

Croisement synchronisé de flux de véhicules autonomes dans un réseau

Mohamed Tlig, Olivier Buffet, Olivier Simonin

► **To cite this version:**

Mohamed Tlig, Olivier Buffet, Olivier Simonin. Croisement synchronisé de flux de véhicules autonomes dans un réseau. RJCIA - 11èmes Rencontres des Jeunes Chercheurs en Intelligence Artificielle, Jul 2013, Lille, France. 2013, Actes des 11èmes Rencontres des Jeunes Chercheurs en Intelligence Artificielle. <hal-00914578>

HAL Id: hal-00914578

<https://hal.inria.fr/hal-00914578>

Submitted on 5 Dec 2013

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

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

Croisement synchronisé de flux de véhicules autonomes dans un réseau

Mohamed Tlig^{a,b}, Olivier Buffet^{a,b}, Olivier Simonin^{b,a}

^a INRIA équipe Maia, LORIA
^b Université de Lorraine, France
prenom.nom@loria.fr

Résumé :

Les véhicules autonomes sont aujourd'hui considérés comme une approche prometteuse pour le transport des ressources et la régulation du trafic. Dans cet article, nous examinons la possibilité de faire croiser des flux de véhicules autonomes sans les arrêter afin d'éviter les congestions (retards) et la perte d'énergie. Nous proposons un contrôle aux intersections basé sur la synchronisation temporelle des véhicules. Nous décentralisons le problème en gérant indépendamment chaque intersection. Nous définissons un agent de contrôle qui est capable de synchroniser les véhicules arrivant sur une intersection, en assurant l'alternance entre les flux. Nous présentons un simulateur qui permet d'évaluer l'approche et de la comparer avec la stratégie standard des feux de circulation. Les résultats expérimentaux montrent un gain important en termes de temps et d'énergie pour les véhicules à une intersection et dans un réseau.

Mots-clés : Système Multi-Agent, Synchronisation de flux de véhicules, Véhicules autonomes, Simulation de trafic.

1 Introduction

Dans les systèmes multi-agents réels envisagés pour le transport automatique, comme dans les grands ports maritimes¹, les véhicules autonomes sont confrontés à des problèmes de conflit de passage à chaque intersection. Afin d'améliorer leurs temps de parcours, plusieurs solutions existent. Les plus simples favorisent généralement un flux par rapport à un autre comme les feux de circulation et les panneaux "STOP". De tels événements sont générateurs de retard pour les agents car ils obligent plusieurs

1. Voir, e.g., le projet européen InTraDE <http://www.intrade-nwe.eu/>

véhicules à s'arrêter pendant un certain temps. Si les flux de véhicules sont importants, ces retards locaux peuvent mener à l'émergence d'embouteillages.

Nous nous intéressons, dans ce papier, au problème de la gestion automatique du croisement de flux de véhicules autonomes à des intersections (Dresner & Stone (2004)), c'est-à-dire lorsqu'ils doivent emprunter le même lieu au même moment. Une façon triviale de résoudre le problème est de stopper tous les flux pour n'en laisser passer qu'un seul (comme la gestion actuelle des intersections dans les villes). Or il a été montré que cette stratégie est coûteuse en termes de temps (Fok *et al.* (2012)), puisqu'elle oblige l'arrêt systématique des flux.

Une autre façon d'aborder ce problème est de laisser se croiser les flux sans arrêter les véhicules en s'assurant de suffisamment les espacer et de synchroniser leur passage. C'est cette idée que nous développons dans ce papier. Il faut pour cela examiner les paramètres spatiaux d'une intersection (taille, l'angle entre les routes) qui influencent la synchronisation temporelle. A partir de cette étude, nous définissons les agents de contrôle qui déterminent la vitesse des véhicules qui arrivent à une intersection.

La section 2 présente le problème des intersections en général, et les travaux existants. En section 3 nous présentons le principe de la synchronisation temporelle de deux routes à une intersection, nous expliquons (i) comment calculer la période minimale entre deux véhicules qui se suivent et (ii) l'algorithme utilisé par l'agent de contrôle. La section 4 est consacrée à l'étude expérimentale en simulation de notre modèle comparé à une solution existante. Enfin, nous discutons ces résultats et la poursuite de ce travail dans la conclusion.

2 Travaux existants

Nous nous concentrons, dans ce papier, sur des agents autonomes ayant pour mission de transporter des ressources sur un réseau de routes. Nous traitons le problème général de la gestion des croisements des flux. Ces problèmes de gestion des croisements sont traditionnellement étudiés en recherche opérationnelle et en théorie des files d'attente. Ils concernent des véhicules conduits par des humains. Mais avec l'arrivée des nouvelles technologies, des travaux intégrant des décisions en ligne permettent d'envisager un contrôle automatique et temps réel. Plusieurs approches basées sur les communications et le GPS proposent d'améliorer des solutions existantes telles que les feux de circulation. Dans Bhourri *et al.* (2011) les

auteurs proposent une stratégie qui vise à améliorer les temps de parcours des transports en commun. Le système est basé sur les réservations des phases de feux vert à chaque intersection et donne la priorité aux bus qui sont le plus en retard. Dans le même contexte, les auteurs de Hounsell & Shrestha (2012) proposent une stratégie qui donne une priorité aux bus calculée à partir de son progrès (lors de son trajet) et aussi sur le progrès du bus qui le succède.

D'autres solutions s'intéressent au contrôle de véhicules totalement autonomes. On peut les classer généralement en deux catégories. Les méthodes par réservation, introduites par Dresner et Stone dans Dresner & Stone (2004, 2005), sont fondées sur un agent qui gère une intersection. Chaque véhicule voulant passer doit réserver un temps de passage et une route à prendre. L'avantage de cette approche est que si on a plusieurs véhicules qui veulent passer et que leurs trajectoires ne se croisent pas, on peut satisfaire tout le monde. Mais dans le cas contraire, on est obligé de donner la priorité à un véhicule par rapport aux autres.

La méthode décentralisée introduite par Rasche et Naumann dans Naumann & Rasche (1997); Naumann *et al.* (1997) repose essentiellement sur la communication et la négociation entre les véhicules afin de déterminer l'ordre de passage et de sortie de l'intersection. Cette approche est connue pour ses limites qui dépendent du nombre de véhicules voulant négocier leurs passages par l'intersection et l'évolution du domaine des communications sans fil.

3 Synchronisation d'un croisement

3.1 Approche par synchronisation locale

Nous introduisons dans cette section notre approche. Avant de décrire son fonctionnement, nous présentons le type de réseau de route sur lequel nous travaillons. Nous considérons des routes avec une ou deux voies, c'est à dire un ou deux flux opposés. Les intersections (représentées en figure 1) permettent de croiser une route mais pas d'en changer. Nous traitons des réseaux routiers (qu'on appellera aussi *graphe*) composés de croisements de routes à une ou deux voies. Les routes peuvent se croiser avec un angle quelconque (nous considérons d'abord le cas quand $\theta = 90^\circ$ puis nous généralisons).

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

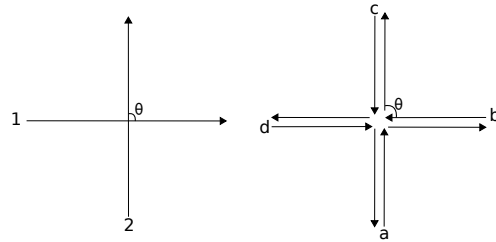


FIGURE 1 – Croisements à 2 et 4 flux

distance suffisante afin d'éviter les collisions. La régulation est donc locale et temporelle.

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

L'objectif de l'agent de contrôle est de réguler les vitesses des véhicules à l'intérieur de son rayon d'action. Nous détaillons dans ce qui suit comment déterminer la période d'alternance entre les routes et comment l'agent, en s'appuyant sur cette période, contrôle les véhicules qui s'approchent de son intersection.

3.2 Période minimale de croisement

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

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

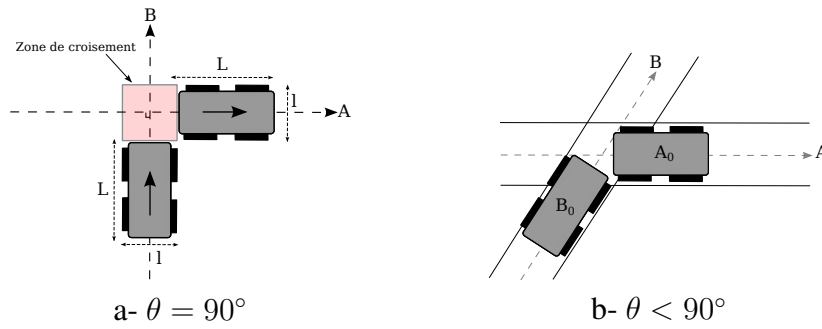


FIGURE 2 – Zone de croisement de deux flux

3.2.1 Cas $\theta = 90^\circ$

La zone de croisement est définie par le rectangle correspondant à l'espace partagé par les deux routes (voir figure 2-a). Pour qu'un véhicule traverse complètement cette zone il lui faut un temps $(l + L)/V$. On en déduit la période minimale de passage des deux véhicules de chaque flux :

$$T = 2(L + l)/V. \quad (1)$$

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

3.2.2 Cas $\theta < 90^\circ$

Dans ce cas de figure, un véhicule peut entrer dans la zone de croisement avant que l'autre ne soit complètement sorti (voir figure 2-b). La zone de croisement doit être partagée par les deux véhicules de chaque file. On en déduit la période de passage minimale des deux flux (voir calculs en annexe A.1) :

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

3.2.3 Cas $\theta > 90^\circ$

Dans ce cas de figure, il n'y a pas de partage de la zone de croisement. La période de passage minimale est (voir calculs en annexe A.2) :

$$T = 2 \left(l \frac{(1 + \cos \theta)}{\sin \theta} + L \right) / V. \quad (3)$$

En pratique T sera augmentée de ϵ une marge de sécurité d'inter-distance des véhicules.

3.3 Synchronisation temporelle

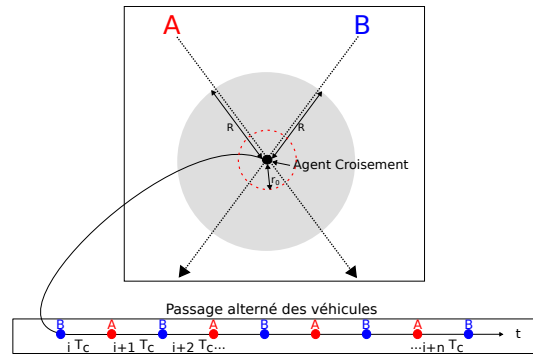


FIGURE 3 – Principe de synchronisation d'une intersection

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

La figure 3 illustre ce principe de passage alterné de période T_c . L'agent de contrôle est chargé de calculer les consignes de vitesse des véhicules quand ils sont dans son rayon d'action, c'est à dire à une distance comprise entre r_0 et R . r_0 étant la distance à partir de laquelle le véhicule sera trop près du croisement pour recevoir une modification de vitesse ($0 < r_0 < R$).

Chaque flux peut être contrôlé indépendamment des autres. L'algorithme 1 donne le contrôle d'un flux par l'agent connaissant sa parité $p \in [0, 1]$. L'algorithme est exécuté à fréquence régulière (tous les pas de temps dans la simulation). Il traite toujours les véhicules du plus proche de l'intersection au plus éloigné pour assurer une affectation respectant le même ordre de passage (ligne 2). L'agent de contrôle a besoin de percevoir leur distance à l'intersection (r_i) et leur vitesse (V_i).

Considérons un seul véhicule i , le calcul de l'instant de passage par le centre de l'intersection est donné par la ligne 5 (en supposant sa vitesse V_i). Nous faisons le choix de n'autoriser que des ralentissements, alors le véhicule devra passer par l'intersection au moment $(2n_i - p)T$, où n_i est

Algorithme 1 : Contrôle d'un flux par l'agent de contrôle

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

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

un entier calculé comme dans la ligne 8. L'agent de contrôle doit veiller à ce que cette valeur n'a pas été déjà attribué à un autre véhicule. À cette fin, la variable $last$ enregistre la dernière valeur attribuée à un véhicule et un test est effectué (ligne 9) pour déterminer s'il faut utiliser la valeur calculée ou $last + 1$.

L'agent calcule n_i une seule fois à l'entrée du véhicule dans son rayon d'action. Dans le cas contraire, à cause des variations de vitesse, la valeur n_i pourrait changer d'une itération à une autre. Ceci est obtenu en affectant la valeur $n_i = -1$ aux véhicules en dehors des rayons d'action des agents. L'agent de contrôle stocke n_i d'une itération à une autre et ne calcule n_i que si et seulement si $n_i = -1$ (ligne 6).

Une fois n_i connu, l'agent calcule la date de passage t_p du véhicule i à la distance r_0 du centre d'intersection, puis en conséquence, la commande de vitesse v à appliquer jusqu'à $r_i = r_0$ (par la suite il a une vitesse constante V).

3.4 Routes à deux voies opposées

Dans cette section, nous allons brièvement montrer comment adapter la solution présentée dans les sections précédentes au cas de deux routes composées chacune de deux voies opposées, nous tiendrons compte dans nos expériences du cas présenté sur la figure 1 (à droite). La période de passage minimale T pour la traversée doit être adaptée à ce cas précis,

pour tenir compte de la taille de la nouvelle zone croisement. Soit λ la distance qui sépare deux voies opposées de la même route. La période de passage minimale est (voir détails en annexe A.3) :

$$T = 2 \left(\frac{2(l + \frac{\lambda}{2})}{\sin \theta} + L \right) / V. \quad (4)$$

L'algorithme de l'agent de contrôle reste inchangé, chaque voie étant traitée indépendamment des autres voies, chaque voie ayant sa propre variable *last*. Il faut juste faire attention au fait que, pour une voie donnée, les distances ne sont plus calculées par rapport au centre de la zone de croisement, mais par rapport au milieu du segment de la voie à l'intérieur de la zone de croisement, comme illustré sur la figure 4.

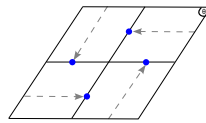


FIGURE 4 – Zone de croisement : 2 flux opposés sur chaque route

4 Expérimentations

Dans cette section nous évaluons notre approche en plaçant un agent de contrôle sur une intersection de deux routes avec deux flux opposés comme le montre la figure 5-a. Nous comparons notre approche à une stratégie basée sur les feux de circulation à durée fixe. Un feu de circulation est placé à l'entrée des croisements pour chaque flux. Les feux permettent d'alterner le passage des véhicules entre deux routes, pendant des périodes fixes et égales.

4.1 Simulations

Nous avons développé (en java) un simulateur à espace continu et à temps discret d'un réseau de routes (figure 5-a est une capture d'écran).

Les paramètres de simulation utilisés sont :

- chaque route est de longueur $1000m$;
- l'angle de croisement est $\frac{\pi}{3}$;
- le rayon d'action R de l'agent de contrôle est de $200m$ et $r_0 = 30m$;

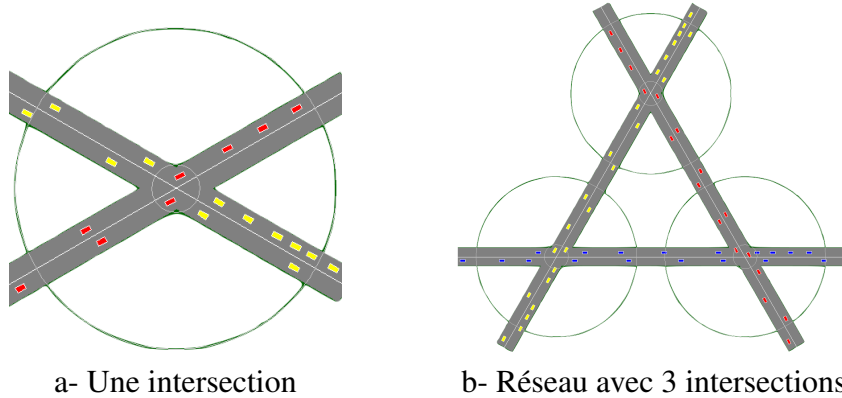


FIGURE 5 – Illustration du simulateur

- la vitesse maximale des véhicules est de $10m/s$, l'accélération maximale $3m/s$ et la décélération maximale $-10m/s$;
- nous avons utilisé un contrôle longitudinal développé dans Scheuer *et al.* (2009) qui assure la non-collision entre les véhicules d'un même flux uniquement en dehors de la zone de croisement ;
- à chaque entrée du réseau, nous avons installé une source qui génère des véhicules ;
- les véhicules ont une dimension $L = 12.5m$, $l = 6.25m$ et la marge de sécurité d'inter-distance $\epsilon = 4m$;
- Le pas de simulation est de $0.5s$.

Chaque source injecte des véhicules suivant une loi de Bernoulli de paramètre $\lambda = \frac{1}{D}$ où D est la durée moyenne, en secondes, entre deux injections consécutives.

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

4.2 Comparaison des stratégies

La figure 6 présente les résultats de simulation. Nous avons calculé la moyenne et l'écart type des retards des véhicules ayant traversés le réseau, pour chaque approche, avec le paramètre $D = 10s$. Le retard est la différence entre un temps de passage théorique et le temps de passage effectué. Nous avons simulé l'approche par feux de circulation avec différentes

durées (8s, 10s, 30s) sachant que 8s est théoriquement le temps minimum pour qu'un véhicule quitte la zone de croisement.

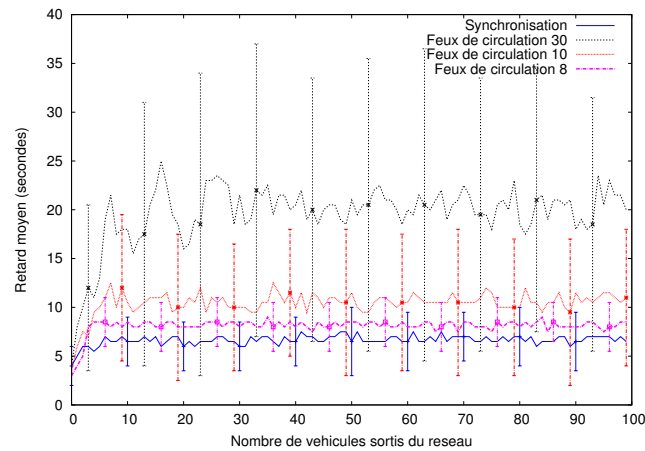


FIGURE 6 – Synchronisation vs. Feux de circulation

Les courbes tracées sont des moyennes sur 100 simulations. Sur l'axe des abscisses nous avons le nombre de véhicules ayant quitté le réseau et sur l'axe des ordonnées le retard moyen des véhicules en secondes. Nous arrêtons les simulations dès que 100 véhicules sont sortis mais nous n'arrêtons pas les injections de véhicules. Nous avons utilisé une fréquence d'injection élevée en chaque source ($D = 10s$) donnant un trafic dense.

On remarque que notre stratégie est très performante sur cette expérimentation par rapport aux feux de circulation (spécialement quand on la compare à la courbe utilisant une durée de feux 30s). Sur 100 simulations, le pire retard réalisé par un véhicule avec l'approche par synchronisation ne dépasse pas les 19s, alors que pour l'approche par feux de circulation il dépasse les 35s (valeur max des écart-types utilisant une durée de feux 30s). La moyenne générale des retards est de l'ordre de 6s en synchronisation locale alors que pour la meilleure courbe des feux de circulation est de l'ordre de 8s (utilisant une durée de feux de 8s).

4.3 Effet de variation du paramètre R

Dans cette section, nous avons mesuré l'effet de la variation du rayon d'action R de l'agent de contrôle afin d'évaluer si il influe sur la qualité de notre solution (sans modifier r_0). Le tableau 1 donne les moyennes et les écarts-types de la consommation d'énergie lorsque $R = 50m$, $R =$

100m et $R = 200m$. Cette énergie consommée est mesurée par la variation totale de la vitesse pendant la traversée de 1000 véhicules. Nous observons que les retards ne changent pas de manière significative, mais il y a une différence en termes de consommation d'énergie lorsque nous réduisons le rayon. Elle augmente quand on utilise un petit R . Cela est dû à la faible distance qui oblige l'agent de contrôle à ralentir brutalement les véhicules.

TABLE 1 – Moyennes et Écart-types de l'énergie consommée

$D = 10$	50m	100m	200m
Moyennes d'énergie	14, 11	9, 84	7, 92
Écart-types	$\pm 5, 18$	$\pm 1, 29$	$\pm 0, 88$

4.4 Effet de variation du paramètre D

Nous voulons étudier, ici, si notre approche est efficace lorsque le débit de véhicules est plus dense (généralement les approches basées sur les réservations, comme dans Dresner & Stone (2004), ont une faible résistance). La figure 7 montre les variations des temps d'attente selon les débits d'injection. Les histogrammes tracés sont des moyennes de 1000 véhicules. Nous avons utilisé les valeurs de D 5s, 10s, 15s et 20s qui correspondent respectivement à des débits 12, 10, 4 et 3 (véhicules / minute). Selon la figure, le délai moyen augmente légèrement lorsque le débit devient plus fort.

4.5 Réseau avec plusieurs intersections

Nous avons testé notre approche sur un réseau de trois intersections (figure 5-b). Les mesures sur ce réseau donnent des résultats équivalents à ceux d'une intersection. Nous avons vérifié dans toutes nos simulations que nous n'avons jamais de collisions entre les véhicules, par conséquent, il s'agit d'une approche décentralisée efficace.

5 Conclusion

Dans cet article nous avons abordé le problème du croisement automatique de véhicules sur un réseau, ou plus généralement d'agents mobiles

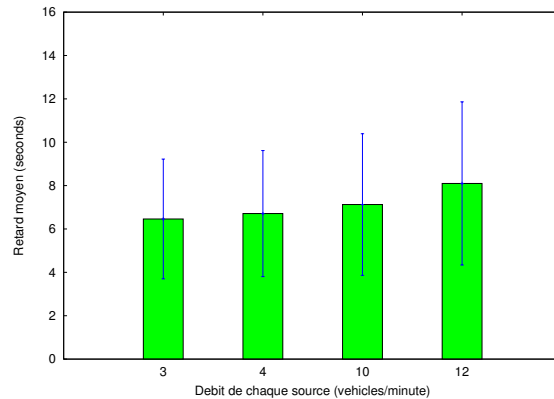


FIGURE 7 – Variation du débit

(par exemple des robots). Pour cela, nous avons proposé une stratégie de contrôle locale aux intersections. Elle consiste à espacer les véhicules de sorte qu'ils peuvent se croiser sans être arrêtés. En synchronisant l'arrivée des véhicules avec une période similaire sur les flux, mais déphasée d'une demi-période, un véhicule de chaque flux passe alternativement. Ainsi les véhicules ne subissent potentiellement qu'un léger ralentissement, ce qui est mieux que de les arrêter.

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

La poursuite de cette étude concerne des expérimentations plus approfondies afin de mesurer les limites de notre approche et de l'évaluer sur des scénarios plus variés (topologie du réseau). L'autre perspective qui motive nos recherches est la volonté de faire communiquer les agents de croisement pour les synchroniser entre eux, dans le but d'améliorer encore le trafic sur le réseau. Cela passe par la recherche de relations de synchronisation entre plusieurs croisements liés par une même route.

Références

BHOURI N., BALBO F., PINSON S. & TLIQ M. (2011). Collaborative agents for modeling traffic regulation systems. In *Proceedings of WI-IAT*, p. 7–13.

- DRESNER K. & STONE P. (2004). Multiagent traffic management : A reservation-based intersection control mechanism. In *Proceedings of AAMAS 2004*.
- DRESNER K. & STONE P. (2005). Multiagent traffic management : An improved intersection control mechanism. In *Proceedings of AAMAS 2005*.
- FOK C., HANNA M., GEE S., AU T., STONE P., JULIEN C. & VISHWANATH S. (2012). A platform for evaluating autonomous intersection management policies. In *Proceedings of Cyber-Physical Systems (ICCPs 2012)*, p. 87–96.
- HOUNSELL N. & SHRESTHA B. (2012). A new approach for co-operative bus priority at traffic signals. *Intelligent Transportation Systems*, **13**(1), 6–14.
- 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., TACKEN J. & TAHEDI C. (1997). Validation and simulation of a decentralized intersection collision avoidance algorithm. In *Intelligent Transportation System*, p. 818–823.
- SCHUEUR A., SIMONIN O. & CHARPILLET F. (2009). Safe longitudinal platoons of vehicles without communication. In *Proc. of ICRA 2009*, p. 2835–2840.

A Calcul des périodes minimales

A.1 Cas $\theta < 90^\circ$

On considère deux flux A et B . L'ordre de passage est $A_0, B_0, A_1, B_1, A_2, B_2, \dots$ (figure 8-a). Pour chaque véhicule on va spécifier deux situations :

- quand un véhicule commence à sortir de la zone de croisement (le véhicule A_0 dans la figure 8-a),
- et quand un véhicule continue à entrer dans la zone de croisement (le véhicule B_0 dans la figure 8-a).

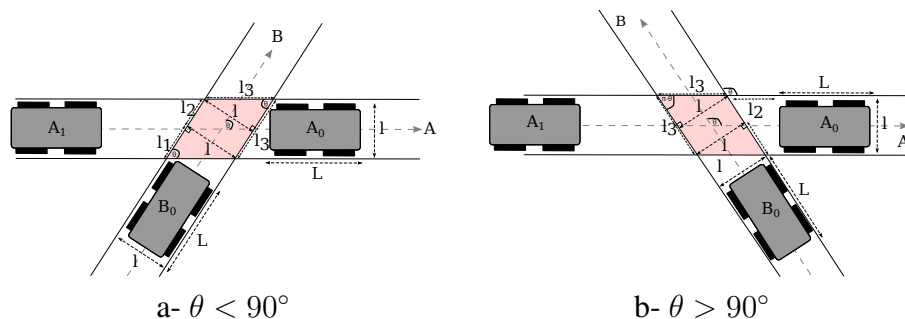


FIGURE 8 – Croisement de deux flux

A partir de ces deux cas, on va essayer, en s'appuyant sur la figure 8-a (qui introduit les distances l_1, l_2, l_3), de déterminer la période et l'inter-distance nécessaires. On observe l'évolution du scénario :

- à $t = 0$, A_0 commence à sortir, B_0 continue à entrer ;

- à $t = (l_2 + L)/V$, B_0 commence à sortir, A_1 continue à entrer ;
- à $t = [(l_2 + L) + (l_2 + L)]/V$, A_1 commence à sortir, B_1 continue à entrer ;
- ...

D'après la figure 8-a, on peut déterminer $l_1 = \frac{l}{\tan \theta}$ et $l_3 = \frac{l}{\sin \theta}$, d'où : $l_2 = l_3 - l_1 = \frac{l}{\sin \theta} - \frac{l}{\tan \theta}$. On peut donc écrire la période sous la forme :

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

A.2 Cas $\theta > 90^\circ$

On considère deux flux A et B . L'ordre de passage est $A_0, B_0, A_1, B_1, A_2, B_2, \dots$. D'après la figure 8-b, on peut déterminer : $l_3 = \frac{l}{\sin \pi - \theta} = \frac{l}{\sin \theta}$ et $l_2 = l_3 |\cos(\pi - \theta)| = l_3 |\cos(\theta)| = \frac{l}{\tan \theta}$. On peut donc écrire la période sous la forme :

$$T = 2 \left(\frac{l}{\sin \theta} + \frac{l}{\tan \theta} + L \right) / V = 2 \left(l \frac{(1 + \cos \theta)}{\sin \theta} + L \right) / V.$$

A.3 Croisement avec routes à deux voies opposées

Considérons deux routes qui se croisent avec deux voies opposées de chaque ((A, C) et (B, D)). Soit λ la distance entre deux voies opposées sur une route. Les deux voies opposées d'une route donnée devrait trivialement traverser l'intersection en même temps, de sorte que l'ordre de passage soit (A_0, C_0), (B_0, D_0), (A_1, C_1), (B_1, D_1)... (figure 9). La

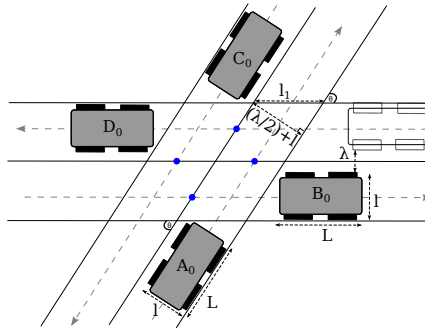


FIGURE 9 – Synchronisation de 4 flux

zone de croisement est maintenant défini par le losange correspondant à l'espace partagé par les deux routes (voir figure 9). Pour traverser et sortir complètement la zone de croisement, un véhicule passe $(2l_1 + L)/V$ temps. D'après la figure 9, nous pouvons déterminer $l_1 = \frac{l + \frac{\lambda}{2}}{\sin \theta}$. Donc la période de passage minimale des deux routes est :

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