

Intelligent Traffic Light Management using Multi-Behavioral Agents

Luis Cruz-Piris, Diego Rivera, Ivan Marsa-Maestre, Enrique de la Hoz y Susel Fernandez
Departamento de Automática,
Universidad de Alcalá
Edificio Politécnico, 28805 Alcalá de Henares (Madrid), Spain
{luis.cruz, diego.rivera, ivan.marsa, enrique.delahoz, susel.fernandez}@uah.es

Abstract—One of the biggest challenges in modern societies is to solve vehicular traffic problems. In this scenario, our proposal is to use a Multi-Agent Systems (MAS) composed of three types of agent: traffic light management agents, traffic jam detection agents, and agents that control the traffic lights at an intersection. This third type of agent is able to change its behaviour between what we have called a *selfish mode* (the agent will try to influence the other neighbour agents of its type to achieve its goal) or an *altruistic mode* (the agent will take into consideration the other neighbour *selfish* agents indications). To validate our solution, we have developed a MAS emulator which communicates with the *Simulation of Urban MObility* (SUMO) traffic simulator using the Traci tool to realize the experiments in a realistic environment. The obtained results show that our proposal is able to improve other existing solutions such as conventional traffic light management systems (static or dynamic) in terms of reduction of vehicle trip duration.

Keywords—Multi-Agents System, Intelligent Transportation System, Smart Cities, Sensor Networks and Traffic Simulations

I. INTRODUCTION

Intelligent Transportation Systems (ITS) have experimented a fast improvement over the last years thanks to the evolution in the technologies that they rely on. Specifically, subsystems such as Advanced Traffic Management Systems (ATMS) have benefited from the rise of technologies like those used in devices derived for the Internet of Things (IoT) paradigm. These technologies have allowed to increase the data volume used to make decisions.

A side effect of the evolution of the ATMS in the last years has become a great challenge in modern systems: how to process such a high information volume so that the decision-making process is correct and efficient.

Our proposal consists of defining a MAS for intersection management and coordination between intersections about traffic lights management. This kind of system has shown its utility to solve problems within distributed environments.

In section II we review existent systems that modify traffic light phases in order to solve traffic congestion problems, and we discuss why MAS have been proven as an effective tool in this topic. In our case, the main goal of the system will be to control the traffic lights scheduling in a road network, so they can adapt it to the environment. Three types of agent can be distinguished in the system. The first type will manage the variation of the duration of phases in a traffic light, guaranteeing its correct operation. The second type will be in some of the elements of the traffic scenario (i.e. vehicles or sensors in strategic points of the vial network), and will use the information of their environment to make decisions about whether there is a possible congestion situation. Finally, the third type of agent will be located at intersections and will make decisions about the variation of traffic lights phases using the information generated by the other agents. This agent will be responsible for notifying the changes on the duration of phases of all the traffic lights in a single intersection. This third type of agent will vary its own operation per the congestion degree of the intersection where it is located. For low traffic loads, it will work in what we call *altruistic* or *collaborative* mode, in which the agent will help other agents of the same type in different intersections to achieve their goal. However, when high traffic loads are detected, the agent will work in a *selfish* or *isolated* mode, in which the agent will only react to reports from the agents located in the same intersection.

We have modelled and developed the agents described before to validate the solution and emulate the MAS in a simulation environment. We have connected this development with the widely recognized traffic simulator SUMO [1]. Furthermore, we have selected a portion of the well-known traffic scenario “TAPAS Cologne” (Travel And Activity PAtterns Simulation Cologne) [2], [3] for the validation experiments. The portion we have selected reproduces the traffic in a portion of the map of the city of Cologne (Germany), as shown in Figure 1, for a whole

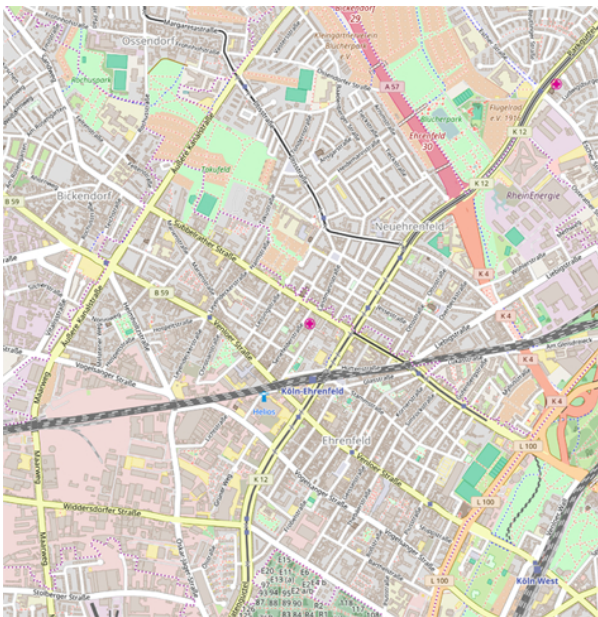


Fig. 1: OpenStreetMap capture of the portion of the scenario used in the system validation.

day.

One of the biggest challenges in the traffic management systems is being able to process all the information that is produced in their environment and make effective decisions using it. The main advantage of using a MAS to solve this kind of scenarios is the possibility to divide the problem in smaller sub-problems, and therefore improve scalability and efficiency of the system. This paper contributes to this goal in the following ways:

- By defining the agents and mechanisms needed to develop the proposed system, applying different operation modes to them depending on the circumstances of the environment (section III).
- By developing a MAS emulator, capable of interacting with the traffic simulator (section IV).
- By validating the proposed solution and evaluating it (section V). We have used a scenario based on real-life data, giving more validity to the obtained results. In addition, the data-set used is open source, allowing reproducibility of the results.

Finally, in sections B and VI we discuss this results and we expose the conclusions and future work lines derived from our work.

II. RELATED WORK

There are many factors involved in the correct circulation of vehicles. In fact, problems related to vehicle traffic are widely studied, as it is one of the most relevant challenges in modern societies.

The intelligent traffic management systems aim to have a global overview of the problem so they can make the right decisions in each case. We will focus here specifically in the solutions that are based on the optimization (static or dynamic) of the traffic light behaviour, as it is the goal that our proposed system pursues too.

One way of addressing the problem is to improve the traffic light scheduling, allowing a correct traffic light synchronization while trying to optimize the time that each vehicle should wait at the traffic light. It is possible to find many works where evolutionary algorithms (EA) are used to solve this problem. [4] investigate the potential of EA for the optimization of traffic light controllers using a Genetic Algorithm (GA), one of the most popular algorithms of this category. They conclude that to solve this type of problems it is useful to use evolution strategies. In [5] authors show the use of an iterative optimization algorithm (specifically a Particle Swarm Optimization (PSO) algorithm) to find successful cycle programs of traffic lights. They validate their proposal using the SUMO microscopic traffic simulator, obtaining an improvement in terms of total trip times and number of vehicles that arrive at their destination in a predefined simulation time.

[6] present a review on agent-based technology for traffic. The authors approach several topics and they have grouped them in two main categories: modelling and simulation, and control and management.

If we briefly analyse some of the challenges of traffic related problems, it is safe to say that it is geographically distributed, the environment is dynamic, and there is a strong interaction between the elements that compose them. [7] remark that the challenges that raise these properties can be well addressed with agent-based technology.

Therefore, we are going to focus in applications of agent technology used to improve traffic related problems. The work of [8] shows a review about this kind of solutions, where we are going to pay special attention to those which interact with traffic lights.

There are other Intelligent agent-based urban signal control systems like the ones presented by [9] or [10]. The first paper defines in a generic way, the necessary elements that should compose this kind of systems. The second one proposes a hierarchical multiagent architecture defined with some depth, including a description of the internal operation of the system. The most remarkable proposal in both works is the demonstration of the viability of the usage of MAS applied to intelligent traffic light management. Another approach to Intelligent agent-based transportation systems is shown in the work of [11], where an ontology-driven architecture is defined aiming to improve the driving environment through a traffic sensor network. This paper presents satisfactory results, but has only been validated in a small simple scenario.

ACTAM (Adaptive and Cooperative Traffic light Agent Model) [12] aims to reduce traffic congestion in urban roads. It proposes a complete agent system, able to synchronize and improve the efficiency in traffic light management in cities. It shows the improvement of using multiagent systems instead of a static traffic light scheduling. Unfortunately, it has only been validated in a small traffic network with about 30 intersections, which is far from a real-life scenario. Furthermore, the results are only compared with fixed-time signal control, and are not compared with other available management systems like

actuated control traffic.

III. PROPOSED SYSTEM

The main goal of our proposal is to reduce the duration of vehicle trips. To achieve this goal, we will modify the intervals of the traffic lights (phases) in a traffic network using a MAS. In the following sections, we are going to describe the types of agents involved in the system and the general operation of the system itself.

A. Agents

The development of our MAS is based on the definition of three main types of agents where each agent will be defined to perform a set of specific tasks.

1) “*TAgent*” *Traffic Light Management Agent*: These agents are located at each traffic light. Their goal is to manage the light changes. At system initialization, there are predefined static phases to change the states of the traffic light that the agent manages. This type of agent is also aware of the existence of the rest of agents of the same type located in the same intersection.

If the agent does not receive any message indicating otherwise, the lights will be set following the static predefined preferences. In case that another agent sends a request asking to change the duration of its phases, the agent will store and use the new phase duration if the new duration is between two predefined thresholds, and the rest of agents of the same type have confirmed the phase variation.

2) “*TJamAgent*” *Traffic Jam Detection Agent*: These agents are located both in vehicles and in sensors installed across the map. They receive information from the environment and decide, per their preferences, if they need to communicate their decision about the state of the environment (if the agent considers that there is a traffic jam or not). Agents located in vehicles know their geographic position, and their instantaneous velocity. If the obtained values for the velocity are below the threshold value defined in preferences during an established period, the agent will communicate its situation.

When this type of agent is in a sensor (induction loops, video-vehicle detections, etc.), the operation is similar. The only variation consists on the way it obtains the information to determine possible traffic jams, that will depend on the sensor.

3) “*IntersectionAgent*”: These agents are in the communications system of the traffic lights of a certain intersection. They receive the information from the nearest *TJamAgents* and decide whether they should change the phases of the traffic lights to prioritize the traffic flow in one of the ways of the intersection.

These agents can manage from 2 to N traffic lights, depending on the number of lanes that end in the intersection. This allows to reduce the amount of data used in the decision-making process by grouping the traffic lights in those which have the same state (red light or green light), as a variation in one group will affect the other.

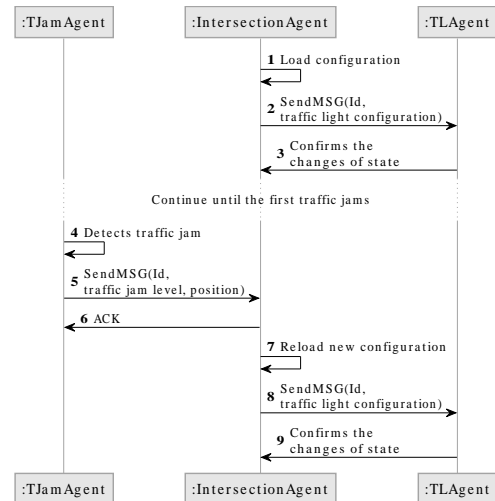
When the system is being initialized, the *IntersectionAgent* will request the *TLAGents* that are under its

management to send reports about the current phases of their traffic lights. Using this information, it will create a state machine that will be used later to perform the needed changes.

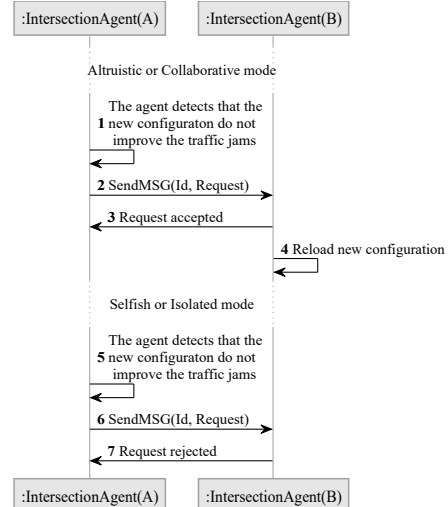
These agents have two operating modes:

- **Altruistic or Collaborative:** Additionally to the information received by the agent from *TJamAgents*, the agent accepts requests from other *IntersectionAgents* for the prioritization of some traffic flows.
- **Selfish or Isolated:** In this mode, the agent determines that the map zone under its management is congested enough not to cooperate with other agents and make its own decisions.

By default, every *IntersectionAgent* starts operating in *altruistic* mode. To avoid a possible system block, every time the agent starts operating in *selfish* mode, a timer is started, so the time an agent can operate in that mode is always limited. This behavior, together with the main actions and exchange of messages that the agents perform, is shown in Figure 2.



(a) Main actions and messages



(b) Altruistic vs. Selfish mode

Fig. 2: Communication between agents.

B. Agent behaviour

The operation of the system is based in the existence of a set of agents able to obtain information from the environment, and to decide when there is a congestion situation, agents able to change the traffic light states, and agents able to manage the external and internal synchronization of the phases of each traffic light in an intersection.

In this section, we are going to show the operation of the proposed MAS by using a basic use case formed by the two intersections shown in Figure 3. The different elements that compose the system are represented in the following way: The traffic lights (TLAgents) are represented by red lines or green lines, depending on their state, the yellow triangles represent the vehicles moving in each moment (TJamAgents), and the IntersectionAgents appear as red areas.

When we talk about horizontal traffic flow we are referring to the flow generated by vehicles going from the right to the left of the figures. On the other hand, vertical traffic flows are those generated from the top to the bottom of the figures and vice versa.

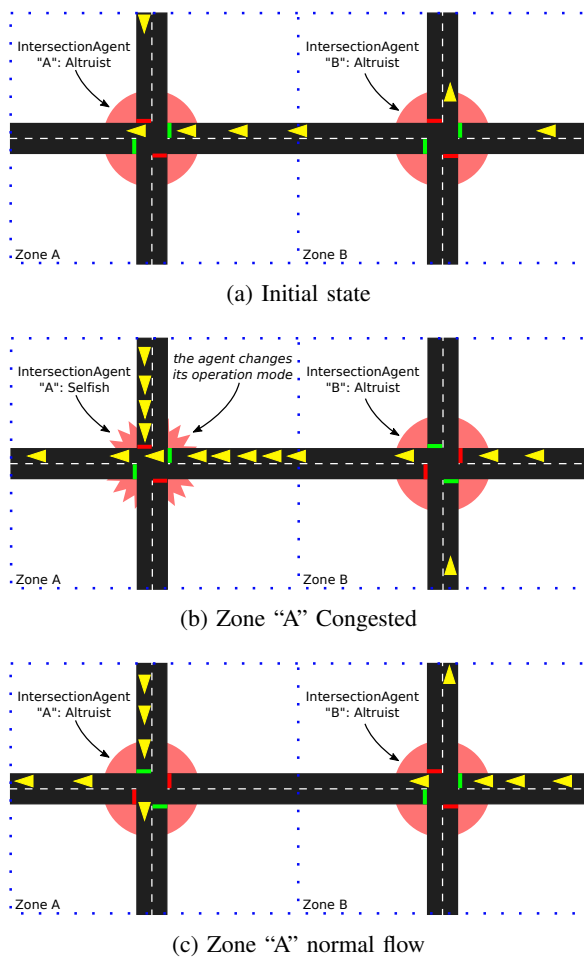


Fig. 3: MAS operation use case.

- 1) **Initial state** (Figure 3a): The IntersectionAgent "A" and the IntersectionAgent "B" start in the *altruistic*

operation mode, so they will collaborate to improve the traffic flow where the traffic volume is higher.

- TJamAgents that are in Zone A have a velocity below the maximum velocity value due to the high traffic volume. Each agent decides individually that it must report this situation to the IntersectionAgent "A".
- IntersectionAgent "A", using the received information, decides to prioritize the horizontal traffic flow and makes two requests:
 - Reports to the TLAgents of its intersection that they must increase the duration of the traffic lights green state of their phases.
 - Reports to the IntersectionAgent "B" this change so it can adjust the synchronization of the phases of the traffic light it manages.

- 2) **Zone "A" Congested:** IntersectionAgent "A" changes its operation mode to the *selfish* mode. In Figure 3b it is shown traffic jams both in the vertical traffic flow of the Zone "A" and the horizontal traffic flow. IntersectionAgent "A" stops collaborating with the rest of IntersectionAgents, performing the following actions:
 - Reports to the IntersectionAgent "B" that it must limit the horizontal traffic flow. Given that the IntersectionAgent "B" is still operating in *altruistic* operation mode, it will prioritize this request over the reports received from the TJamAgents. Therefore, it will request his TLAgents that they must reduce the duration of the traffic lights green state of their phases in the horizontal traffic flow.
 - Reports to the TLAgents that they should change the traffic lights phases so red and green states are of the same duration (for vertical and horizontal flows).

- 3) **Zone "A" normal flow;** After the actions of the previous step, The Zone "A" returns to normal traffic flow values, and therefore, the IntersectionAgent "A" changes its operation mode to the *altruistic* mode. Figure 3c shows how the horizontal traffic flow has been reduced. Each IntersectionAgent manages again the duration of its traffic lights using the information received from the TJamAgents and other IntersectionAgents.

IV. IMPLEMENTATION

Once the proposal has been described, it is important to validate it using an implementation where it can be confronted with a realistic situation. In our case, we have developed a simulation platform composed basically by two modules: Traffic simulation module (SUMO traffic simulator + TAPASCologne simulation scenario) and MAS emulation module implemented using Python.

The communication between both modules is performed using the TraCI (*Traffic Control Interface*) tool [13], included in the SUMO package. This tool provides a TCP-based client/server architecture that allows to control

and modify the SUMO simulations through an external application.

A. Simulation scenario

Using a realistic simulation scenario is essential to perform the validation of the system, as it will provide conditions like the real-life scenarios.

In our proposal we have chosen the scenario called “TAPAS Cologne” that is referenced in the SUMO documentation. It is a complete simulation scenario of the German city of Cologne. It was created by the Institute of Transportation Systems at the German Aerospace Center (TIS-DLR), and its goal is to reproduce, with the maximum possible realism, the urban traffic of Cologne. It defines a map of 400 km^2 and 24 hours of traffic.

The original simulation scenario is composed by a road network with 68642 edges, 30354 nodes and 1547333 routes. The size of this scenario causes very high simulation times. Therefore, for the validation of our proposal, we have decided to crop the scenario in a smaller portion that, still being representative of the original scenario, will yield lower simulation times. The solution proposed by [14] is tested on a simulated network of the Lower Downtown Toronto network. Analysing the features of that network, we have cut the scenario of “TAPAS Cologne” to obtain a new sub-scenario.

This sub-scenario has 1416 edges and 716 nodes (73 of those are intersections managed by traffic lights and the rest are managed by priority rules). Equally, the routes of the scenario have been reduced to a more manageable number, using just the routes that are related to the chosen portion of the scenario. The total selected routes are 246374.

Moreover, in the “TAPAS Cologne” documentation it is said that for a proper simulation, the parameter *scale* must be set at 0.3. This means that only the 30% of the routes will be actually inserted during the simulation. For our sub-scenario, this scale value must be calibrated again. We have done this calibration by executing consecutive simulations increasing the value of *scale* in 0.01 for each new simulation. For each simulation, the number of teleports have been measured (a teleport is an event that happens in SUMO simulations when vehicles are blocked for a given time. It consists on the automatic disappearing and appearing of the vehicle, in order to unlock the simulation). The result of these simulations is shown in Figure 4.

Using the results of these tests, we have decided to increase the value of the parameter *scale* to 0.40. Because at this value is where we detect that the number of teleports starts to raise (although it is still a reasonable low value: 241), and allows us to use a high volume of traffic. Using this value for *scale* means that the actual number of routes inserted during the simulation is 98550.

B. System operation

As we have pointed before, the MAS proof of concept module has been completely written in Python. In the

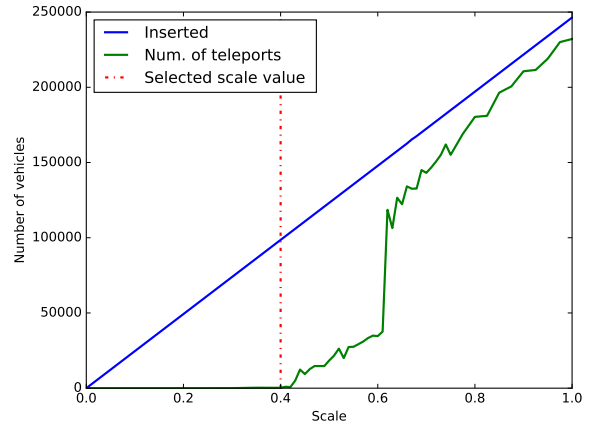


Fig. 4: Evolution of the teleports number according to the scale parameter.



Fig. 5: Representation of network edges (black lines) and intersection agents (red dots).

Figure 6, we show a block diagram of the whole simulation system.

The main steps followed while the system is being executed are:

- 1) System initialization: The application reads the data from the sub-scenario, generates the map division, initializes the data structures that will be used later, and launches a subprocess that starts the SUMO simulator.
- 2) Loop until end of the simulation: There is a parameter in the application configuration that states the duration in seconds of each simulation step. Specifically, we have chosen a value of 30 seconds for this parameter. The MAS module will perform the following tasks including requests to the simulation module (via TraCi):

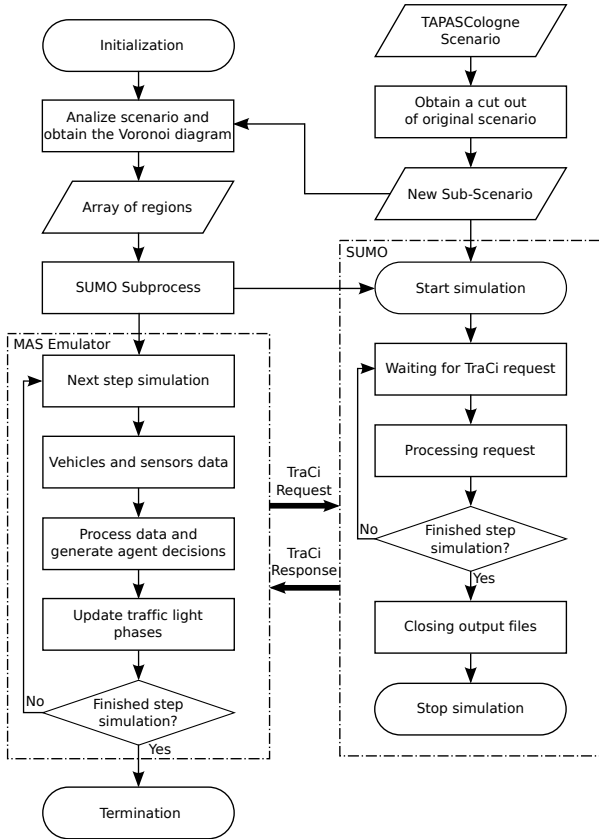


Fig. 6: Block diagram of the Multiagent System and simulation platform.

- a) Requests one step simulation (30 seconds) and then waits for the end of the step.
 - b) Requests the current parameters of each vehicle that was active in that step of simulation, and also the information from the net induction loops.
 - c) Processes the information, and models it for the agents to make decisions.
 - d) Requests the variation of traffic light phases.
- 3) End of the simulation.
 - 4) Analysis of the data contained in the SUMO output file, containing the simulation results. This results are compared with those obtained from a previous simulation performed in the same scenario but without the intervention of our MAS.

V. EVALUATION

A. Experiment settings

The evaluation of the proposed system has been carried out by defining different sets of simulations over the same sub-scenario. In these experiments, the input data have been the same, and the results obtained have been evaluated over the same set of routes. These are the carried-out experiments:

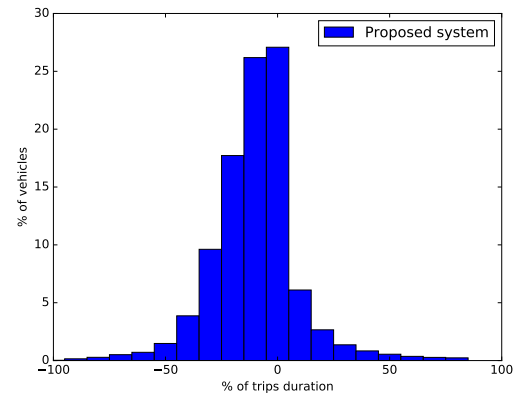
- Experiment 1 (DEFAULT): Simulation executed using the default configuration offered by the TAPAS-Cologne scenario. The traffic light phases are stati-

cally defined and do not change during the simulation.

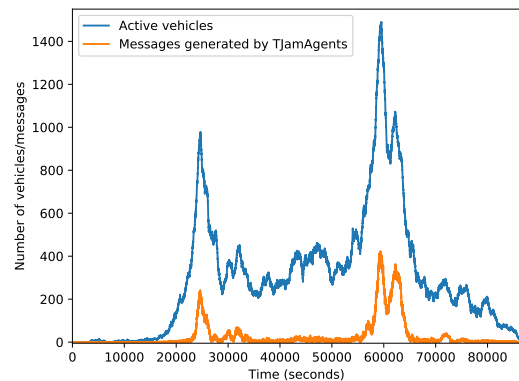
- Experiment 2 (ACTUATED): Simulation executed using the SUMO Actuated Traffic Lights system.
- Experiment 3 (MAS): Simulation executed using the MAS emulation module.

After finishing all the experiments, we have focused on the evaluation of the duration of each simulated trip. Therefore, a decreasing duration of trip will show the improvement in traffic efficiency introduced by the variations in traffic light phases.

In Figure 7a, we show the results of comparing the experiment “DEFAULT” with the experiment “MAS”. The x axis represents the percentage of increase or decrease in the trip durations (using steps of 10%) and the y axis represents the percentage of vehicles inserted over the total vehicles in the simulation. We also measured the number of active vehicles at each time step and the messages generated by the TJamAgents. These results are shown in Figure 7b.



(a) % of increase or decrease in the trip durations over the % of vehicles.



(b) Active vehicles during the simulation vs. messages generated by the TJamAgents

Fig. 7: Experiment results.

B. Results discussion

To allow for an easier analysis of the obtained results, the values have been grouped in three sets: one set for

the duration of trips that are lower than the ones in the experiment 1, other for the duration of trips that are equal, and a third one for the trips that are higher in duration. Table I shows the percentage of vehicles obtained in each category for experiments 2 and 3.

Tabla I: Results summary (% of total vehicles)

Trip durations	Lower	Equal	Higher
Actuated Traffic Lights	58.70	28.41	12.89
MAS	60.52	27.08	12.41

Given that the system has been defined to prioritize some traffic flows over others, it is expected that not every vehicle in the simulation is able to reduce the duration of its trips. The obtained results show that the percentage of vehicles suffering an increase in the duration of its trip is low. The value is lower than the 13% in the experiment 2 and 3. It is also worth mentioning that half of those vehicles only experienced a 10% of increase in trip duration.

The improvement between using the “actuated traffic lights” system and our MAS may seem not too remarkable, but it must be contextualized. The simulation using the first method needs an induction loop in every edge of the network that ends in an intersection with traffic lights. Besides that, it must evaluate in every simulation step the registered values by each sensor so it can modify the duration of the traffic lights accordingly. Conversely, the MAS can perform the same task without using fixed sensors distributed along the network, as it is able to obtain information from the vehicles themselves (and incidentally from the possible induction loops installed in some roads), it also limits the amount of data needed to make decisions, as each agent decides if it is necessary to communicate its state or not.

On the other hand, it is important to highlight that, in very high traffic congestion situations, such as the ones simulated in our experiments, there are certain intersections that can reach blocking states where the variations in the actuated traffic lights are not enough to solve them. In that kind of situations, the proposed MAS has been able to “unlock” 1790 vehicles, that have been able to reduce the duration of their trip in Experiment 3.

Finally, the values shown in Figure 7b validate that the MAS is capable of reacting to situations of serious traffic jams, regardless of the number of active vehicles.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have addressed traffic congestion situations which is one of the most relevant challenges in the most of big cities around the world. More specifically, the main goal of our proposal was to reduce the traffic congestion situations by changing duration of traffic light phases. After analysing the results, and comparing them with other solutions like the static definition of traffic light scheduling or the actuated traffic lights, it is possible to say the use of a MAS is effective.

A secondary goal, but mandatory for the validation of the proposal, has been the implementation of the process

that emulates the MAS and managing to communicate it with a widely-used traffic simulator such as SUMO. This goal has been accomplished, and has allowed us to perform experiments over a simulation scenario, which provides higher guarantees about the usefulness of the solution.

Although the conducted experiments yield satisfactory results, there are some avenues for further research in this topic. The decision-making rules of the agents used in this first proof of concept implementation have been very simple but effective, just defining some triggering values for the agents from which the TJamAgents report the situation, and a weighted value applied to the data received by each InteserctionAgent. Those values are used to decide if the traffic is prioritized in one way or another. Once the viability of the system has been proven, and the connection with the simulation platform has been developed, the future work will be related to make more complex decision-making algorithms, that allow better and more effective results. Finally, it would be important to study which is the minimum percentage of agents that should participate in the MAS without deteriorating the system effectiveness.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministry of Economy and Competitiveness grants TIN2016-80622-P, TIN2014-61627-EXP and TEC2013-45183-R, and by the University of Alcalá through CCG2016/EXP-048.

REFERENCES

- [1] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO - Simulation of Urban MObility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.
- [2] S. Uppoor and M. Fiore, “Large-scale urban vehicular mobility for networking research,” in *Vehicular Networking Conference (VNC), 2011 IEEE*. IEEE, 2011, pp. 62–69.
- [3] “TAPASCologne scenario,” <http://sumo.dlr.de/wiki/Data/Scenarios/TAPASCologne>, online; accessed 14 July 2017.
- [4] H. Taale, T. Bäck, M. Preuss, A. Eiben, J. De Graaf, and C. Schippers, “Optimizing traffic light controllers by means of evolutionary algorithms,” in *Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing*, vol. 3, 1998, pp. 1730–1734.
- [5] J. Garcia-Nieto, E. Alba, and A. C. Olivera, “Swarm intelligence for traffic light scheduling: Application to real urban areas,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 2, pp. 274 – 283, 2012, special Section: Local Search Algorithms for Real-World Scheduling and Planning. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0952197611000777>
- [6] A. L. C. Bazzan and F. Klugl, “A review on agent-based technology for traffic and transportation,” *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 375–403, 006 2014.
- [7] J. L. Adler and V. J. Blue, “A cooperative multi-agent transportation management and route guidance system,” *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5, pp. 433–454, 2002.
- [8] B. Chen and H. H. Cheng, “A review of the applications of agent technology in traffic and transportation systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 485–497, 2010.
- [9] D. A. Roozmond, “Using intelligent agents for pro-active, real-time urban intersection control,” *European Journal of Operational Research*, vol. 131, no. 2, pp. 293–301, 2001.
- [10] M. C. Choy, D. Srinivasan, and R. L. Cheu, “Cooperative, hybrid agent architecture for real-time traffic signal control,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 33, no. 5, pp. 597–607, Sept 2003.

- [11] S. Fernandez, R. Hadfi, T. Ito, I. Marsa-Maestre, and J. R. Velasco, "Ontology-based architecture for intelligent transportation systems using a traffic sensor network," *Sensors*, vol. 16, no. 8, p. 1287, 2016. [Online]. Available: <http://www.mdpi.com/1424-8220/16/8/1287>
- [12] C. Ruey-Shun, C. Duen-Kai, and L. Szu-Yin, "Actam: Cooperative multi-agent system architecture for urban traffic signal control," *IEICE transactions on Information and Systems*, vol. 88, no. 1, pp. 119–126, 2005.
- [13] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "Traci: An interface for coupling road traffic and network simulators," in *Proceedings of the 11th Communications and Networking Simulation Symposium*, ser. CNS '08. New York, NY, USA: ACM, 2008, pp. 155–163. [Online]. Available: <http://doi.acm.org/10.1145/1400713.1400740>
- [14] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): Methodology and large-scale application on downtown toronto," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, Sept 2013.