurs sans fil. In *Conférence Internationale sur les NOuvelles TEchnologies de la RÉpartition - Colloque Francophone sur l'Ingénierie des Protocoles (NOTERE-CFIP)*, Anglet, France.
- Gordon, R., W. Tighe, U. S. F. H. A. O. of Operations, D. E. Associates, et I. Siemens (2005). *Traffic control systems handbook*. US Dept. of Transportation, Federal Highway Administration, Office of Operations. <http://ops.fhwa.dot.gov/publications/fhwahop06006/>.
- Jurdziński, T., M. Kutylowski, et J. Zatopiański (2002). Efficient algorithms for leader election in radio networks. In *21th annual symposium on Principles of distributed computing*, pp. 51–57. ACM.
- Knaian, A. N. (2000). A wireless sensor network for smart roadbeds and intelligent transportation systems. Master's thesis, Massachusetts Institute of Technology.
- Tubaishat, M., Q. Qi, Y. Shang, et H. Shi (2008). Wireless sensor-based traffic light control. In *5th IEEE Conference on Consumer Communications and Networking (CCNC 2008)*, Las Vegas, USA.
- Wiering, M., J. Vreeken, J. van Veenen, et A. Koopman (2004). Simulation and optimization of traffic in a city. In *Intelligent Vehicles Symposium, 2004 IEEE*, pp. 453 – 458.
- Yousef, K. M., J. N. Al-Karaki, et A. M. Shatnawi (2010). Intelligent traffic light flow control system using wireless sensors networks. *Journal of Information Science and Engineering* 26(3).
- Zhou, B., J. Cao, et H. Wu (2011). Adaptive traffic light control of multiple intersections in wsn-based its. In *73rd IEEE Vehicular Technology Conference (VTC Spring)*, pp. 1–5. IEEE.
- Zhou, B., J. Cao, X. Zeng, et H. Wu (2010). Adaptive traffic light control in wireless sensor network-based intelligent transportation system. In *72nd IEEE Vehicular Technology Conference Fall (VTC 2010-Fall)*, Ottawa, Canada.
- Zou, F., B. Yang, et Y. Cao (2009). Traffic light control for a single intersection based on

Contrôle des feux de circulation par un réseau de capteurs sans fil

wireless sensor network. In *9th International Conference on Electronic Measurement & Instruments (ICEMI 2009)*, Beijing, China.

Summary

In this article, we detail and evaluate a distributed algorithm that defines the green lights sequence and duration in a multi-intersection intelligent transportation system (ITS). We expose the architecture of a wireless network of sensors deployed at intersections, which takes local decisions without the help of a central entity. We define an adaptive algorithm that uses data collected by this sensor network to decide dynamically of the green light sequences. Simulation results performed show that our algorithm achieves a low average waiting time and reacts quickly to traffic load increases, compared to other dynamic strategies and to pre-determined schedules.