Automated Negotiation for Traffic Regulation

Matthis Gaciarz, Samir Aknine, Neila Bhouri

Automated Negotiation for Traffic Regulation

Matthis Gaciarz[1], Samir Aknine[1], and Neila Bhouri[2]

[1] LIRIS - Université Claude Bernard Lyon 1 - UCBL
69622 Villeurbanne Cedex - FRANCE
matthis.gaciarz@liris.cnrs.fr
samir.aknine@univ-lyon1.fr
[2] IFSTTAR/GRETTIA, "Le Descartes 2", 2 rue de la Butte Verte
93166 Noisy Le Grand Cedex - FRANCE
neila.bhouri@ifsttar.fr

Abstract. Urban congestion is a major problem in our society for quality of life and for productivity. The increasing communication abilities of vehicles and recent advances in artificial intelligence allow new solutions to be considered for traffic regulation, based on real-time information and distributed cooperative decision-making models. The paper presents a mechanism allowing a distributed regulation of the right-of-way of the vehicles at an intersection. The decision-making relies on an automatic negotiation between communication-equipped vehicles, taking into account the travel context and the constraints of each vehicle. During this negotiation, the vehicles exchange arguments, in order to take into account various types of information, on individual and network scales. Our mechanism deals with the continuous aspect of the traffic flow and performs a real-time regulation.

Keywords: Urban traffic control, regulation, negotiation, cooperative systems, intersection, multi-agent system

1 Introduction

Various traffic control methods have been developed in the last decades in order to optimize the use of existing urban structures. As the intersection is a conflict zone causing important slowdowns, most urban traffic control systems focus on the intersection regulation, optimizing the right-of-way at traffic lights. Artificial intelligence enabled to investigate new methods for traffic modeling and regulation, especially with multi-agent technologies that are able to solve various problems in a decentralized way [6]. Today's communication technology enables the design of regulation methods based on real-time communication of accurate information. Each vehicle on a network has a traffic context, and the information that constitutes this context can be useful to perform an efficient regulation: the accumulated delay since the start of the vehicle's journey, its current position, its short and long-term intentions, etc.

Due to the large amount of information, some strategies regulate the traffic on isolated intersection [12]. Some strategies are network-wide control [16] and others focus on the coordination on several intersections creating what is called "green waves" [10]. Green wave reduces stops and gos that cause important time losses. The efficiency of this phenomenon in classical regulation highlights the importance of designing mechanisms enabling coordination at the scale of several intersections. [12] proposes a right-of-way awarding mechanism based on reservation for autonomous vehicles. It relies on a policy called FCFS (First Come First Served), granting the right-of-way to each vehicle asking for it, as soon as possible. This mechanism allows to take into account human drivers by using a classical traffic light policy for human drivers, and giving the right-of-way on red lights to automatic vehicles using the FCFS policy. Although this mechanism accommodates human drivers, its main benefits are due to the FCFS policy and the presence of autonomous vehicles.

In this paper, we propose a different right-of-way awarding mechanism on the intersection scale and tackle two complementary aspects. Firstly, we take into account the traffic context in order to make accurate decisions: the global context (network scale information) and the individual context of each vehicle (history, current information, intentions) are useful information that can be used to produce a fair and efficient regulation policy. Secondly, to have a distributed decision, the vehicles make the decision by themselves in order to deal with the large amount of information. To achieve these goals, we propose a regulation method based on an automatic negotiation mechanism, supported by intelligent agents representing the vehicles' interests. Our mechanism has to bring the vehicles to reach a collective decision in which each vehicle can put forward its individual constraints, suggest solutions and take part in the final decision in real time. Such right-of-way awarding mechanism has to efficiently take into account both autonomous vehicles and human drivers in a communication-equipped vehicle. A fundamental part of our research consists in the conceptualization of multilateral interactions in terms of individual and collective interests. This paper shows a possibility to take some steps towards new foundations of interactions. Based on this, we propose a new negotiation framework for an agent-based traffic regulation and tackle the continuous aspect of the traffic flow. In such negotiations, vehicles build various right-of-way awarding proposals that we call "configurations". These configurations are expounded to the other vehicles of their area, that can raise arguments about the benefits and drawbacks of each configuration. The vehicles decide on the configuration to adopt collectively, with the help of the intersection that contributes to the coordination of the interactions.

The remainder of this article is organized as follows. Section 2 presents the intersection model we opted for, and the problem of right-of-way awarding as it stands at an intersection. Section 3 details the method used by agents to build configuration proposals while turning the problem into a CSP (Constraint Satisfaction Problem). Section 4 presents the negotiation mechanism enabling the vehicles to make a collective decision from their individual configuration proposals. It introduces the continuity problem and we detail how the agents tackle it. Section 5 gives the experimental results. Finally, section 6 explores future directions and concludes the paper.

2 Problem description and intersection modeling

The problem we are concerned with in this paper is to allocate an admission date to each vehicle arriving at an intersection. This date is defined as a time-slot during which the vehicle has the right-of-way to go into the intersection and cross it. A configuration has to enable an efficient traffic and respect various physical and safety constraints, taking the individual travel context of the vehicles and the global traffic context into account. An agent-based model is used where vehicles and intersections are the agents. The physical representation of the network consists in a cellular automaton model. Cellular automaton models are widely used in literature because they keep the main properties of a network while being relatively simple to use [7]. The intersection is composed of several incoming lanes, called "approaches", and a central zone called "conflict zone". We call "trajectory" the path of a vehicle across the intersection. Each approach and each trajectory is a succession of cells (cf. Figure 1). A cell out of the conflict zone belongs to exactly one approach. A cell in the conflict zone may belong to one or several trajectories. In this case, this cell is called a "conflict spot".

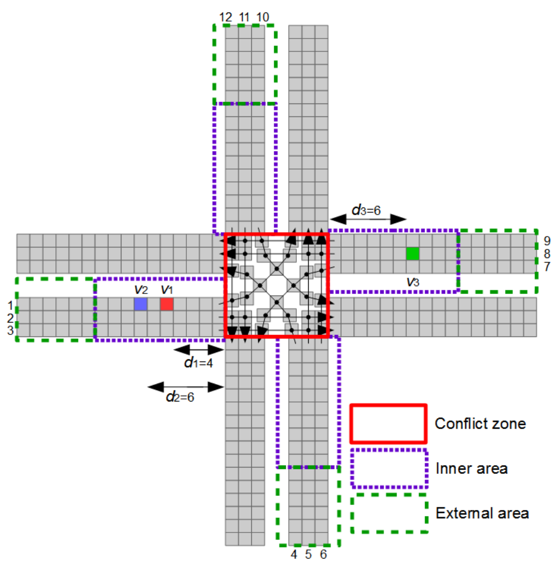

Fig. 1. Intersection with 12 approaches and 12 outcoming lanes, divided into cells. The approaches are numbered from 1 to 12. The conflict zone is crossed by various trajectories, also divided in cells. The cells of the conflict zone are conflict spots. Colored cells are vehicles, e.g. v_1 on the approach 1 is a vehicle coming from the west, about to cross the intersection to the north.

The moving rules of the vehicles are: (1) If a vehicle is on the front cell of an approach, this vehicle moves one cell forward and drives into the intersection (the first cell of its trajectory) if and only if it has the right-of-way. (2) If a vehicle is on an approach, it moves forward if and only if the next cell of the approach is empty, or becomes empty during this time step. (3) If a vehicle is in the conflict zone, it necessarily moves forward. Our method has to guarantee for each vehicle that it will not meet any other vehicle in the cells of its trajectory. The decision is distributed: each vehicle agent is able to reason and communicate with the intersection and the other vehicles. To propose a mechanism enabling the vehicles to perform a distributed decision making, the agents may build partial solutions based on their individual constraints, and then merge these partial solutions. Since the admission dates making a configuration are strongly interdependent because of safety constraints, merging partial solutions would be a complex task that would require multiple iterated interactions for the agents with several messages to exchange, and would slow down the decision process. Therefore, in our approach the vehicles build individually full configurations of the intersection and then collectively deliberate on these configurations.

3 Modeling the right-of-way allocation problem to build configurations

In order to build configurations, we model the right-of-way allocation problem as a Constraint Satisfaction Problem (CSP) [13]. The CSP fits our problem since it is easy to represent its structural constraints (physical constraints and safety constraints). Let V be the set of all vehicles approaching an intersection, and t_{cur} be the current date in time steps. A configuration is a set $c = \{t_1, ..., t_k\}$ where each t_i is the admission date in the conflict zone accorded to $v_i \in V$. For each $v_i \in V$, app_i is the approach on which is v_i, d_i the distance (in number of cells) between v_i and the conflict zone, $traj_i$ is v_i's trajectory inside the conflict zone. T is the set of all the trajectories inside the conflict zone. $pos(cell_1, traj)$ is the distance, in number of cells, between the cell $cell_1$ and the beginning of the conflict zone on the trajectory $traj$ (the first cell in the conflict zone has the position 0). sp is the speed of the vehicles in cells by time step. In our model, $sp = 1$ cell/time step. We identify 3 types of structural constraints for vehicles, based on the following rules:

R1. Distance rule A vehicle has to cross the distance separating it from the conflict zone before entering it. We have: $\forall v_i \in V, t_i > t_{cur} + \frac{d_i}{sp}$

R2. Anteriority rule A vehicle cannot enter the conflict zone before the vehicles preceding it on its lane (this rule could be removed with a more complex model that would take overtaking into account). We have:
$\forall v_i, v_j \in V^2, app_i = app_j, d_i < d_j \Rightarrow t_i < t_j$

R3. Conflict rule Two vehicles cannot be in the same cell at the same time. If the vehicles belong to the same lane or trajectory, the moving rules prevent this case. However, if a cell is a conflict point then we have to model this rule for the vehicles belonging to different trajectories. In a basic version, we have: $\forall v_i, v_j \in V^2, \forall cell_1 \in traj_i, cell_1 \in traj_j \Rightarrow (t_i + \frac{pos(cell_1, traj_i)}{sp}) \neq$

$(t_k + \frac{pos(cell_1,traj_j)}{sp})$. This rule must be reinforced for safety reasons. Indeed, adding a time lapse t_{safe} between the passage of a vehicle on a cell $cell_1$ and the passage of a vehicle in a conflicting trajectory on this cell enhances the drivers' safety (t_{safe} is fixed by an expert). The complete conflict rule is the following:
$\forall v_i, v_j \in V^2, \forall cell_1 \in traj_i, cell_1 \in traj_j \Rightarrow$
$\left|(t_i + \frac{pos(cell_1,traj_i)}{sp}) - (t_k + \frac{pos(cell_1,traj_j)}{sp})\right| > t_{safe}$

A configuration c is valid iff c respects the three rules R1, R2 and R3 and: $\forall v_i \in V, \exists t_i \in c$, where each t_i is v_i's admission date. The scenario represented in Figure 1 illustrates these three types of structural constraints. Let's consider the three vehicles v_1, v_2, v_3 approaching the intersection at $t_{cur} = 0$. The above rules generate the following 6 constraints:
- **R1** (ct_1) $t_1 > 4$; (ct_2) $t_2 > 6$; (ct_3) $t_3 > 6$
- **R2** (ct_4) $t_2 > t_1$
- **R3** (ct_5) $|(t_1 + 4) - (t_3 + 2)| > 2$; ($ct_6$) $|(t_2 + 4) - (t_3 + 2)| > 2$

With this CSP model, an agent uses a solver to find compatible admission dates (i.e. respecting the above constraints) for a set $V^{neg} \subseteq V$ of vehicles approaching an intersection. For any configuration c, $\forall v_i \in V^{neg}, \exists d_i \in c$ such as d_i respects the above structural constraints. Several possible configurations may exist for a given situation. A vehicle initially has limited perceptions, however it is able to know in real-time the position of the vehicles around the intersection. As this work conforms the cooperative approach of intelligent transportation systems ([2],[9]), each vehicle has a cooperative behavior with the intersection and communicates its trajectory when it enters the approach of the intersection. With its computation abilities and the available information, a vehicle runs a solver to produce configurations. The use of an objective function enables to guide the CSP solver's search. Moreover, an agent can add additional constraints to its solver as guidelines. If an agent estimates that a particular constraint may produce configurations likely to improve its individual utility or social welfare, this agent considers adding it. However, since this constraint is not a structural constraint resulting from the above rules, it may be violated. The chosen objective function and these potential guideline constraints depend on each vehicle agent's strategy. A configuration built in this manner may satisfy different arguments than the other configurations, and this may be useful in the negotiation to make it chosen.

4 Right-of-way negotiation model

Each vehicle builds configurations allowing it to cross the intersection, however only one configuration will be applied at a given moment. A negotiation process takes place to select it. The mechanism we propose relies on an argumentation-based model [5]. Through the negotiation process, agents aim to reach a collective agreement by making concessions. To perform a negotiation, the vehicle agent relies on its own mental state, made of knowledge, goals and preferences. This mental state evolves during the negotiation. The agents use arguments to make the other agents change their mental states, in order to reach a better compromise. Each agent a_i has the following bases: $\mathcal{K}_i$ is the knowledge base of a_i about its environment. Its beliefs are uncertain, so each belief $k_i^j \in \mathcal{K}_i$ has a certainty level ρ_i^j. $\mathcal{KO}_i$ is the knowledge base of a_i about other vehicles. Each $ko_i^j \in \mathcal{KO}_i$ is a base containing what a_i's believes the knowledge of a_j are. Each of these beliefs has a certainty level δ_i^j. $\mathcal{G}_i$ is the goal base of a_i. These goals have various priority, so each goal $g_i^j \in \mathcal{G}_i$ has a priority level λ_i^j. $\mathcal{GO}_i$ is a_i's base of supposed goals for other vehicles. Each $go_i^j \in \mathcal{GO}_i$ is a base containing what a_i's believes the goals of a_j are. Each of these beliefs has a priority level δ_i^j. Each vehicle has a weight given by the intersections, as detailed in the next section. Two kinds of arguments may be used by the agents, favorable and unfavorable arguments. An argument for (resp. against) a configuration decision d is a quadruple $A =< Supp, Cons, d, w_A >$ where $Supp$ is the support of the argument A, $Cons$ represents its consequences, w_A is the weight of the argument (fixed by the vehicle v_i that produces this argument and has a weight w_i), such that:

- $d \in \mathcal{D}$, $\mathcal{D}$ being the set of all possible decisions
- $Supp \subseteq \mathcal{K}^*$ and $Cons \subseteq \mathcal{G}^*$
- $Supp \cup \{d\}$ is consistent
- $Supp \cup \{d\} \vdash Cons$ (resp. $\forall g_i \in Cons, Supp \cup \{d\} \vdash \neg g_i$)
- $Supp$ is minimal and $Cons$ is maximal (for set inclusion) among the sets satisfying the above conditions.
- $0 \leq w_A \leq w_i$

Example: A bus b_1 proposes a configuration c_1 allowing it to cross the intersection as quick as possible to catch up its lateness. A vehicle v_1 precedes this bus on the same lane. Giving a quick admission date to b_1 (below a fixed threshold t^b_{quick}) implies to give a quick admission date to v_1(below a fixed threshold t^v_{quick}), and one of the goals of v_1 is to cross the intersection as quick as possible. Thus:

$\mathcal{K}_{v_1} = \{crossesQuickly(b_1) \rightarrow crossesQuickly(v_1)\}$
$\mathcal{G}_{v_1} = \{crossesQuickly(v_1)\}$
v_1 may take advantage of this configuration, so it produces the following argument:
$< \{crossesQuickly(b_1), crossesQuickly(b_1) \rightarrow crossesQuickly(v_1)\},$
$\{crossesQuickly(v_1)\}, c_1 >$.

For safety reasons, the intersection has a current configuration at any time. The goal of an agent through the negotiation is to change this current configuration c_{cur} by another c_{best} that improves its individual utility. In a negotiation the agents rely on a communication language to interact. The set of possible negotiation speech acts is the following: $Acts = \{Offer, Argue, Accept, Refuse\}$.

Offer(c_{new}, c_{cur}): with this move, an agent proposes a configuration c_{new} to replace c_{cur}. An agent can only make each offer move once.

Argue$(c, arg(c))$: with this move, an agent gives an argument in favor of c or against c.

Accept(c_{new}, c_{cur}), **Refuse**(c_{new}, c_{cur}): with these moves, an agent accepts (resp. refuses) a configuration c_{new} to replace c_{cur}.

c_{new} is accepted iff $\frac{\sum_{v_i \in V(c_{new})} w_i}{\sum_{v_i \in V^{neg}} w_i} \geq th_{accept}$, where:
th_{accept} is an acceptance threshold ($th_{accept} > 0.5$).
$V(c_{new}) \subseteq V^{neg}$ is the set of vehicles accepting the configuration $c_{new} \in \mathcal{D}$ to replace c_{cur}. w_i is a weight given by the intersections to the vehicle v_i. When a configuration is adopted by the agents, this configuration becomes the current configuration of the intersection.

4.1 Role of the intersection agent

In order to perform a right-of-way allocation that maximizes the social welfare and encourages cooperative behaviors, the intersection agent takes part in the negotiation process. Each vehicle first defends its own interests, and also defends other interests that may guide the negotiation towards a favorable outcome for it. A vehicle can represent the interests of other vehicles outside V^{neg} (for example the vehicles that follow it) or network scale interests (for example clearing some lanes) if it can get advantage of it. However, it may happen that these arguments do not directly concern the vehicles of V^{neg}, that may ignore these arguments despite their positive contribution to global social welfare. To avoid this effect, the intersection agent is able to represent these external interests. Like the vehicle agents, the intersection agent has its own mental states and is able to produce arguments. However, it cannot accept or refuse proposals.

The weight the intersection agent gives to each of its arguments depends on the importance of the external interests represented by these arguments. A weight w_i of a vehicle v_i is given by the intersection agents to encourage the vehicles to have cooperative behaviors. According to v_i's cooperation level in its negotiation behavior, the intersection increases or decreases w_i for the remainder of v_i's journey. A vehicle refusing a proposal having numerous strong arguments for it (or accepting a proposal having numerous strong arguments against it) gets an important weight penalty. On the contrary, a vehicle accepting a proposal having numerous strong arguments for it (or refusing a proposal having numerous strong arguments against it) gets a weight reward. For a vehicle, these rewards and penalties are significant in the middle and long term since it affects durably its capacity to influence the choice of the configurations on the next intersections. To perform this, the intersection uses arguments to assign a reward (or penalty) value to each proposal, so that the vehicles may evaluate the benefits and risks from each decision about configurations before making it.

4.2 Continuous negotiation mechanism

Since the flow of vehicles is continuous, the mechanism has to manage this dynamic aspect by defining the agents that take part in each negotiation step, the vehicles for which this configuration provides an admission date, and the conditions under which this configuration could be revised once chosen. In order to manage technical failures, the intersection has a current configuration c_{cur} at any time. According to the chosen continuity policy, the negotiation mechanism may allow the vehicles to collectively change this configuration. However, the mechanism has to consider safety measures before allowing this change. Changing the configuration at the last moment is risky because of the slowness of the reaction of the drivers. To avoid this, we define a safety time threshold th_{safe}. The admission date of a vehicle cannot be revised (removed or granted) in a too short term. Let t_i^{cur} be the admission date of vehicle v_i in the current configuration and t_i^{next} be its admission date in a configuration c. c is an eligible proposal iff c is valid and: $\forall v_i \in V^{neg}, (t_i^{cur} = t_i^{next}) \vee ((t_i^{cur} \geq t_{cur} + th_{safe}) \wedge (t_i^{next} \geq t_{cur} + th_{safe}))$

We propose several policies to manage the continuity problem. First, we distinguish two areas on the approaches of the intersection: the inner area, where all the vehicles are about to reach the conflict zone in a short term, and the external area, where the agents will reach the conflict zone in a slightly longer term (cf. Figure 1). The size of each area depends on the intersection. At each time step t_i, the set V_i of the incoming vehicles is divided in two subsets: V_i^{inn} the vehicles of the inner area and V_i^{ext} the vehicles of the external area. $V_i = V_i^{inn} \cup V_i^{ext}, V_i^{inn} \cap V_i^{ext} = \emptyset$

Let $\mathcal{T}$ be the period allowed for the negotiation. Let Δ^{ref} be the threshold which is the maximum number of $Refuse$ that an agent can send and δ_i^{ref} the number of $Refuse$ an agent v_i has sent during $\mathcal{T}$. If $\delta_i^{ref} = \Delta^{ref}$, v_i cannot do any $Offer$ or $Refuse$ move. Let Δ^{arg} be the threshold which is the maximum number of $Argue$ that an agent can send and δ_i^{arg} the number of $Argue$ an agent v_i has sent during $\mathcal{T}$. If $\delta_i^{arg} = \Delta^{ref}$, v_i cannot do any $Argue$ until the end of $\mathcal{T}$. An agent can only make each offer once during a negotiation. Once an agent has made the move $Offer(c_x, c_y)$ during $\mathcal{T}$, it cannot make it again during the negotiation. We get the following set of rules.

- **NR1:** $\forall v_i \in V^{neg}$, the move $Offer(c_x, c_y)$ can be made at any time by v_i if this move has not been made yet by v_i during $\mathcal{T}$ and if $\delta_i^{ref} < \Delta^{ref}$.
- **NR2:** $\forall v_i \in V^{neg}$, the move $Accept(c_x, c_y)$ can be made at any time by v_i. Furthermore, the move $Offer(c_x, c_y)$ was made at time $t_0 \in \mathcal{T}, t_0 < t$.
- **NR3:** $\forall v_i \in V^{neg}$, the move $Refuse(c_x, c_y)$ can be made at any time $t \in \mathcal{T}$ by v_i if $\delta_i^{ref} < \Delta^{ref}$. Furthermore, the move $Offer(c_x, c_y)$ was made at time $t_0 \in \mathcal{T}, t_0 < t$.
- **NR4:** $\forall v_i \in V^{neg}$, the move $Argue(c_x, arg(c_x))$ can be made at any time $t \in \mathcal{T}$ by v_i if $\delta_i^{arg} < \Delta^{arg}$. Furthermore, the move $Offer(c_x, c_y)$ was made at time $t_0 \in \mathcal{T}, t_0 < t$, for any $c_y \in \mathcal{D}$.

Iterated Policy (IP) With this policy, the vehicle agents join the negotiation by waves, and perform iterated decisions that cannot be revised. At a given instant t_{i-1}, V^{inn} is empty. At the next time step t_i, since the vehicles have moved, V^{inn} and V^{ext} change. The set of negotiating vehicles V_i^{neg} becomes equal to V_i^{inn}. Then the vehicles of V_i^{neg} perform a collective decision about the configuration for all the vehicles of V_i^{neg}. A negotiation process starts, with a limited duration d_{neg} in addition to the above set of rules. $\mathcal{T} = [t_0^{neg}, t_0^{neg} + d_{neg}]$, where t_0^{neg} is the starting date of the negotiation. With this limited duration, the agents have interest to quickly make reasonable proposals for every vehicle. At the end of this negotiation step, a configuration c_i is chosen, awarding an admission date to each vehicle of V_i^{neg}.

At t_{i+1}, a new iteration begins, and $V_{i+1}^{neg} = V_{i+1}^{inn} \setminus V_i^{neg}$. The vehicles of V_{i+1}^{neg} start a new negotiation, but the vehicles that already have taken part in a previous negotiation step do not take part in this one. The agents of V_{i+1}^{neg} are not allowed to revise c_i, the agents only negotiate the admission dates of the vehicles of V_{i+1}^{neg} since the other vehicles of V_i^{inn} already have an admission date defined in c_i or in previous configurations. A new configuration c_{i+1} is chosen, similar to c_i except it adds admission dates for the vehicles of V_{i+1}^{neg}. $c_i \setminus c_i^{out} \subseteq c_{i+1}$ where c_i^{out} is the set of the vehicles admitted in the conflict zone: $\forall t_j \in c_i, t_j < t_i \Leftrightarrow t_j \in c_i^{out}$.

The policy continues to iterate and to produce new admission dates for the next vehicles in the inner area without revising those of the vehicles that already were in it.

Continuous Policy (CP) When this policy is applied the vehicles dynamically join the current negotiation while entering the inner area, $V^{neg} = V^{inn}$ at any time. When a vehicle v_{new} joins V^{inn}, all the useful information about the current state of the negotiation (configurations and arguments) are communicated to v_{new} so that it can join the negotiation. The current configuration of the intersection can be totally revised by a collective decision, except for the vehicles that are concerned by the security threshold.

Whenever new vehicles join V^{inn}, the current configuration of the intersection and the configurations under negotiation do not provide admission dates for these vehicles, since the configurations were emitted before these vehicles joined V^{inn}. However, the intersection provides an ordering on these vehicles. With this ordering, it is possible for any vehicle in the negotiation to extend any of the vehicles' configuration proposal. Extending a configuration consists in adding an admission date for each new vehicle with the FCFS strategy, using the ordering on these vehicles. The agents consider that any proposal in the negotiation that do not provide an admission date to each vehicle of V^{inn} will be extended with FCFS. It guarantees that the intersection always has an admission date for each vehicle of V^{inn}. Thus, even if the negotiation always fails, the FCFS policy is applied.

A possible perspective is to extend CP with a new policy CPA (Continuous Policy with Anticipation). In CP, when a vehicle builds a configuration, this configuration only incorporates vehicles of V^{inn}. In CPA, each vehicle $v_1 \in V^{neg}$ can take into account any other vehicle from $v_2 \in V^{ext}$ while building configurations, in order to take advantage of it. Then, whenever v_2 joins V^{inn}, some proposals (including the current configuration of the intersection) may already include an admission date for it. According to the result of the previous negotiations these configurations may be better than the one produced by the FCFS strategy.

5 Experimentation and discussion

This work has been implemented in Java with the Choco library for CSP [8], on an intersection with 12 approaches (cf. Figure 1). The length of the inner area is 6 cells on each approach. Agents are implemented as threads: each agent has its own solver and its own negotiation strategy. The agents communicate with other agents with direct messages. On a personal computer (RAM 2Gb, 1.9 GHz mono-core processor), 2 seconds are enough to run the solver and compute several good configurations for about 30 vehicles, and the negotiation time is low enough to enable to run the mechanism in real time. In this section, we present the results of the comparison between FCFS and the CP policy. We simulated a continuous incoming flow of vehicles (1.2 vehicle/step in average). Vehicles appear on a randomly chosen lane. We chosed to apply $t_{safe} = 2$. These simulations were performed on a more computer with RAM 32Gb, 64-core processor. Results are shown on Figures 2 and 3. These figures respectively represent the number of vehicles in the intersection area and the average number of vehicles waiting for the right of way on each approach, relatively to the time. For example in simulations of the CP policy, after 100 time steps the average number of vehicles in the area were 37.9 (cf. Figure 2) and 0.64 vehicles were waiting for the right of way on each approach of the intersection (cf. Figure 3).

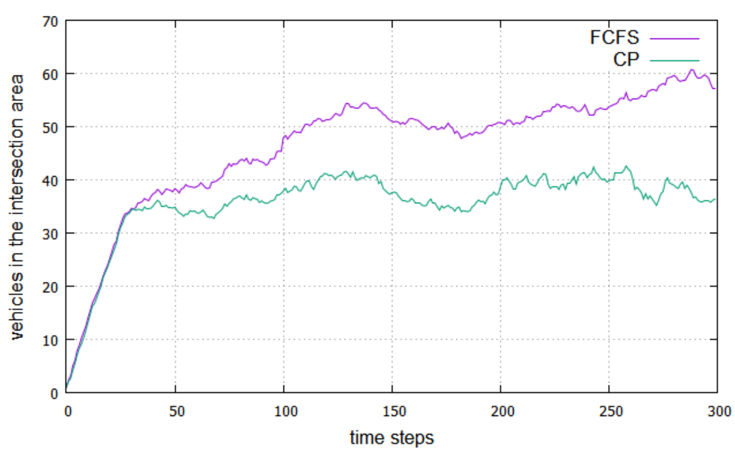

Fig. 2. Number of vehicles in the area

The main improvements of our negotiation-based mechanism are expected to appear on the network scale, and so far we only experimented it on a single

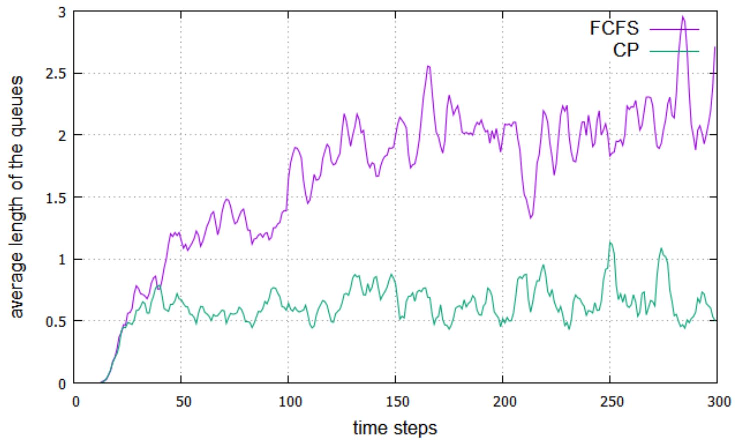

Fig. 3. Average length of the queues

intersection. The main goal of these early experiments, and our main result, is to show the feasability of this mechanism. The slight performance improvements shown on Figures 2 and 3 may also be explained by the use of the solver to optimise the right-of-way of the vehicles. Moreover, this improvement is accentued with the use of the safety time lapse t_{safe} defined in the conflict rule (R3) that gives more importance to the ordering of the vehicles.

6 Conclusion

In this paper, we have proposed a coordination mechanism which represents a large step towards easing traffic, minimizing time losses while respecting safety constraints. The contribution of this paper is threefold. Firstly, it defined the problem of intelligent agent-based intersection management. Secondly, it presented a negotiation mechanism that deals with continuous negotiations and applies a set of policies, and behavior rules that show how to exploit this framework over intersection control methods. Finally this paper suggested that it is both algorithmically feasible and reasonable in terms of delay and computational cost to enable such sophisticated reasoning. Thus, this paper shows the possibility to make one step forward towards a system that can take action to manage the decision of the vehicles cooperatively.

However, substantial work must still be done. For example, a possible direction concerns the intersection agent that can switch among several policies, for instance by learning from the reservation history to find the best policy suited to particular traffic conditions. In current work we are adapting the behavior of the intersection to handle vehicle priorities.

References

1. A. Abbas-Turki et al. Cooperative intersections for emerging mobility systems. In *15th Euro Working Group on transportation*, 2012.
2. A. Abbas-Turki, F. Perronnet, J. Buisson, A. El-Moudni, M. Ahmane, R. Zéo, et al. Cooperative intersections for emerging mobility systems. In *15th Euro Working Group on transportation*, 2012.
3. J. L. Adler et al. A multi-agent approach to cooperative traffic management and route guidance. *Transportation Research*, 2005.
4. J. L. Adler, G. Satapathy, V. Manikonda, B. Bowles, and V. J. Blue. A multi-agent approach to cooperative traffic management and route guidance. *Transportation Research Part B: Methodological*, 39(4):297–318, 2005.
5. L. Amgoud, S. Belabbes, and H. Prade. Towards a formal framework for the search of a consensus between autonomous agents. In *AAMAS*, 2005.
6. A. L. C. Bazzan and F. Klügl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 2013.
7. E. Brockfeld, R. Barlovic, A. Schadschneider, and M. Schreckenberg. Optimizing traffic lights in a cellular automaton model for city traffic. *Physical Review E, 64*, 2001.
8. T. choco team. choco: an open source java constraint programming library. 2010.
9. M. C. Choy, D. Srinivasan, and R. L. Cheu. Cooperative, hybrid agent architecture for real-time traffic signal control. *IEEE SMC*, 33(5):597–607, 2003.
10. D. de Oliveira, A. Bazzan, and V. Lesser. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 463–470. ACM, 2005.
11. K. Dresner and P. Stone. Sharing the road : autonomous vehicles meet human drivers. In *IJCAI*, 2007.
12. K. Dresner and P. Stone. A multi-agent approach to autonomous intersection management. *Journal of artificial intelligence research*, 2008.
13. V. Kumar. Algorithms for constraint-satisfaction problems: A survey. *AI magazine*, 13(1):32, 1992.
14. S. Maerivoet and B. De Moor. Cellular automata models of road traffic. *Physics Reports, 419*, 2005.
15. J. Monteil, R. Billot, and N.-E. El Faouzi. Towards cooperative traffic management: methodological issues and perspectives. In *Proceedings of Australasian transport research forum 2011 proceedings, Adelaide, Australia*, 2011.
16. D. Roozemond. Using intelligent agents for proactive, realtime urban intersection control. *European Journal of Operational Research*, 2001.