



HAL
open science

Régulation du trafic urbain avec priorité aux transports en commun à l'aide d'un système multi-agents

S Haciane, N Bhourri

► **To cite this version:**

S Haciane, N Bhourri. Régulation du trafic urbain avec priorité aux transports en commun à l'aide d'un système multi-agents. 8ème Conférence Internationale de MOdélisation et SIMulation (MOSIM'10), May 2010, HAMMAMET, Tunisie. pp.1450-1458. hal-01473186

HAL Id: hal-01473186

<https://hal.science/hal-01473186v1>

Submitted on 21 Feb 2017

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.

REGULATION DU TRAFIC URBAIN AVEC UNE PRIORITE AUX TRANSPORTS EN COMMUN A L'AIDE D'UN SYSTEME MULTI-AGENTS

Sofiane Haciane, Neïla Bhourri

INRETS / GRETIA

2, Rue de la Butte Verte, 93166 Noisy le grand cedex, France.
neila.bhourri@inrest.fr, hacianesofiane35@yahoo.fr

RESUME : Nous proposons dans ce papier une stratégie de régulation du trafic urbain bi-modal qui se base sur une modélisation multi-agents. L'objectif de cette stratégie est d'agir sur les durées des feux de circulation pour réguler le trafic des voitures particulières et des transports collectifs principalement les bus. Le modèle respecte la méthode de régulation des bus avec un agent ligne qui supervise tous les véhicules de la ligne et veille à ce qu'ils respectent le maintien d'un intervalle régulier entre eux. La priorité bus est accordée par réservation d'une phase verte par les bus dès leur entrée sur les tronçons. Cette phase est accordée en priorité pour les bus qui sont en retard et pour qui la priorité ne détériore pas la régularité de l'intervalle de temps entre eux et leur prédécesseur; intervalle qu'ils doivent respecter aux passages aux arrêts. La régulation est obtenue grâce à des communications, des collaborations et des négociations entre les agents ; la programmation est réalisée sous la plate-forme JADE.

MOTS-CLES : Système multi-agents (SMA), plateforme JADE, collaboration, communication, régulation du trafic urbain, priorité bus.

1 INTRODUCTION

Dans le but d'améliorer le temps de parcours des véhicules de transport collectif de surface (bus, tramways, navettes, etc.), plusieurs villes ont recours aux systèmes de régulations des carrefours à feux permettant de donner la priorité à ces véhicules ; on parle alors de systèmes munis d'une *priorité bus*. L'objectif de ces stratégies est d'augmenter la vitesse moyenne de traversée du carrefour à la fois pour tous les véhicules et pour les véhicules de transport collectif.

L'usage de ces systèmes est efficace quand le trafic n'est pas très chargé ou pour favoriser une seule ligne de bus sur un axe chargé. Des études montrent que l'usage de la priorité bus peut donner un gain variant entre 20 et 40% sur le temps de parcours global des bus (Stif, 2001) Cependant la réduction du temps de parcours des bus, bien que très importante pour l'exploitation d'une ligne, n'est pas le premier facteur considéré par les exploitants des transports collectifs qui ont l'obligation d'accomplir un service vis-à-vis des usagers.

En effet, pour les exploitants d'un réseau de transport collectif, la régulation est un processus d'adéquation en temps réel du tableau de marche aux conditions d'exploitation. Ce tableau contient l'ensemble des missions, chaque mission étant l'ensemble des courses (trajets horodatés) réalisées par un véhicule et un chauffeur durant une journée d'exploitation. Un des objectifs des régulateurs consiste à faire respecter au mieux les horaires du tableau de marche théorique (Bhourri et al., 2008). Cependant, en cas de situation très perturbée, pour respecter les objectifs fondant

l'élaboration du tableau de marche théorique, le régulateur œuvre en le modifiant en temps réel. On distingue trois principales logiques de régulation des transports en commun.

- Logique d'enlèvement de la charge : le tableau de marche est conçu pour charger les voyageurs sans en laisser aux arrêts. En cas de perturbation, le régulateur choisit une procédure permettant de répondre à leur demande tout en prenant en compte la répartition des véhicules sur la ligne.

- Logique de régularité : l'objectif est de minimiser le temps moyen d'attente des voyageurs aux arrêts, ainsi que de répartir la charge sur l'ensemble des bus de la ligne. Cette logique s'applique en heure creuse et aux heures de pointe.

- Logique de ponctualité : l'objectif est de faire respecter les horaires prévus par le tableau de marche théorique. Cette logique s'applique notamment aux premiers et derniers départs de la journée et dans le cas d'échanges inter-modaux.

Une même ligne peut être gérée suivant des logiques différentes, principalement selon l'heure et le lieu de l'intervention. Ainsi, elle peut être gérée en enlèvement de charge et régularité en journée, puis en enlèvement de charge et ponctualité la nuit. En effet, la logique de ponctualité est souvent utilisée quand la fréquence de passage des véhicules de transport en commun n'est pas élevée : le véhicule doit alors respecter au mieux les horaires du tableau de marche théorique. Quand le trafic est dense ou que la fréquence de passage sur une ligne est élevée, les courses sont effectuées à intervalles réguliers. L'importance n'est

plus alors accordée au décalage entre l'horaire théorique et l'horaire réel, mais à l'intervalle entre les véhicules.

Nous nous intéressons dans ce travail à ces conditions de circulation très chargée : le trafic est dense et la fréquence de passage sur les lignes de bus est élevée, en plus nous supposons que plusieurs lignes de bus empruntent le réseau. Il s'agit donc de réguler le trafic global, de favoriser le passage des bus et de veiller à la régularité des intervalles de passages aux arrêts entre les véhicules d'une même ligne de bus.

Le problème de la régulation du trafic global favorisant les bus a été traité dans (Bhouri et Lotito, 2005) et (Bhouri et Touazi, 2008) ; cependant dans ces recherches, la régulation de l'intervalle n'a été traitée que d'une manière indirecte. Le problème abordé était de réduire la perte de temps dans les embouteillages de manière que le bus respecte son tableau de marche. (Kachroudi et Bhouri, 2008 et 2009) ont traité le problème de la régulation du trafic global avec en plus une régulation de l'intervalle entre les bus. La dynamique du système, dans ces travaux, a été décrite d'une manière macroscopique pour le trafic général et d'une manière microscopique pour les bus. Une optimisation multi-objective a été appliquée au système en utilisant la méthode d'optimisation par essais particuliers. La complexité du modèle nous laisse penser que l'on peut obtenir de meilleurs résultats grâce aux systèmes multi-agents. C'est ce que nous testons dans cette recherche.

L'application des systèmes multi-agents est de plus en plus présente pour la recherche dans le domaine de la régulation du trafic. Nous pouvons citer par exemple les travaux de (Bazzan, 2008), (De Oliveira et al. 2005), (Mailler et Lesser, 2004) et (Mizuno et al., 2008) qui ont essayé de résoudre le problème de la coordination des feux sur des artères du réseau routier. (Ferreira et al. 2001), (France et Ghorbani, 2003) et (Roozmond, 2001) ont quand à eux posé le problème du contrôle des feux des carrefours pour améliorer la circulation. Nous ne connaissons aucune application pour la régulation du trafic bimodal qui est l'objet de notre recherche.

La section suivante de cet article sera consacrée à la modélisation du réseau et à l'identification des agents pour la modélisation. Nous consacrons la deuxième section à la description détaillée des agents, leurs attributs, leurs objectifs ainsi que les communications et collaborations entre eux. La quatrième section donne les détails du réseau et des résultats des tests en simulation effectués. Enfin nous concluons dans la cinquième section.

2 MODELISATION DU RESEAU

Le réseau urbain est modélisé par un graphe orienté $G = (I, A)$. Les nœuds (I) représentent les carrefours (ou intersections) et les arcs (A) représentent les voies qui relient ces carrefours. Deux intersections peuvent être reliées par un ou plusieurs arcs suivant le nombre de voies sur les tronçons.

Arc : un arc correspond à une voie. Il est caractérisé par un ensemble d'informations statiques et dynamiques, notamment la liste des arcs successeurs vers lesquels les véhicules peuvent se diriger, sa longueur L (en mètres), sa capacité C (en unité de véhicule particulier, UVP), son débit de saturation qui est le débit maximum de sortie de l'arc D (en UVP/seconde), le nombre de véhicules présents sur l'arc, N_i (en UVP) et enfin l'état des feux à l'extrémité de l'arc : *vert* ou *rouge*. Si le feu est vert alors les véhicules présents sur l'arc peuvent le quitter.

Carrefour : un carrefour est spécifié par l'ensemble des arcs qui y entrent (E) et l'ensemble des arcs qui en sortent (S). un carrefour est géré par un ensemble de phases $P = \{P_1, P_2, \dots, P_n\}$. Chacune des phases spécifie la liste des arcs auxquels est accordé le feu vert si la phase est active.

La figure suivante illustre un exemple d'un carrefour à deux phases $P = \{P_1, P_2\}$. P_1 permet d'évacuer les arcs a_1 et a_3 . P_2 évacue les arcs a_5 et a_8 . Les entrées/sorties du carrefour sont respectivement $E = \{a_1, a_3, a_5, a_8\}$ et $S = \{a_2, a_4, a_7, a_6\}$.

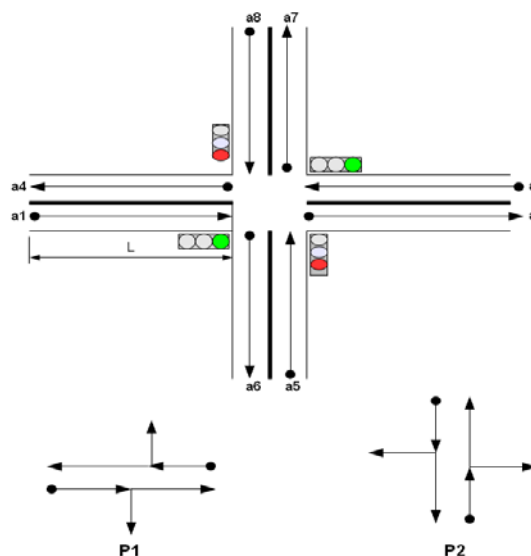


Figure 1: Exemple d'un carrefour à quatre arcs et deux phases.

Le réseau est emprunté par un ensemble de lignes de bus. Chaque ligne est constituée par un ensemble de bus qui ont la même origine, la même destination et desservent à des intervalles de temps, qui devraient être réguliers entre les différents bus de la ligne, un ensemble d'arrêts commerciaux prédéfinis. Comme nous le verrons par la suite, le temps passé par un bus à un arrêt commercial sera égal à un temps fixe prévu pour la montée des passagers plus un temps qui sera rajouté pour la régulation de l'intervalle quand nécessaire.

2.1 Identification des agents

Afin d'identifier les agents du système, nous faisons abstraction du système réel : pour chaque entité du monde réel nous associons une entité (agent) dans le monde virtuel.

L'ensemble de ces entités interagissent entre elles et forment un système multi-agents (SMA). Nous appelons type d'agent ou classe d'agent un ensemble d'agents *homogènes*. Notre SMA est constitué des types d'agent suivants :

Agent carrefour (AC) : c'est l'agent clé de notre architecture. Cet agent est chargé de contrôler un carrefour à feux et de générer les plans de feux en fonction de son état actuel mais aussi de l'état actuel des carrefours adjacents. Deux carrefours sont adjacents s'ils sont reliés par un ou plusieurs arcs.

Agent phase (AP) : vue la complexité du processus de l'élaboration des plans de feux. Nous envisageons des agents phase. L'élaboration des plans de feux pourra être ainsi résolue d'une façon collaborative par les agents phase. Chaque agent phase se charge de déterminer la durée de temps de vert optimale pour évacuer les véhicules en attente de cette phase. Ainsi, quelle que soit la complexité du carrefour (quelle que soit sa configuration physique), il est vu comme un ensemble d'agents phase qui interagissent avec l'agent carrefour pour élaborer le plan de feux.

Agent bus (AB) : un agent bus représente un bus du monde réel. Il circule d'un arc à un autre, s'arrête aux arrêts commerciaux, s'arrête aux feux rouges et obéit aux consignes de l'agent ligne. L'objectif de chaque agent bus est de minimiser son temps perdu aux feux (i.e. : minimiser son temps de parcours).

Agent ligne de bus (AL) : les agents bus (AB) possèdent une vue locale de leur environnement et plus particulièrement de la course qu'ils effectuent. Ainsi, l'optimisation locale effectuée par les agents bus peut avoir un impact négatif sur la ligne, notamment sur sa régularité (i.e. : la création de trains de bus). Pour remédier à ce problème, nous proposons un agent qui a une vue globale sur les agents de la ligne, qui peut contrôler et modifier leur comportement pour assurer un service régulier.

2.2 Description des agents

Dans cette partie nous donnons une description de l'architecture interne des agents de notre système.

2.2.1 Agent bus

L'objectif de chaque agent bus est de minimiser son temps d'arrêt aux feux. Pour atteindre cet objectif, l'agent bus interagit avec les agents carrefour et son supérieur hiérarchique (agent ligne). L'ensemble des bus doivent offrir un service régulier et éviter les *trains de bus*, en d'autres termes, la fréquence de passage des bus par les arrêts doit rester stable. Pour atteindre ce dernier objectif, l'agent bus peut recevoir des ordres de l'agent ligne (par exemple, *rester en arrêt durant t secondes* si le bus est en avance par rapport à la position de son prédécesseur).

L'agent bus est constitué d'un module de données qui représente son état interne et d'un module de communication qui lui permet d'échanger avec les autres agents du système.

Comportement de l'agent bus

L'agent bus est injecté dans sa ligne à l'instant t_0 , l'agent se comporte de la façon suivante :

- à l'entrée de chaque arc, l'agent bus récupère les informations de cet arc (le nombre des véhicules qui sont passés devant lui, la longueur, la capacité et le débit de sortie de l'arc). En utilisant ces données, l'agent calcule une réservation d'un espace-temps et l'envoie à l'agent carrefour pour éviter un éventuel arrêt au feu rouge au carrefour suivant. L'agent carrefour tente ensuite de satisfaire cette demande (voir agent carrefour),
- à la perception d'un arrêt, l'agent bus en informe l'agent ligne auquel il est rattaché. L'agent ligne calcule ensuite la durée de régulation et son nouveau niveau de priorité et les envoie au bus. Le bus doit attendre pendant le temps de chargement des usagers plus éventuellement cette durée de régulation pour quitter l'arrêt actuel.

La figure 2 représente le comportement de l'agent bus comme un automate à état fini : C (en Circulation), A (en arrêt) et F (en attente au feu). Les paramètres entre parenthèses correspondent à la condition de transition et les paramètres entre crochets correspondent à l'action (y compris l'envoi de messages) exécuté après la transition.

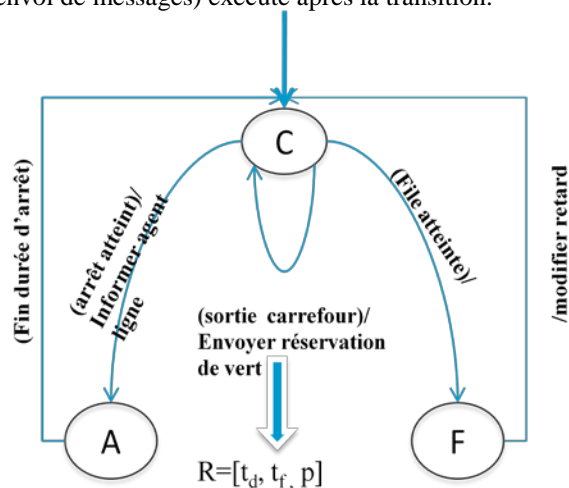


Figure 2 : diagramme de transition de l'agent bus.

Calcul d'une réservation de vert

Une réservation d'un temps de vert est spécifiée par l'intervalle de temps pendant lequel le feu vert doit être accordé à l'arc actuel pour que le bus passe sans s'arrêter au prochain carrefour.

Soit t_0 l'instant actuel, la réservation R est donnée par : $R=[t_d, t_f]$

t_d et t_f étant respectivement les instants de début et de fin de la réservation. Le calcul de ces instants est réalisé de la manière suivante : le bus entrant sur l'arc trouve N véhicules devant lui, ceux-ci vont se placer sur la ligne de feux en attendant le feu vert formant ainsi une file d'attente de longueur F (voir figure 3). Pour que le bus ne s'arrête pas,

il faut évacuer cette file d'attente avant son arrivée. Le feu vert devrait donc être accordé à l'arc à l'instant :

$$t_d = t_0 + T - EvF$$

où T est le temps nécessaire au bus pour parcourir la distance Dis le séparant de la file d'attente et EvF le temps nécessaire pour évacuer la file d'attente F (voir figure 3). Substituant chacun par sa valeur, nous obtenons la formule :

$$t_d = t_0 + 3.6 * \left(\frac{L}{V} - \frac{N * L}{C * V} \right) - \frac{N}{D}$$

Utilisant le même raisonnement, nous obtenons l'instant de fin de la réservation, t_f :

$$t_f = t_d + \left(\frac{N * L}{3.6 * C * V} \right)$$

Où :

- L : la longueur de l'arc en m,
- C : la capacité de l'arc en UVP,
- D : le débit de saturation de l'arc en UVP/seconde
- N : le nombre de véhicules qui se trouvent sur l'arc devant le bus en UVP,
- V : la vitesse du bus en km/h.

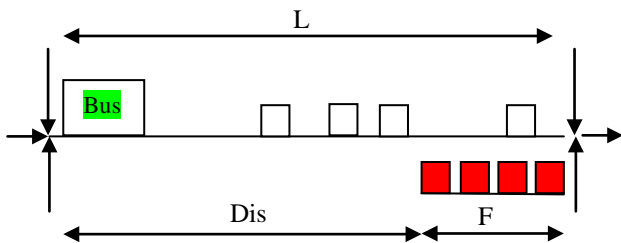


Figure 3 : Réservation de temps de vert par un bus.

La réservation R accompagnée d'autres informations (numéro du bus, sa priorité, l'arc actuel du bus, le prochain arc vers lequel transitera le bus) sont envoyées à l'agent carrefour (le carrefour prochain) qui tente de modifier son plan de feux pour satisfaire cette réservation.

2.2.2 Agent carrefour

Chaque agent carrefour est chargé de contrôler un carrefour à feux et de gérer les plans de feux optimaux ou quasi-optimaux. L'ensemble des agents carrefours communiquent entre eux pour éviter une optimisation locale. Un agent carrefour supervise un ensemble d'agents phase (AP). Ceux-ci collaborent entre eux pour établir un plan de feux qui d'une part, maximise la capacité du carrefour et d'autre part, essaye de satisfaire au mieux les réservations des bus. Cet agent possède un ensemble de données statiques et dynamiques qui le caractérisent.

Données statiques

Les données statiques représentent les contraintes auxquelles est soumis l'agent carrefour, on trouve :

- l'ensemble des valeurs possibles pour le cycle : la longueur du cycle (en secondes) ne doit pas excéder une cer-

taine valeur. Généralement le cycle prend l'une des valeurs suivantes : cycle $\in \{40, 60, 80, 100, 120, 140\}$. Quand le trafic est dense dans une ou plusieurs directions, on augmente la longueur du cycle. En effet, pour chaque cycle, il y a une période de temps perdu comme la période de l'orange ou bien la période de *rouge intégral*. Le rouge intégral étant une période durant laquelle tous les arcs d'un même carrefour ont un feu rouge afin de dégager le centre du carrefour et éviter les accidents. Cette période fixée suivant l'architecture du carrefour ne dépend pas de la longueur du cycle, d'où le fait que plus un cycle est court plus il y a du temps perdu et une perte de capacité du carrefour.

- L'ensemble des phases du carrefour : $P = \{P_1, P_2, \dots, P_m\}$: l'ensemble des phases représente la configuration du carrefour (les mouvements permis). La détermination des phases est une tâche exécutée en temps différé par des experts de trafic.

Données dynamiques

- Le plan de feux : le plan de feux spécifie l'ordre de passage des phases ainsi que la durée de chaque phase. Entre deux phases successives une durée de rouge intégral de 2 secondes est imposée.
- La liste des réservations reçues : chaque réservation est spécifiée comme suit : $R = (P_i, t_d, t_f, \text{Priorité})$. où P_i est la phase qui va permettre le passage du bus, t_d l'instant où le bus prévoit d'arriver à la ligne de feu, t_f l'instant auquel l'arrière du bus quitte l'arc et enfin « Priorité » est le niveau de priorité défini par l'Agent ligne pour ce bus.

Comportement de l'agent carrefour

A la fin de chaque cycle l'agent carrefour lance le processus de calcul du plan de feux pour le cycle prochain. Ce plan doit déterminer la durée de vert et l'ordre de passage de chaque phase.

Quand l'agent carrefour reçoit une réservation, il l'enregistre dans sa base de données. Une réservation $R = (P_i, t_d, t_f, \text{Priorité})$ n'est pas prise en compte au moment de sa réception (t_0). L'agent carrefour décidera d'accepter ou de refuser cette demande à l'instant t_d .

Nous explicitons le comportement de l'Agent carrefour grâce à l'exemple suivant (voir Figure 4) dans lequel le carrefour a un plan à quatre phases et deux lignes de bus antagonistes.

Dans cet exemple, t représente l'instant actuel ; à l'instant t_1 le carrefour reçoit la réservation R_1 qui demande l'extension de la phase P_1 jusqu'à l'instant t_4 pour un bus qui a le niveau de priorité 2. A l'instant t_2 l'agent carrefour reçoit une autre réservation, R_2 d'un bus plus prioritaire (Priorité = 4). Ce dernier demande la phase P_4 . La satisfaction des deux demandes (R_1 et R_2) n'est pas possible car elles concernent deux phases différentes à deux intervalles de temps qui chevauchent entre eux. L'agent carrefour retarde le plus possible la modification de son plan. Ainsi, la demande de réservation R_1 est étudiée à l'instant t_3 (instant de son début) et non pas à l'instant t_1 (instant de sa réception). Ce retardement permet à l'agent carrefour de

recevoir plus de réservations (recevoir R_2 à l'instant t_2 dans l'exemple) et d'éviter des modifications inutiles car finalement R_1 est refusée puisque la réservation R_2 est plus prioritaire.

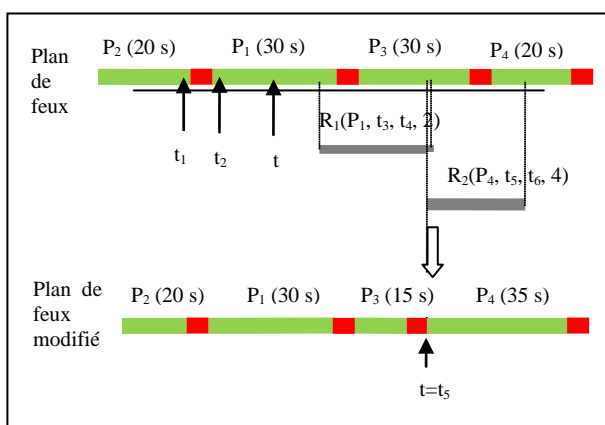


Figure 4 : Exemple d'un plan de feux et sa modification

La modification d'un plan de feux suite à une demande de priorité par un bus peut concerner :

- L'extension d'une phase (retarder ou avancer une phase), sans dépasser la durée maximale d'une phase.
- Faire un appel de phase : insérer une nouvelle phase sur le plan.

Calcul d'un plan de feux

Le plan de feux est calculé par la collaboration entre l'agent carrefour et les agents phase. A chaque phase est associé un agent phase qui se charge de calculer sa durée.

L'agent carrefour joue le rôle de *Manager* en supervisant les agents phase qui jouent le rôle de *participants*.

- 1) L'agent carrefour commence par la création d'un groupe de collaborateurs appelé *collab_group* contenant la liste des agents phase qu'il doit gérer.
- 2) L'agent carrefour initialise les variables : $C = \text{CycleMax}$ (140 secondes) et $t = 0$. La variable C permet de contrôler la taille du cycle calculé.
- 3) L'agent carrefour envoie un message aux agents phase pour les informer que le protocole de calcul d'un plan de feux est lancé.
- 4) L'agent carrefour envoie un message *request* aux agents de *collab_group* leur demandant le temps nécessaire pour évacuer tous les véhicules de leurs phases, en commençant à l'instant t .
- 5) Chaque agent, i , de *collab_group* calcule sa durée de vert souhaitée (d_i) et un index qui mesure l'urgence de la phase (I_i) et les renvoie à l'agent *Manager*. Nous présentons dans la partie consacrée à la description de l'agent phase la façon dont celui-ci calcule ces paramètres (d_i et I_i).
- 6) Quand le *Manager* reçoit toutes les réponses, il calcule la somme de toutes les durées : $d = \sum_{i=0}^m d_i$. Si $d > C$ alors le *Manager* doit résoudre le conflit (la taille du cycle dépasse le maximum). La résolution du conflit se termine par la modification des d_i calculés auparavant de telle sorte que $d \leq C$.

7) Le *Manager* choisit la phase la plus urgente, soit P_j , sa durée est d_j :

- il envoie un message *accept* à l'agent phase qui gère cette phase ;
- il retire l'agent phase correspondant de *collab_group* ;
- il met à jour les variables $C = C - d_j$, $t = t + d_j$;
- et enfin il passe à 4) tant que *collab_group* n'est pas vide.

Résolution de conflit

Quand la somme des durées de vert demandées par les agents phase dépasse la taille du cycle, l'agent *Manager* doit ramener cette somme à la valeur maximale du cycle. Pour ce faire le *Manager* utilise la négociation avec les agents phase pour réduire cette somme avec un minimum de coût. L'agent *Manager* lance un appel d'offre auprès des agents phase. Les agents phase peuvent offrir une partie (Δt) de leur durée de vert pour résoudre le problème. Le coût d'une offre se traduit par le nombre de lignes de bus pénalisées si l'agent phase offre une quantité Δt de son temps de vert.

- 1) l'agent *Manager* initialise le coût $c = 1$;
- 2) il envoie l'appel d'offre (*cfp*) avec le coût c à tous les agents phase de son carrefour,
- 3) les agents phase répondent avec un message *propose* ou *refuse* (les modalités sont ci-dessous expliquées dans la section consacrée au comportement de l'agent phase).
- 4) Une fois toutes les réponses reçues par l'agent *Manager*, il fait la somme de toutes les offres soit : $S = \sum_{i=0}^n \Delta t_i$.
- 5) Si cette somme permet de ramener la taille du cycle à la valeur souhaitée alors le conflit est résolu. Sinon, l'agent *Manager* incrémente le coût c et relance son appel à nouveau (aller à l'étape 3)

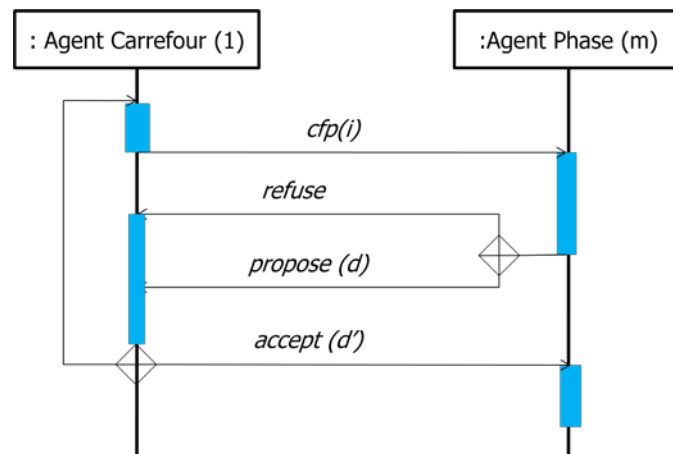


Figure 5 : Collaboration pour la résolution des conflits

2.2.3 Agent phase

Un agent phase gère un ensemble de mouvements compatibles à l'entrée d'un carrefour. Il est chargé de calculer la durée optimale pour évacuer les véhicules en attente sur les arcs concernés et de collaborer avec les autres agents pour élaborer le plan de feux.

Etat interne de l'agent phase

Cet agent possède un ensemble de données statiques et un ensemble de données dynamiques qui représentent son état interne :

Données statiques

- Liste des arcs entrants : l'ensemble des arcs qui sont autorisés à évacuer si la phase est active (ou verte).
- le carrefour origine pour chaque arc entrant.

D'une façon plus formelle la liste des arcs entrants est représentée par $E = \{(a_1, c_1), (a_2, c_2), \dots, (a_n, c_n)\}$ tel que a_i est l'arc entrant i , c_i est le carrefour origine de l'arc a_i . Cette structure de données permet à l'agent phase de récupérer les informations du trafic entrant à l'intersection par cette phase et de communiquer avec les agents carrefours voisins qui peuvent envoyer un trafic au carrefour actuel.

Données dynamiques

Les données dynamiques concernent :

- Etat de la phase : active ou non active. Active signifie que les feux qui gèrent les arcs concernés par cette phase sont au vert. Les véhicules sont donc autorisés à les quitter.
- Durée de vert affectée à la phase,
- Date de début de l'exécution de la phase.

Comportement de l'agent phase

L'agent phase participe au processus de calcul du plan de feux. Il se charge de fixer la durée de vert optimale pour sa phase.

- 1) Quand l'agent phase est interrogé par l'agent carrefour sur la durée de vert souhaitée, il calcule cette durée (d_i) et un index qui mesure l'urgence de la phase (I_i) et les transmet à l'agent carrefour.
- 2) Si l'agent phase reçoit la confirmation de l'agent carrefour alors l'agent phase arrête ce processus.
- 3) Si l'agent phase reçoit un appel d'offre avec un coût c , il calcule une offre et l'envoie à l'agent carrefour.

Calcul de la durée de vert souhaitée

La durée de vert optimale est calculée par la formule suivante :

$$T = \max_{i=1..m} \{T_i\}$$

Où m est le nombre d'arcs entrants par cette phase et T_i le temps nécessaire pour évacuer l'arc i et qui est donné par :

$$T_i = \frac{N_i}{d_i} + \frac{N_i * L_i}{C_i * V_i}$$

Indice d'urgence d'une phase

L'indice d'urgence d'une phase j pour permettre de donner la priorité à un bus est défini de sorte que plus cet indice est grand plus la phase est urgente :

$$I_j = \sum_{i=1}^m e^{w_i} + e^{b_i}$$

Où :

- $w_i = \frac{N_i}{C_i}$: mesure le degré de congestion de l'arc i , inférieur à 1.
- b_i : le nombre de bus présents sur l'arc i .
- m : le nombre d'arc entrant via la phase j .
- e : constante (strictement supérieur à 1) ; nous avons utilisé la constante d'Euler dans notre cas.

Calcul d'une offre

Quand la taille du cycle dépasse le maximum (140 secondes pour notre exemple), l'agent carrefour fait un appel d'offre auprès des agents phase qui peuvent céder une partie de leur durée de vert. Une offre possède un coût c qui se traduit par le nombre d'arcs sur lesquels resteront des véhicules après exécution de la phase. L'agent phase calcule une offre comme suit :

- Soit $E = \{T_1, T_2, \dots, T_m\}$ la liste dont les éléments sont les temps de vert nécessaires pour évacuer chaque arc a_i concerné par cette phase, de la totalité de ses véhicules, triée par ordre croissant (i.e. : $T_m \geq T_{m-1} \geq \dots \geq T_1$).

- Soit c le coût de l'offre (ce coût est communiqué par l'agent carrefour lors de son appel d'offre).

L'offre correspondant au coût c est calculé par :

$$\Delta T = T_m - T_{m-c} \text{ tel que } c \leq m \text{ et } T_0 = 6 \text{ secondes (le vert minimum)}$$

ΔT : La durée de vert supplémentaire que l'agent phase cédera à l'agent carrefour.

2.2.4 Agent ligne

Le rôle de l'agent ligne est de superviser un ensemble d'agents bus pour éviter une régulation au niveau local du bus et la création des trains de bus. En d'autres termes, cet agent peut modifier le comportement des agents bus de deux façons différentes :

D'une façon directe et cela en gardant les bus ayant une avance par rapport à leur prédécesseur au niveau des arrêts pendant une certaine période, ou d'une façon indirecte et cela en modifiant les priorités des bus. Cet agent possède une vue globale sur la ligne qu'il gère, cela lui permet de détecter les trains de bus et d'agir en conséquence pour décomposer ces trains formés. Dans l'exemple suivant (Figure 6), l'agent ligne AL123 doit détecter le train de bus formé par les agents bus B4, B5 et B6 et de le décomposer en retardant B6 et B5 qui sont en avance.

Chaque ligne est constituée d'un ensemble de bus qui desservent cette ligne (B_1, B_2, \dots, B_m). Les bus sont initialement stockés au niveau de la station initiale (A_0). A chaque période f (fréquence) un bus est injecté dans la ligne. L'idéal est d'avoir pour chaque arrêt de la ligne une fréquence de passage égale à f .

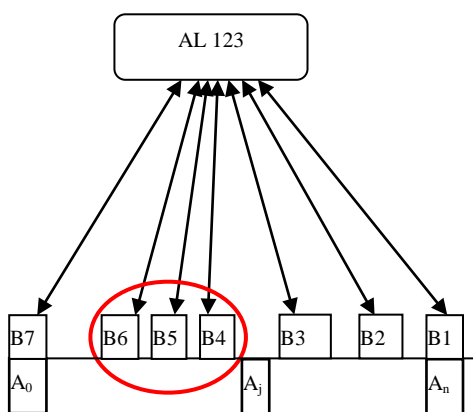


Figure 6 : Supervision des agents bus par l'agent ligne bus.

Etat interne de l'agent ligne

L'agent ligne bus possède la structure de données suivante :

- La liste des arcs qu'empruntent les bus de cette ligne.
- La liste des arrêts de la ligne : pour chaque arrêt, sa position et la distance qui le sépare de l'arrêt suivant.
- La liste des bus de la ligne.
- La fréquence d'injection des bus sur la ligne.

Pour deux arrêts consécutifs A_i et A_j , l'agent ligne garde la durée de parcours $d_{i,j}$ du dernier bus. Cela permet de suivre l'évolution des bus et de calculer l'avance ou le retard d'un bus par rapport à son prédécesseur.

Comportement de l'agent ligne

Quand un agent bus passe à l'état "en arrêt", il transmet à l'agent ligne le temps consommé (t) pour parcourir la distance $L_{i,i-1}$ qui sépare les deux arrêts A_i et A_{i-1} . L'agent ligne compare t à la durée ($d_{i,j}$) consommée par le bus précédent et décidera par la suite si le bus est en avance ou en retard. L'agent ligne calcule la nouvelle priorité de l'agent bus ainsi qu'une durée que le bus doit attendre à son arrêt commercial s'il est en avance. La priorité du bus est recalculée comme suit :

- Si $d_{i,j} > t$ alors $p=0$ (le bus est en avance).
- Si $t - d_{i,j} \in [0, 0.1 * d_{i,j}[$ alors $p=1$,
- Si $t - d_{i,j} \in [0.1 * d_{i,j}, 0.2 * d_{i,j}[$ alors $p=2$,
- Si $t - d_{i,j} \in [0.2 * d_{i,j}, 0.3 * d_{i,j}[$ alors $p=3$,
- Si $t - d_{i,j} \in [0.3 * d_{i,j}, 0.4 * d_{i,j}[$ alors $p=4$,
- Si $t - d_{i,j} \geq 0.5 * d_{i,j}$ alors $p=5$.

3 IMPLEMENTATION DU PROTOTYPE

Nous avons implémenté un prototype de système multi-agents sous la plateforme JADE.

3.1.1 Environnement JADE (Java Agent DEvelopment framework)

JADE est une plate-forme multi-agents créée par le laboratoire TILAB (Telecom Italia Lab). JADE (Jade, 2009) permet le développement de systèmes multi-agents et d'applications distribuées conformes aux normes FIPA (Foun-

dation for Intelligent Physical Agent). Aucune méthodologie n'est spécifiée.

JADE possède trois modules principaux (nécessaire aux normes FIPA).

- Director Facilitator : fournit un service de « pages jaunes » à la plate-forme,
- Agent Communication Channel : gère la communication entre les agents,
- Agent Management System : supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système.

Le langage de Communication de la plate-forme JADE est FIPA-ACL (Agent Communication Language).

3.1.2 Le réseau urbain utilisé pour les tests en simulation

Nous testons la stratégie sur un réseau de douze carrefours. La configuration du réseau est la suivante :

- Les carrefours possèdent entre deux et quatre phases,
- La distance entre deux carrefours adjacents varie entre 200 et 400 mètres.
- Chaque section est constituée d'une ou deux voies (voir figure ci-dessous).
- Le débit maximum de sortie des arcs est identique et est égal à 0.6 véhicules/seconde.
- A chaque entrée du réseau on installe une source qui génère des véhicules à une fréquence $F \in [4 \text{ s} .. 10 \text{ s}]$

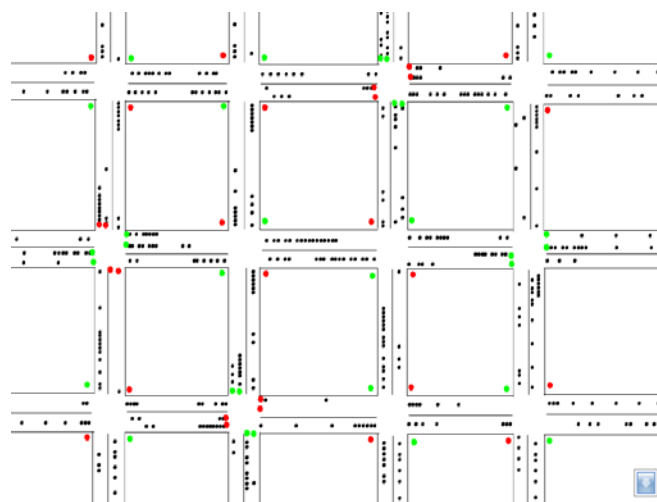


Figure 7 : architecture du réseau utilisé.

Chaque véhicule qui parcourt le réseau calcule le temps consommé pour traverser le réseau ainsi que la proportion du temps perdu aux feux. Nous faisons la moyenne du temps perdu sur chaque 30 secondes.

3.1.3 Résultats

Nous comparons notre stratégie à une stratégie temps fixe. Cette dernière utilise des plans de feux fixés à l'avance qui ne varient pas en fonction de la charge du réseau, contrairement à notre stratégie qui recalcule un nouveau plan à la

fin de chaque cycle. Nous ne donnons ici que les résultats des retards enregistrés par les véhicules particuliers ; d'autres tests en simulation seront bientôt effectués pour analyser le comportement des bus.

Nous considérons deux scénarios. Dans le premier, la fréquence d'entrée des véhicules est de 5 secondes. Cela veut dire pour toutes les entrées, un véhicule est injecté chaque 5 secondes, pour le second la demande sera variable d'une entrée à l'autre même si nous conservons le même volume de trafic sur la totalité du réseau (Haciane, 2009).

Les résultats pour le premier scénario sont donnés sur les figures 8 et 9 respectivement pour la stratégie temps fixes et la stratégie multi-agents (SMA).

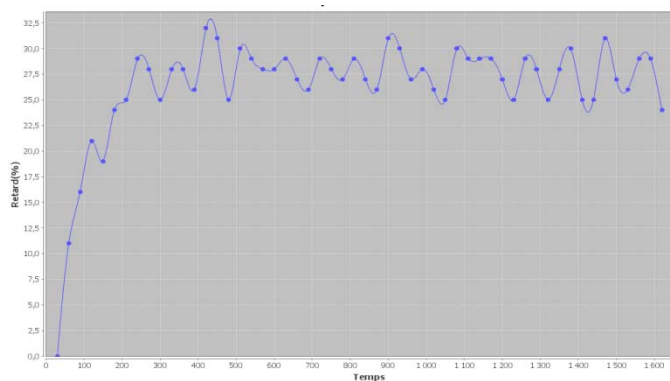


Figure 8 : Évolution du retard en fonction du temps à l'aide de la stratégie à temps fixe pour le scénario 1

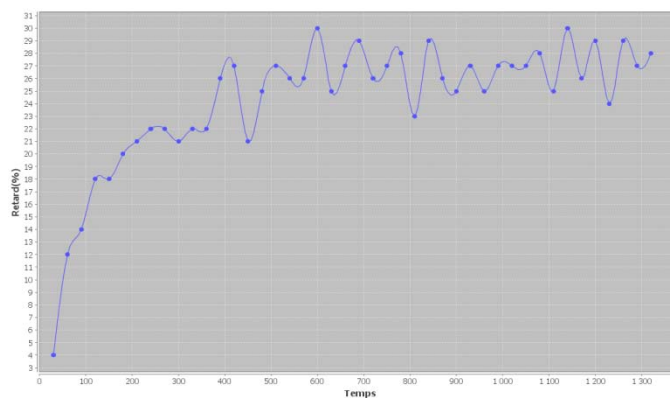


Figure 9 : Évolution du retard en fonction du temps quand la stratégie SMA est utilisée pour le scénario 1

Nous remarquons une légère amélioration. Cela est dû à la situation du trafic qui reste stable et le choix de la stratégie que nous avons utilisé pour contrôler la situation.

Pour le scénario 2, nous modifions donc la fréquence d'entrée des véhicules au niveau des sources installées aux entrées du réseau : la fréquence d'entrée des véhicules de l'est et de l'ouest du réseau est de 3 secondes. Par contre, la fréquence d'entrée du sud et du nord est de 7 secondes. Les résultats sont donnés par les deux courbes sur les figures 10 et 11.

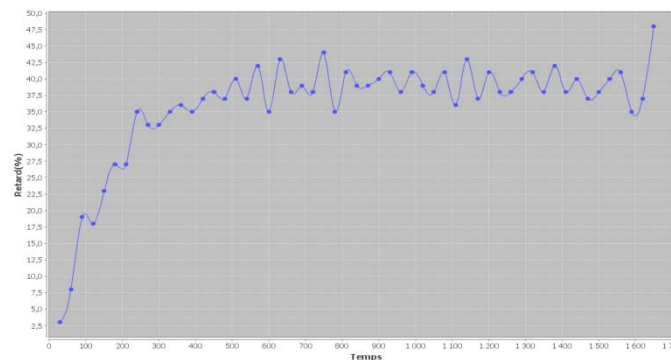


Figure 10 : Évolution du retard en fonction du temps quand la stratégie temps fixes est appliquée pour le scénario 2

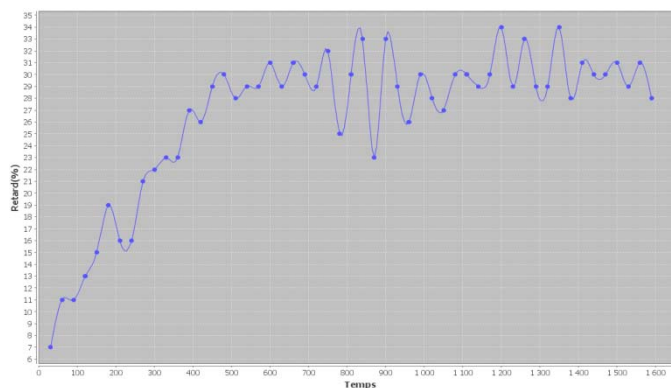


Figure 11 : Évolution du retard en fonction du temps quand la stratégie SMA est appliquée pour le scénario 2

Nous remarquons que les résultats donnés par la stratégie SMA restent inchangés malgré le changement effectué sur le volume du trafic car les plans sont élaborés en fonction de la situation actuelle du trafic (figure 11).

Cependant, le retard des véhicules a augmenté d'une façon remarquable si on utilise la stratégie fixe (figure 10).

Coût de la communication entre les agents

Dans notre modélisation nous avons fait appel à la négociation, qui se passe par l'échange de messages entre plusieurs agents pour résoudre les conflits. Cependant, cette négociation n'est déclenchée que lorsque la taille du cycle dépasse un certain seuil (140 secondes) qui est égale au cycle de calcul. Cela signifie que la négociation reste faisable même si le réseau est de grande taille.

Dans le cas où le trafic est dense, la taille du cycle devient plus longue, les agents utilisent souvent la négociation car les durées demandées par les agents phase ne sont pas satisfaites.

Dans le cas où le trafic n'est pas dense, la négociation est absente. Par contre le cycle de calcul devient court.

La figure 12 montre la quantité de messages échangés entre un carrefour C8 et les agents phase C81, C82, C83 pour calculer le plan de feux dans une situation où le trafic n'est pas dense.

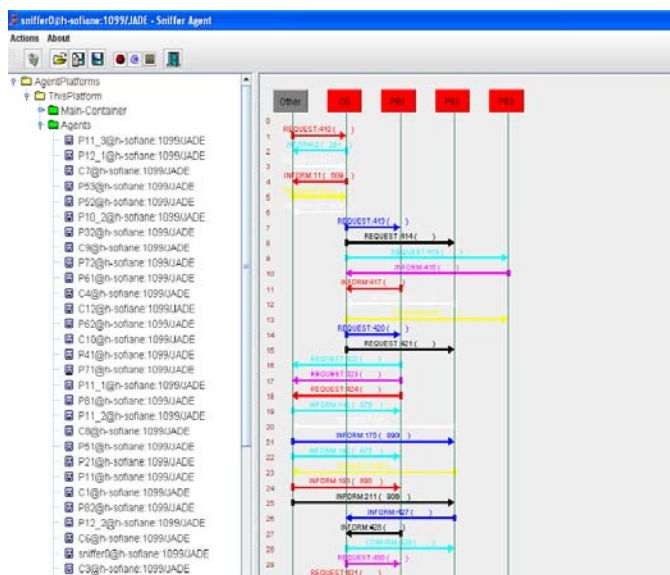


Figure 12 : Exemple d'interaction entre les agents

Le temps de calcul reste très faible et le traitement par les agents de la régulation du trafic bimodale peut être appliqué en temps réel.

4 CONCLUSION

L'usage des méthodes classiques de commande des systèmes pour la régulation du trafic urbain bimodal présente quelques inconvénients. En effet, le problème concerne la régulation sur de grands réseaux d'un trafic mixte. Une modélisation macroscopique du système ne peut permettre de réguler les intervalles entre les bus. D'autre part, la modélisation microscopique est très consommatrice en temps de calcul et n'est pas pratique pour le développement d'une loi de commande. Dans ce papier, nous avons montré qu'une régulation sur la base d'un système multi-agents permet de résoudre le problème et d'améliorer la circulation sur le réseau tout en respectant les intervalles des bus. Grâce aux nouvelles technologies de l'information et de la communication, les entités constituant le réseau urbain peuvent communiquer entre elles et négocier pour résoudre les problèmes liés au trafic.

Nous avons présenté dans la seconde partie un premier modèle multi-agents qui permet de calculer les plans de feux en temps réel en fonction de la situation actuelle du trafic.

Les résultats montrent que la stratégie développée donne de meilleurs résultats dès que le trafic est perturbé, par contre quand le trafic est stable la stratégie temps fixe donne d'aussi bons résultats que la stratégie développée.

REFERENCES

Bazzan A. L. C. 2008. "Opportunities for multiagent systems and multiagent reinforcement learning in traffic

control", *Published online: 7 September 2008, Springer Science+Business Media, LLC 2008*

Bhouri, N., F. Boillot F. et P. Vinant, 2008. Régulation multimodale du trafic routier et des transports en commun de surface. Une classification des méthodes. *Revue Recherche Transport Sécurité RTS n° 98* Janvier-Mars 2008.

Bhouri, N. et P. Lotito, 2005. An intermodal traffic control strategy for private vehicle and public transport. Tenth Euro Working Group on Transportation, Poznan, Pologne.

Bhouri, N. et D. Touazi, 2008. Commande prédictive à base de modèle pour le trafic urbain bi-modal. *5ème Conférence Internationale Francophone d'Automatique, CIFA'2008*, Bucarest, Roumanie, 3-5 septembre 2008.

De Oliveira D, A.L. Bazzan et V. Lesser V, 2005. Using Cooperative Mediation to Coordinate Traffic Lights, *AAMAS'05, July 2529, 2005, Utrecht, Netherlands*.

Ferreira, E.D., E. Subrahmanian et D. Manstetten, 2001. Intelligent agents in decentralized traffic control, *IEEE Intelligent Transportation Systems Conference Proceedings - Oakland (CA), USA - August 25-29*.

France, J. et A. Ghorbani, 2003. A multiagent system for optimizing urban traffic. *In Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (pp. 411-414)*. Washington, DC: IEEE Computer Society. 2003.

Haciane S., 2009. *Régulation du trafic des Transports en Commun par Système Multi-Agents*. Stage de Master II. Avril-Aout.

JADE : Site de présentation de la plateforme JADE par TILAB <http://jade.tilab.com/>. Site consulté le 20 novembre 2009.

Kachroudi, S. et N. Bhouri, 2008. A multimodal model for an urban traffic control policy. *17th IFAC World Congress Seoul - Corée, 6-11 juil.*

Kachroudi, S. et N. Bhouri, 2009. A multimodal traffic responsive strategy using particle swarm optimization. *12th IFAC Symposium on Control in Transportation Systems (CTS'09)*. Redondo Beach, California, USA, September 2-4.

Mailler R. et V. Lesser V, 2004. Solving distributed constraint optimization problems using cooperative mediation". *In Third International Joint Conference on Autonomous Agents and Multiagent Systems, pages 438-445*. IEEE Computer Society.

Mizuno, K., Y. Fukui et S. Nishihara, 2008. Urban Traffic Signal Control Based on Distributed Constraint Satisfaction, *Proceedings of the 41st Hawaii International Conference on System Sciences*. Hawaii.

Roosmond D. A., 2001. using intelligent agents for proactive, real-time urban intersection control", *European Journal for Operational Research* 131. pp293-301.

Stif, 2001. Guide technique des systèmes de priorité bus aux carrefours à feux. *Rapport stif*, Paris.