A Multi-Agent System to Regulate Urban Traffic: Private Vehicles and Public Transport

Neïla BHOURI, Sofiane HACIANE and Flavien BALBO

Abstract—This paper proposes a strategy of urban bi-modal traffic regulation based on multi-agent modelization. The objective of this strategy is to work on the duration of the traffic lights to regulate traffic for transport modes that include private and public transport vehicles, principally the bus. The model respects the method of bus regulation with a route agent, who supervises all the vehicles on the route while ensuring that they respect the regular time intervals between them. Bus priority is granted by reserving a green stage for buses as soon as they enter the route in question. This stage is prioritised for those buses running late, and whose priority does not lead to deterioration in the regularity of the time intervals between them and the preceding vehicle; an interval they must respect at bus stops. Regulation is obtained thanks to communication, collaboration and negotiation between the agents; the programming is developed under the JADE (Java Agent Development Framework) platform.

I. INTRODUCTION

Several cities employ regulation systems at junctions that grant priority to vehicles, the aim of which is to improve the route times of public surface transport (bus, tramways, shuttles, etc.), which are referred to as systems equipped with bus priority. The aim of these strategies is to increase the average speed taken to cross the junction for all vehicles as well as public transport vehicles.

The use of these systems is efficient when traffic is light or when used to benefit a single congested bus route. Studies show that the use of bus priority can produce a time saving of between 20% and 40% on the global bus journey [1]. However, reducing the time of the bus journey, although very important for the global bus journey [1]. However, reducing the time of the bus journey is not the primary factor considered by public transport operators whose obligation is to provide a service for passengers.

Furthermore, for the public transport network operators, regulation is a process of real time adaptation of the bus schedule to the operational conditions. This bus schedule comprises the sum of missions, each mission being the sum of journeys (timed trips) by a vehicle and a driver during one day of operation. One of the objectives of regulators consists of respecting, as much as possible, the theoretical timetables [2]. However, in the case of heavy congestion, and to respect the objectives based on the elaboration of the theoretical bus schedule, the regulator functions by real time modification.

Depending on the time of day and the place of the intervention, the regulators follow one of the three different logical principles of public transport regulation. For example, the logic of punctuality is often used when the frequency of passing public transport vehicles is low; the vehicle must therefore respect, as much as possible, the times of the theoretical bus schedule. When traffic is heavy or the frequency of passage along the route is high, the trips are carried out with the logic of mounting the load. Consequently, the onus is not on the difference between the theoretical time and real time, but on the intervals between vehicles. Near schools or theatres and at night the logic of mounting the load is favoured. The focus of this logic is to mount passengers without leaving them at bus stops.

This work concentrates on conditions of very heavy circulation; traffic is dense and the frequency of passage on bus routes is high. In addition, we are assuming that several bus routes are using the network; we are therefore concerned with regulating global traffic, to favour the passage of buses and to supervise the regularity of intervals at bus stops between vehicles along the same bus route.

The problem of global traffic regulation that favours buses has been studied by [3] and [4], however, in the research, the regulation of intervals was only indirectly treated such that the problematic was to reduce the time spent in traffic jams so that the bus adheres to its bus schedule. [5] studied the problem of global traffic regulation with, in addition, the regulation of intervals between buses. The work carried out described the dynamic of the system in a macroscopic way for the general traffic, and in a microscopic way for the bus. A multi-objective optimization had been applied to the system by using the method of Particle Swarm Optimization (PSO). The complexity of the model suggests that we can obtain better results with a system of multi-agents. Our research aims to test this.
Application of classic control theories to regulate the bi-mode traffic (private vehicles and public transport) is confronted to the bi-mode modelization problem. Because microscopic modelling is time-consuming, it is not well adapted to build real time control strategies for wide urban networks. Private vehicles traffic can be considered as a continuous flow and represented by a macroscopic model. However it is difficult to consider public transport traffic as a continuous flow. Indeed, even when there is a high number of public transport vehicles on a network link, these vehicles are often due to different public transport routes and cannot be considered as a same traffic flow. Multi-agent modelization can be a suitable answer to this scale problem.

The application of a multi-agent system is increasingly present in the field of traffic regulation. For example, [6], [7],[8] and [9] resolved the problem of traffic lights coordination on the thoroughfares of the route network. [10], [11] and [12] tackled the issue of how to control traffic lights at junctions in order to improve circulation. However, we are not aware of any application for the regulation of bi-modal traffic, which is the object of our research.

The following section of this article describes the modelization of the network and the identification of the agents for the modelization. The second section provides a detailed description of agents, their attributes, their objectives, as well as the communication and collaboration between them. The fourth section provides details of the network and the results of the simulation tests carried out. Finally, we conclude in the fifth section.

II. NETWORK MODELIZATION

A. Network physical modelization

The urban network is modelled by a graph oriented \( G = (I, A) \). The nodes (I) represent the junctions (or intersections) and the arcs (A) represent the lanes that connect the junctions. Two intersections can be connected by one or several arcs depending on the number of lanes on the thoroughfare.

![Figure 1: A junction with 4 arcs and two stages](image)

**Arc**: an arc corresponds to a lane. It is characterized by a sum of static and dynamic information, notably the list of successive arcs on which the vehicles can travel, its length \( L \) (in metres), its capacity \( C \) (in private car unit, PCU). Its saturation output, which is the maximum output of exists from arc \( D \) (in PCU/second), the number of vehicles present on the arc, \( N_i \) (in PCU), and finally, the state of the traffic lights at the extremity of the arc: green or red. If the light is green then the vehicles present on the arc can depart.

**Junction**: a junction is specified by the sum of the arcs that enter it (E) and the sum of the arcs that leave it (S). A junction is managed by a sum of stages \( P = \{P_1, P_2, ..., P_n\} \). Each of the stages specifies the list of arcs for which the green light is awarded if the stage is active.

The following figure illustrates an example of a two-stage junction \( P=\{P_1, P_2\} \). P1 allows for the clearing of the arcs a1 and a3. P2 clears arcs a5 and a8. The entries and exits of the junction are respectively \( E=\{a1, a3, a5, a8\} \) and \( S=\{a2, a4, a7, a6\} \).

The network is used by a number of bus routes. Each route comprises the sum of buses of the same origin and in the same direction, and which services a number of predefined commercial bus stops at regular time intervals between the different buses on the route. As we shall see later, the time spent by a bus at a commercial stop will be equal to the pre-set time for passengers to mount, plus additional time to regulate the interval, if required.

B. Identification of agents

In order to identify agents, it is make an abstraction of the real system; for every entity of the real world is associate an entity (agent) in the virtual world. All these entities interact among themselves and form a Multi-Agent System (MAS). Homogenous agents are called “agent-type” or “agent-class”. The developed MAS is than made up of the following agent-types:

**Junction Agent (JA)**: is the key agent of our architecture. It is in charge of controlling a junction with traffic lights, and to develop a traffic light plan.

**Stage Agent (SA)**: the traffic lights plan is elaborated thanks to the collaboration of the junction stage agents. Each stage agent is expected to determine the optimal green light split to clear the waiting vehicles on the arcs concerned by the stage. Thus, whatever the complexity of the junction (and its physical configuration), it is seen as a sum of stage agents interacting with the junction agent in order to develop a plan of action for the traffic lights.

**Bus Agent (BA)**: represents a bus in the real world. It circulates from one arc to another, halts at commercial stops, halts at red lights and obeys the instructions of the route agent. The objective of each bus agent is to minimize the time spent at traffic lights (i.e. to minimize journey times).
**Bus Route Agent (BRA):** the bus agent (BA) only provides a local overview of their environment and, in particular, only the journey covered by the BA. Thus, local optimization carried out by bus agents can have a negative impact on the route, notably on its regularity (i.e. the formation of bus queues). To tackle this problem, we propose an agent who has a global overview of the route agents, and who can control and modify their behavior in order to guarantee an efficient and regular service.

C. Description of agents

This section provides a description of the internal architecture of the agents.

1) **Bus Agent**

In order to minimize the time spent at traffic lights the bus agent interacts with junction agents and the hierarchical superior (route agent). All the buses have to provide a regular service and avoid bus queues, in other words, the frequency of buses passing commercial stops must remain stable. To achieve this latter objective, the bus agent can receive orders from the route agent (for example, stay at the stop for t seconds, if the bus is ahead with respect to the position of the preceding bus).

The bus agent is comprised of a data module, which represents its internal state, and a communication module, which enables exchange with other system agents.

**Behavior of the bus agent**

The bus agent is introduced on the route at instant \( t_0 \), the agent behaves in the following way:

- On entering each arc, the bus agent retrieves information from the arc (the number of vehicles that precede it, the length, capacity, and exit output of the arc). By using the data, the bus agent calculates a time-space reservation, which is transmitted to the junction agent in order to prevent an eventual stop at the red light at the following junction. The junction agent then attempts to satisfy the demand (see junction agent);

- when approaching a stop, the bus agent informs the associated route agent. The route agent then calculates the duration of the regulation and its level of priority and sends it to the bus. The bus must wait during the passenger loading time, as well as the potential regulation time, before leaving the stop.

**Calculation of a reservation of a green light**

A reservation of a time-space for the green light is specified by the interval of time during which the green light is granted to the actual arc so that the bus can pass without stopping at the next junction.

If \( t_0 \) the actual instant, the reservation \( R \) is given by:

\[
R = [t_0, t_f]
\]

\( t_0 \) and \( t_f \) being the beginning and end moments of the reservation respectively. The calculation of these instants is carried out as follows: the bus enters the arc and finds \( N \) vehicles ahead of it, the vehicles move to the traffic lights lane to wait for the green light thus forming a queue of length \( F \) (see Figure 3).

2) **Junction agent (JA)**

A JA supervises the group of stage agents (SA), who collaborate together to establish a plan for traffic lights, which will, on the one hand, maximize the capacity of the junction and, on the other hand, attempt to satisfy, as far as possible, the reservations of the buses. The JA agent is characterized by the possession of static and dynamic data.

**Static data**

The static data represents the constraints to which the junction agent is subjected, we find:

- The sum of possible values for the cycle: the length of the cycle (in seconds) should not exceed a certain value. Generally, the cycle takes one of the following values: cycle \( \epsilon \{40, 60, 80, 100, 120, 140\} \). When the traffic is dense in one or several directions, we increase the length of the cycle. In fact, for each cycle, there is a lapse of lost time i.e. the period of orange or integral red. The integral red is a period during which all the arcs from the same junction have a red light in order to clear the centre of the junction and thus prevent accidents. This fixed period, in conformity to the architecture of the junction, does not depend on the length of the cycle, with the result that the shorter the cycle the greater the loss of time and capacity of the junction.
The stages is a task executed in staggered time by the traffic experts.

Dynamic data
- The traffic lights plan: specifies the order of the passage of stages as well as the duration of each stage. Between two successive stages a 2-second period of integral red is imposed.
- The list of received reservations: each reservation is specified as follows: R=(P, t_i, t_d, Priority). Where P_i is the stage that will allow the passage of the bus, t_i is the instant when the bus expects to arrive at the traffic lights, t_d is the instant at which the rear of the bus leaves the arc, and finally ‘Priority’ is the level of priority defined by the route agent for the bus.

Behavior of the junction agent
At the end of each cycle, the JA triggers the process to calculate the traffic lights plan for the ensuing cycle. This plan must determine the duration of the green light and the order of passage of each stage.

When the JA receives a reservation, it is recorded in the database. A reservation R= (P_i, t_i, t_d, Priority) is not taken into account the moment it’s received (t_i). The JA decides to accept or to refuse this request at the instant t_i.

The behavior of the JA is explained thanks to the following example (see Figure 4) in which the junction has a plan with four stages and two antagonistic bus routes.

In this example, t represents the actual instant; at the instant t_i the junction receives the reservation R_i which requests the extension of stage P_i up to instant t_d for a bus with a priority 2 level. At the instant t_d the junction agent receives another reservation, R_j from a more prioritized bus (Priority = 4). The latter requests the stage P_j. It is not possible to satisfy both requests (R_i and R_j) because they involve two different stages at two time intervals that overlap each other. The junction agent delays as much as possible the modification of his plan. Thus, the reservation request R_i is studied at the instant t_3 (the instant at the start) and not at the instant t_1 (received instant). This delay allows the junction agent to receive more reservations (receive R_2 at the instant t_2 in the example) and thus avoid ineffective modifications, as finally R_i is refused because reservation R_2 is more prioritized.

The modification of a traffic lights plan following a priority request by a bus can concern:
- The extension of a stage (can delay or advance a stage), without exceeding the maximal duration of a stage.
- Making a stage call: introducing a new stage to the plan.

Calculation of a traffic lights plan
The plan is calculated thanks to the collaboration of the junction agent and the stage agents. Each stage is associated with a stage agent who is in charge of calculating its duration. The junction agent functions as a manager in supervising the stage agents, who function as participants.

1) The junction agent begins by forming a group of collaborators called collab_group comprising the list of stage agents that needs to be managed.
2) The junction agent initializes the variables: C=CycleMax (140 seconds), and t=0. Variable C controls the size of the cycle calculated.
3) The junction agent sends a message to the stage agents to inform them of the launch of the protocol to calculate the traffic lights plan.
4) The junction agent sends a message request to the agents of the collab_group asking them for the time necessary to clear all the vehicles from their stages, by commencing at instant t.
5) Every agent, i, of the collab_group calculates their desired green light duration (d_i) and an index that measures the urgency of the stage (I_i), and sends them to the manager. The section on the stage agent will present the way in which parameters (d_i, et I_i) are calculated.
6) When the manager receives all the responses, the sum of i
If d > C then the manager has to resolve the conflict (i.e. the size of the cycle exceeds the maximum). Conflict resolution occurs when the modification of d, previously calculated, is d ≤ C.
7) The manager selects the most urgent stage, that is P_j, its duration is d_j:
- then sends an accept message to the stage agent responsible for operating this stage;
- then withdraws the corresponding stage agent of collab_group;
- then updates the variables C=C-d_j, t=t+d_j;
- and finally moves to 4) as long as collab_group is not empty.

Conflict resolution
When the sum of green light durations requested by the stage agents exceeds the size of the cycle, the manager must restore this sum to the maximal value of the cycle. To achieve this, the manager must negotiate with the stage agents in order to reduce this sum at minimal cost. The manager launches an appeal

Figure 4: a plan for traffic lights and its modification
to the stage agents. The agents can add a portion (\(\Delta t\)) of their green light duration in order to resolve the problem. The cost of the offer is translated by the number of bus routes penalized if the stage agent adds a quantity \(\Delta t\) to the green light duration.

1) The manager initializes the cost \(c = 1\);
2) then forwards the appeal (cfp) with the cost \(c\) to all the stage agents of the associated junction (see figure 5);
3) the stage agents reply with a propose or refuse message (the modalities are explained in the section on the behavior of the stage agent).
4) Once all the responses have been received by the manager, sum of all the offers are added, such that:

\[ S = \sum_{i=0}^{\Delta t_i} \] 

5) If this sum restores the size of the cycle to the desired value then the conflict is resolved. If not, the manager increments the cost \(c\) and relaunches his appeal (returns to step 3)

3) Stage agent

A stage agent manages the sum of movements compatible with entries onto a junction. The stage agent is in charge of calculating the optimal duration so as to clear the vehicles waiting on the arcs concerned, and collaborating with other agents in order to develop a traffic lights plan.

This agent possesses a collection of both static and dynamic data that represents his internal situation:

**Static data**
- List of entry arcs: the sum of arcs authorized to clear if the stage is active (or green).
- the original junction for each arc entry.

In a strictly formal manner, the list of arc entries is represented by \(E=(a_1, c_1), (a_2, c_2), \ldots, (a_n, c_n)\) such that \(a_i\) is the arc entry \(i\), \(c_i\) is the original junction of the arc \(a_i\). This data structure enables the stage agent to retrieve traffic data at the entrance to the intersection at this stage, and to communicate with neighbouring junction agents who can send traffic to the actual junction.

The dynamic data concerns:
- The state of the stage: active or inactive. ‘Active’ signifies that the traffic lights controlling the arcs concerned by this stage are green. The vehicles are therefore authorized to depart.
- Duration of green light attributed to the stage.
- Date of the start of stage execution.

**Behavior of the stage agent**

The stage agent participates in the calculation process of the traffic lights plan, and is in charge of fixing the optimal duration of green light for the stage in question.

1) When the stage agent is questioned by the junction agent on the desired duration of green light, this duration \(d_i\) and an index that measures the urgency of the stage \(I_i\), are calculated and this information is transmitted to the junction agent.
2) If the stage agent receives confirmation from the junction agent, the stage agent halts the process.
3) If the stage agent receives an appeal with a cost \(c\), an offer is calculated and sent to the junction agent.

**Calculation of the desired duration of green light**

The optimal duration of green light is calculated by the following formula:

\[ T = \max_{i=1 \ldots m} \{T_i\} \]

Where \(m\) is the number of arcs entering at this stage, and \(T_i\) the time necessary to clear arc \(i\), which is given as: \(T_i = \frac{N_i}{d_i} + \frac{N_i \cdot L_i}{C_i \cdot V_i}\)

**Urgency index of a stage**

The urgency index of a stage \(j\), in order to award priority to a bus, is defined by the fact that the higher the index, the greater the urgency of the stage:

\[ I_j = \sum_{i=0}^{N} w_i \cdot e^{b_i} \]

Where:
- \(w_i = N_i/C_i \leq 1\), measures the degree of congestion of arc \(i\),
- \(b_i\): the number of buses present on arc \(i\).
- \(m\): the number of arcs entering via stage \(j\).
- \(e\) : the Euler constant in our example.

4) Route agent

The role of the route agent is to supervise all the bus agents so as to prevent a regulation at the local level of the bus and the creation of bus queues. In other words, this agent can modify the behavior of bus agents in two different ways:

*Directly*: by keeping those buses, which are ahead in the plan compared to the preceding buses, at the bus stop for a certain period of time.

*Indirectly*: by modifying the priorities of the bus.

![Junction Agent](image-url) ![Stage Agent (m)](image-url)

Figure 5: Collaboration in conflict resolution
This agent has a global overview of the route it operates, and can therefore detect bus queues and react to prevent queues forming. In the following example (Figure 6), the route agent AL123 has to detect the bus queue forming by bus agents B4, B5 and B6 and to diffuse it by slowing B6 and B5, which are ahead in the plan.

![Figure 6: Supervision of bus agents bus by the route agent.](image)

Internal state of the route agent
The route agent has the structure of the following data:
- The list of arcs travelled by the bus on this route.
- The list of stops on the route: for each stop, its position, and the distance separating it from the next stop.
- The list of buses on the route.
- The frequency of buses introduced onto the route.

For two consecutive stops \( A_i \) and \( A_{i+1} \), the route agent maintains the journey time \( d_{i,j} \) of the last bus. This helps to follow the bus’ journey and to calculate whether the bus is ahead or late compared to the bus immediately preceding it.

Behavior of the route agent
When a bus agent moves into a ‘stop’ situation, the time taken \( t \) to cover the distance \( L_{i+1,j} \) which separates the two stops \( A_i \) and \( A_{i+1} \), is transmitted to the route agent. The route agent then compares \( t \) to the time \( d_{i,j} \) taken by the preceding bus and will consequently decide if the bus is ahead or late. The route agent calculates the new priority of the bus agent as well as the length of time the bus should wait at the commercial stop if it is ahead (see [13] for more precisions).

III. IMPLEMENTATION OF THE PROTOTYPE

For the development of such distributed Multi-Agent Systems, the JADE platform (Java Agent Development Framework) has been adopted.

1) JADE Environment
JADE offers Java middleware based on a peer-to-peer architecture with the overall aim to provide a runtime support for agents [13]. To guarantee inter-operability between agents, JADE is compliant with FIPA (Foundation for Intelligent Physical Agents) specifications.

The use of JADE simplifies the development of the multi-agent system. It offers a naming and yellow-page service, message transport and parsing service, and a library of FIPA interaction protocols ready for use. The agent platform provides a Graphical User Interface (GUI) for the remote management, monitoring and control of the status of agents.

2) The urban network use for simulation tests
We tested the strategy on a small network of four intersections. The configuration of the network is as follows:
- The junctions have between two and four stages.
- The distance between two adjacent junctions varies between 200 and 400 metres.
- Each section comprises one or two lanes (see Figure 7).
- The maximum exit output of the arcs is identical and equal to 0.5 vehicles/second.
- At each entry onto the network we installed a source that generates vehicles at a frequency \( F \in [4 \text{ s} .. 10 \text{ s}] \).
- Two buses enter the network (see Figure 7). For the first bus (Bus 1), the frequency of the vehicles is of 80 seconds, it is of 180 seconds for the Bus 2.

![Figure 7: Architecture of the network used](image)

3) Results
We compare the developed Multi-Agent control strategy to a fixed time strategy with 30 seconds for each stage. We give here the results of the two strategies when applied for very heavy traffic conditions.
and buses. We can see a gain of 38% on the time spent by buses on the traffic lights and of about 51% of the strategy improves both the traffic of private vehicles and buses. We can see a gain of 38% on the time spent by buses on the traffic light and of about 51% of the vehicles lost time.

As shown on figures 8 and 9, the Multi-agent control strategy improves both the traffic of private vehicles and buses. Thanks to new information and communication technologies, the entities comprising the urban traffic system can communicate among themselves and negotiate in order to resolve traffic-related problems. In the second part, we presented a first multi-agent system for regulating large networks with mixed traffic. A graphical interface is build in order to follow the buses propagation. Further work will include more indicators.

**REFERENCES**


**Figure 9: Vehicles cumulated delays**

**IV. CONCLUSION**

The use of classic methods of control systems in the regulation of bimodal urban traffic presents certain disadvantages. Moreover, the problem concerns the regulation of intervals between buses. A macroscopic modelization of the system does not allow for the regulation of intervals between buses. Furthermore, calculations in microscopic modelization are time-consuming, which is impractical in the development of a control law. In this paper, we have shown that regulation based on a multi-agent system can solve the problem and improve the network circulation while respecting the intervals between buses. Thanks to new information and communication technologies, the entities comprising the urban network can communicate among themselves and negotiate in order to resolve traffic-related problems. In the second part, we presented a first multi-agent model that can calculate a plan for traffic lights based on the actual traffic situation and to favour the buses deteriorating the intervals between the vehicles on the same route.

Using on one hand, real time traffic flow measurements from road links, which can be provided by any traffic sensor such electromagnetic loops, video cameras, etc.; and on the other hand, the buses tracking provided by the AVM (Automatic Vehicle Monitoring), the strategy developed improves conditions of global circulation and reduces bus delays. A graphical interface is build in order to follow the buses propagation. Further work will include more indicators.