

# A MULTI-AGENT SIMULATION PLATFORM APPLIED TO THE STUDY OF URBAN TRAFFIC LIGHTS

Raul Cajias, Antonio Gonzalez-Pardo and David Camacho

*Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad Autónoma de Madrid  
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain  
{raul.cajias, antonio.gonzalez, david.camacho}@uam.es*

**Keywords:** Distributed system, Software engineering, Swarm computing.

**Abstract:** The Multi-Agent system paradigm allows the development of complex software platforms to be used in a wide range of real-world scenarios. One of the most successful areas these technologies have been applied are in the simulation and optimization of complex systems. Traffic simulation/optimization problems are a specially suitable target for such a platform. This paper proposes a new Multi-Agent simulation platform, where agents are based on a Swarm model (lightweight agents with very low autonomy or proactivity). Using this framework, simulation designers are free to configure road networks of arbitrary complexity, by customizing road width, geometry and intersection with other roads. To simulate different traffic flow scenarios, vehicle trajectories can be defined by choosing start and end locations and providing traffic generation functions for each one trajectory defined. Finally, how many vehicles are generated at each time step can be determined by a time series function. The domain of traffic simulation has been selected to investigate the effect of traffic light configuration on the flow of vehicles in a road network. The experimental results from this platform show a strong correlation between traffic light behavior and the flow of traffic through the network that affects the congestion of the road.

## 1 INTRODUCTION

As urban areas grow, more vehicles are introduced into streets and highways, exhausting their capacity. This uncontrolled proliferation directly influences the duration and frequency of traffic jams, resulting in waste of time and money. The quickest way to counter-balance this rise in traffic is to augment the capacity of the most frequently used roads by ways of adding more lanes, or creating new roads. However this solution can be slow and expensive to implement. Short foresight in road expansion projects can result in temporary benefits, while failing to tackle the underlying problem. Urban planners often find themselves mending broken systems, lacking the tools to better understand the inherent cause of their jams in order to make optimal choices. Dynamic traffic controls, capable or adapting to sudden spurs of traffic to maximize traffic flow, can yield better results by keeping up with ever-changing traffic demands without the need to refactor the existing road infrastructure.

Traffic flow models help design dynamic control like the ones just described. Flow models are typically categorized by their level of detail in three broad classes: microscopic, mesoscopic and macroscopic.

Microscopic models describe both space and time behavior of the system's entities (vehicles and drivers) as well as their interactions at a high level of detail. Macroscopic models describe traffic behavior through high-level terms like flow-rate, density and velocity. Finally, mesoscopic models consider dynamical properties that are simple enough to be simulated for long time.

The use of Multi-Agent Systems, MAS, for traffic simulation is not new. There is a lot of literature on the subject, highlighting the advantages of using MAS, (Burmeister et al., 1997). MAS has also been used for microscopic traffic simulation in which different characteristics of the population are modeled. (Zhang et al., 2005) uses MAS for simulating single-lane roads. In this work, agents have a decision tree that allows them to adapt their velocity and acceleration depending on the environment and maximum speed. In (Ehlert and Rothkrantz, 2001), an analysis of agent behavior as a function of their profile is presented: agents are given two behavior profiles representing two types of drivers—fast and aggressive or slow and careful. Finally, agents where given a Believe-Desire-Intention (BDI) reasoning process. That study focused on both vehicles and driver to

study how agent behaviors can impact road traffic flow.

Traffic network systems such as *UTOPIA-SPOT*, (Mauro and Taranto, 1989), and *SCOOT*, (Robertson and Bretherton, 1991), integrate a number of different traffic network models to provide an urban traffic management solution through the coordinated operation of traffic signals to smooth the flow of traffic and increase circulation in cities. The use of multi-agent systems for the modeling of traffic has been studied among other works in (Radecky and Gajdos, 2008) and (Xiao-Fan Zhi, 2008). (M. van den Berg and Hellendoorn, 2003) on the other hand, uses the *METANET* model (Messmer and Papageorgiou, 1990), a macroscopic model for mixed urban and freeway traffic networks, that provides integrated controls for traffic flow by implementing a model predictive control.

Previous work showed how traffic light configuration affects the average speed of the vehicles and how the use of traffic lights with road-congestion based behavior reduces the time needed by the vehicles to arrived to the destination. This work perform a deeper study of the system proposed in (Cajias et al., 2011) with more complex maps. Agents have *swarm*-like behavior in that they follow simple rules, and lack a centralized control structure. The main contribution of this work is related to the implementation of a configurable, agent-based traffic simulation framework, based on the *METANET* model. We model traffic flow by creating of lightweight agents using a time series distribution, enabling us to treat traffic jams as a particle swarm optimization problem. We use this approach to find dynamic traffic control configurations that will yield smooth flow of traffic at any given time. Finally, the paper shows some experimental results on how the behavior of traffic lights can impact flow in the network, as a function of road-load.

This paper is structured as follows: next section will give a description of the multi-agent model, and the simulation framework that has been built to implement it. Section 3 gives an overview of the simulation design process, while section 4 provides a detailed description about the set of experiments carried out in this study. Finally chapter 5 draws the main conclusions reached in this work.

## 2 DESCRIPTION OF THE MULTI-AGENT MODEL

### 2.1 The Simulation Environment

Road networks are modeled as a weighted directed

graph  $N = G(V, E)$ , with source  $s \in V$  and sink  $t \in V$  of  $N$ . The capacity of an edge is a mapping  $c : E \rightarrow \mathbb{R}^+$ , denoted by  $c_{uv}$ . Furthermore, the flow of the network is a mapping  $f : E \rightarrow \mathbb{R}^+$  denoted by  $f_{uv}$ , subject to the following constraints:

1. *Capacity constraint*: For all  $u, v \in V$ , we require  $f_{uv} \leq c_{uv}$ .
2. *Conservation of flows*: For all  $u \in V - \{s, t\}$ , we require  $\sum_{v:(v,u) \in E} f_{vu} = 0$
3. *Skew Symmetry*: For all  $u, v \in V$ , we require  $f_{uv} = -f_{vu}$

Road capacity is a function of the road's width (number of lanes) and length. Of the two, only the number of lanes in a road follows specific rules that can serve to differentiate one road from another.

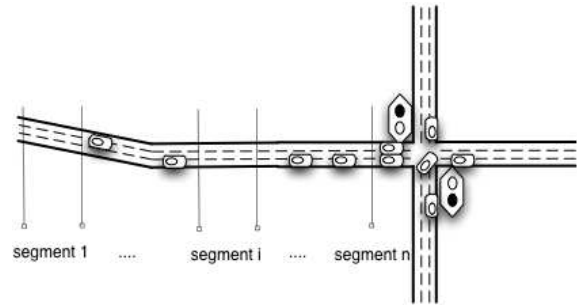


Figure 1: Roads are sectioned into equal-length segments. Each segment has a maximum number of vehicles it allows to pass through each time step.

Given equal-length road segments, road capacity translates to the number of vehicles that can fit that segment, multiplied by the number of segments in a road. Thus, by toggling the number of vehicles that fit in a single section, new road types can be defined. This implementation has the added benefit that traffic accidents can easily be modeled: By decreasing the maximum number of vehicles in a single segment for an arbitrary number of steps, the impact of an accident in roads can be easily simulated.

### 2.2 Vehicle and Traffic Light Agents

The system is composed of two types of entities: vehicles and traffic lights. These agents interact with each other by keeping track of local information available to them, and acting accordingly.

Upon creation, vehicles are assigned a shortest distance trajectory between two arbitrary points in the road network. Once set, this trajectory cannot be changed. They are scheduled to navigate the map in the order they were created, and attempt to move at a constant pace each time step.

These agents can be created following a distribution function or at a constant pace. As they travel through the road network, agents require local information to decide whether to move or stay put. This information is made available to them by consulting the state of each road. As they traverse through the network, vehicles can consult the status of the traffic light and the number of cars currently residing at the immediate next road section, if the next road section has already reached the maximum number of vehicles that can reside it, or a traffic light in red, vehicles do not move. After each time step, each agent updates the status of the local blackboard to reflect their actions.

Traffic light agents have two states  $\{red, green\}$ , which they keep for an arbitrary number of time steps. Their status serves to smooth incoming traffic in the system. These agents are located in the intersections of the different roads, so only two-road intersections are currently supported. Traffic light agents have a cyclical behavior in which the agent will execute all its behavioral states. A parameter called *duration*, defines the number of iterations that the cyclical behavior will take. How much time does the traffic light stay in green or in red, is defined by the *split* parameter.

In order to reduce the traffic jams, an intelligent traffic light agent has been implemented. The difference between this intelligent traffic light agents and the previous one is that the intelligent traffic light adjusts its *split* parameter depending on the traffic flow of the roads. In order to do that traffic light uses the following process:

1. The traffic light agent compares the the level of saturation of vehicles on each road. The equation:

$$f(road) = \frac{n_{Segments} * n_{VehiclesPerSegment}}{n_{VehiclesOnRoad}} \quad (1)$$

is used as a metric of saturation. The operation  $n_{Segments} * n_{VehiclesPerSegment}$  defines the maximum number of vehicles allowed in a road, while  $n_{VehiclesOnRoad}$  is the total number of vehicles on that road.

2. Finally, the split parameter is adjusted such that saturated roads have a longer lasting green light. The parameter is adjusted using the following function:

$$NewSplit_a = \frac{\max(f(road_a))}{\sum_{i=1}^n f(road_i)} \quad (2)$$

Where  $\max(f(road_1))$  represents the traffic flow of the road with more traffic, and  $\sum_{i=1}^n f(road_i)$  is

the sum of the traffic flow of all roads in the intersection.

With this procedure, roads with more traffic will take advantage of the traffic light behavior, because the traffic light will be more time in green state to allow more traffic in this road.

### 3 DESIGNING A SIMULATION

Simulations are designed in an XML description file. This file defines all aspects of the simulation, allowing quick prototyping of new simulations and ease of reproduction of past experiments.

Roads are defined as a series of points in a cartesian plane, an orientation and one of two directions  $\{1way, 2way\}$ . The orientation helps determine which state of the traffic light pertains to the current road, and the direction facilitates the creation of 2-way streets. Finally, the type of road used is determined by the xml element used to define the road. Currently, only the type 'street' is supported—5 cars per segment—but more may be defined by the user.

To create the road network, road specifications are parsed from the XML file. Using the points defined for each road, road intersections are found and a directed road network  $n = G(E, V)$  is generated. As roads may have any number of intersections, a single road is represented by a collection of network edges  $E$  in the network. To address edges making up a road, the following notation is used:  $a_{start} \rightarrow ax1, ax1 \rightarrow a_{finish}$ .

Trajectories are defined under two sub elements: origin and destination. In each, the percentage of traffic to start *from* and finish *in start* and *finish* road nodes is specified. This allows the freedom to play with different traffic flow configurations.

Finally, traffic lights may be placed at road intersections, by simply specifying which road intersection to find in the network graph. A simple static traffic light has been preloaded into the framework, but users may define their own traffic lights by extending the static traffic light's functionality in the form of a Java class, and passing the fully qualified Java path as part of the `tl` element attributes.

Finally, the duration attribute specifies how much time a whole  $\{red, green\}$  iteration is to last, while the traffic light split refers to how that duration time is split between each state. In the above example, a whole iteration lasts 10 time steps, spending 5 time steps in the *green* state and 5 in the *red* state. Once the simulation is completed, an XML output file is created for each trajectory vehicles took to reach their desired destinations.

## 4 EXPERIMENTAL RESULTS

The simulation framework was used to observe the impact different traffic lights can have over the flow of vehicles through a road network. Each experiment was carried out using one of two kinds of traffic light agents—static and adaptive—and the average speed of the vehicles traversing a particular trajectory was then used to measure the impact of these traffic lights at each simulation time step.

### 4.1 A Single Intersection

The first experiment carried out consisted in measuring the effects of static and adaptive traffic lights on the flow of traffic through a single intersection. Vehicles were introduced into the simulation at the rate of one every time step, allowing a maximum of 400 cars to traverse through the network at a time. They were set to start from one of two starting points of  $\{a, 1\}$  with probability  $p(v)$  and finishing at the end of 1. Finally a simulation was launched the same parameters with each of the traffic light agents.

The average speed of vehicles traversing the segment of road  $\{a_{start} \rightarrow aX1\}$  is plotted over time in figure 3. As expected, average speed when 10% of vehicles start from vertex  $a$  is higher for a static traffic light than an adaptive one. However, when 90% of traffic starts from the same vertex, adaptive traffic lights achieve better results. In fact, comparing the average vehicle speed of the road under 90% and 50% traffic distributions, the stressed adaptive traffic lights seem to perform at least as good as static traffic lights with an even traffic distribution.

### 4.2 Four Intersections

Again two separate but similar simulation batteries were defined—one set for static and one for adaptive lights—with the same vehicle distribution and traffic light parameters as with the previous experiment. The average speeds of roads in a single trajectory  $\{a_{start} \rightarrow b_{finish}\}$  are displayed in figure 3. For vehicles traversing the network of traffic lights, periodic spikes and valleys begin to populate each road's average velocity plot. Traffic lights act as a flow filter, smoothing out traffic velocity in the system at each intersection, until vehicles exit the network at maximum speed. Towards the end of the simulation, periodic patterns in velocity suggest that the system reaches some equilibrium.

Introducing adaptive traffic lights into the system yields a more interesting velocity graph—figure 4.

The attempt by early traffic lights to maximize vehicle flow, seems to cause a ripple effect to the network. This results in higher average speed obtained after each traffic light. In general, the average road velocity experiences an improvement over the static traffic light experiment.

These results coupled with the quick simulation turnaround time yielded by this framework, suggest that there is ample room for exploring the impact of more complex traffic light agents, and that such territory can be covered with the help of this platform.

### 4.3 Stressing the System

To gain some perspective on the challenges that real traffic simulation entails, we performed a final batch of tests on a 10x10 city grid. This grid size provides an order of magnitude closer to the realm of traffic problems we will attempt to optimize in the future. At this level, visualizing the dynamics of traffic flow can be quite difficult, as the number of possible roads increases dramatically. To simplify the comparison, we take into account the total number of time steps it takes vehicle to reach their destination, instead of the previous road-by-road analysis.

For consistency, this final batch of experiments was performed using the parameters outlined in 1, the simulation was run for 10000 steps and the creation of vehicles was capped at 100 to avoid over-saturation of the starting node. We note that the arrival time for a traffic distribution of 90% grows at a consistent, almost linear pace. For the 50 and 10% traffic distribution tests, the rate of increment in arrival time grows with each time step. For the 90 and 10% distributions, the adaptive traffic light is able to achieve an arrival time below that of its static counterpart, while matching—as expected—the growth of the 50% distribution.

## 5 CONCLUSIONS AND FUTURE WORK

Recognizing the breadth and depth of traffic flow problems, a Multi-Agent Traffic Simulation Framework was built. Using this framework implements a Swarm agent model, where agent behavior is driven by a set of simple, finite and unchanging set of rules. Simulation designers are free to configure road networks of arbitrary complexity, by customizing road width, geometry and intersection with other roads. To simulate different traffic flow scenarios, vehicle trajectories can be defined by choosing start and end locations and choosing what fraction of the vehicles

Table 1: Above, the parameters used in the static traffic light (TL) simulations. *Duration* refers to the total time it takes to complete a single iteration through the {red, green} state cycle. In these experiments, the percentage of time dedicated to allowing cars to pass from each direction was equal.

Tests	$p(a_{start})$	$p(l_{start})$	Total Duration	Traffic Light
Test 1	10%	90%	100 Steps	Static. 50% Green
Test 2	50%	50%	100 Steps	Static. 50% Green
Test 3	90%	10%	100 Steps	Static. 50% Green
Test 4	10%	90%	100 Steps	Adaptive
Test 5	50%	50%	100 Steps	Adaptive
Test 6	90%	10%	100 Steps	Adaptive

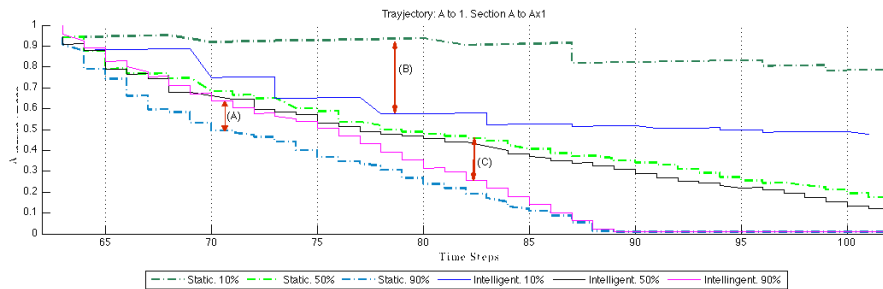


Figure 2: Differences in average speed for vehicles with static and dynamic traffic light agents. Trajectory  $\{a_{start} \rightarrow b_{finish}\}$ . (A) High traffic. (B) Low traffic. (C) Moment when average speed from dynamic traffic lights diverges from a static under medium traffic

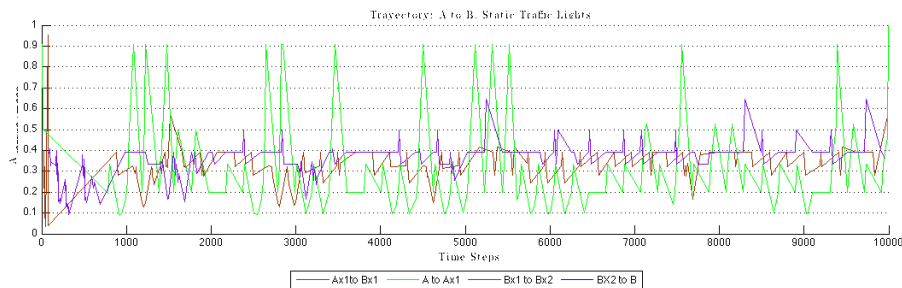


Figure 3: A plot of the average speeds attained by vehicles traversing four roads in the trajectory  $\{a_{start} \rightarrow b_{finish}\}$  under static traffic lights.

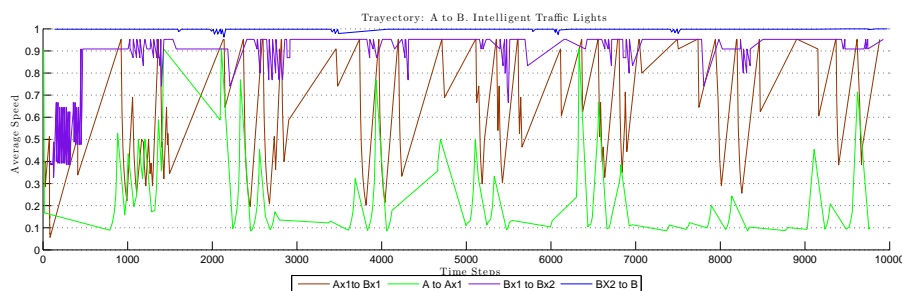


Figure 4: A plot of the average speeds attained by vehicles traversing four roads in the trajectory  $\{a_{start} \rightarrow b_{finish}\}$  under dynamic traffic lights.

generated at each step should take said trajectories. Finally, how many vehicles are generated at each time step can be determined by a time series function.

The flow of traffic thought the network is regu-

lated using two kinds of traffic light agents in order to compare their impact in the overall flow of the network. The first type of agent—a static traffic light—is configured to have a set cycle *duration*, an a set *split*

time fraction between each of its states. The second type of traffic light agents—and adaptive traffic light agent—attempts to optimize the average speed of the vehicles in the network, by adapting its *split* value to match the incoming volume of traffic it has control over. An improvement in average speed was observed for experiments using the adaptive traffic light agent, suggesting that such agents may be key elements in a wider flow optimization strategy, under more demanding traffic scenarios.

Further work is planned to test the performance of these two traffic light agents in network graphs, orders of magnitude larger than the ones used in these experiments, and the implementation of a time series vehicle generation function that better mimics real traffic flow scenarios. We are also interested in allowing cars to adapt their trajectory to optimize their movement through the network, based on local information available to them. Finally there is strong motivation to use automatic discovery methods such as genetic algorithms, to find combinations of different types of traffic light agents in a network graph, that could reach sub-optimal network flows.

## ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation under grant TIN2010-19872 and by Jobssy.com.

## REFERENCES

- Burmeister, B., Haddadi, A., and Matylis, G. (1997). Application of multi-agent systems in traffic and transportation. *IEE Proceedings - Software*, 144(1):51 – 60.
- Cajias, R., Gonzalez-Pardo, A., and Camacho, D. (2011). Multi-agent traffic simulation framework for evaluating the impact of traffic lights. In Science, S. and Publications, T., editors, *Agents and Artificial Intelligence*, Communications in Computer and Information Science, pages 443 – 446. Springer Berlin / Heidelberg.
- Ehlert, P. A. and Rothkrantz, L. J. (2001). Microscopic traffic simulation with reactive driving agents. In *IEEE Intelligent Transportation Systems Conference Proceedings*, pages 861 – 866.
- M. van den Berg, A. Hegyi, B. D. S. and Hellendoorn, J. (2003). A macroscopic traffic flow model for integrated control of freeway and urban traffic networks. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2774–2779, Maui, Hawaii.
- Mauro, V. and Taranto, C. D. (1989). Utopia. In *Proceedings of the 6th IFAC/IFIP/IFORS Symposium on Control, Computers, Communication on Transportation*, pages 245–252, Paris, France.
- Messmer, A. and Papageorgiou, M. (1990). Metanet: A macroscopic simulation program for motorway networks. In *Traffic Engineering and Control*, volume 31, pages 466–470.
- Radecky, M. and Gajdos, P. (2008). Intelligent agents for traffic simulation. In *SpringSim '08: Proceedings of the 2008 Spring simulation multiconference*, pages 109,115, San Diego, CA, USA. Society for Computer Simulation International.
- Robertson, D. and Bretherton, R. (1991). Optimization networks of traffic signals in real time—the scoot method. In *IEEE Transactions on Vehicular Technology*, volume 40, pages 11–15.
- Xiao-Fan Zhi, Zhi-Cheng, L. S.-Y. Z. Y.-p. Z. (2008). An intersection-centric priority algorithm for traffic simulation based on multi-agents. In *NCM archive Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management*, volume 1, pages 189–194. IEEE Computer Society.
- Zhang, F., Li, J., and Zhao, Q. (2005). Single-lane traffic simulation with multi-agent system. In *IEEE Conference on Intelligent Transportation Systems*, pages 1183 – 1187.