Computing and Informatics, Vol. 28, 2009, 1001–1012, V 2009-Mar-2

HISTORY-BASED SELF-ORGANIZING TRAFFIC LIGHTS

Juan C. Burguillo-Rial, Pedro S. Rodríguez-Hernández
Enrique Costa-Montenegro, Felipe Gil-Castiñeira

Department of Telematic Engineering
University of Vigo
36310-Vigo, Spain
e-mail: {`jrial, pedro, kike, xil`}`@det.uvigo.es`

Revised manuscript received 9 December 2008

Abstract. Managing traffic in cities is nowadays a complex problem involving considerable physical and economical resources. Multi-agent Systems (MAS) consist of a set of distributed, usually co-operating, agents that act autonomously. The traffic in a city can be simulated by a MAS with different agents, cars and traffic lights, that interact to obtain an overall goal: to reduce average waiting times for the traffic users. In this paper, we describe an agent-based simulator to model traffic in cities. Using this simulator, we present a self-organizing solution to efficiently manage urban traffic. We compare our proposal with recent approaches, providing better results than classical and alternative self-organizing methods, with lower resources and investments.

Keywords: Self-organization, adaptive systems, urban traffic control, traffic modeling, multi-agent systems, NetLogo

Mathematics Subject Classification 2000: 68T05 – Learning and adaptive systems

1 INTRODUCTION

In nowadays cities, traffic has become a problem of time, energy, patience and resource consuming [19]. While there is no solution when the traffic density is very high, there are several approaches to manage traffic, each of them involving a different amount of resource investments and providing a different level of satisfaction for drivers, pedestrians and traffic managers.

Traditional approaches for traffic management attempt to optimize the solution for static situations, normally considering hourly basic configurations of traffic density. However, these configurations can abruptly change and traditional methods are not able to automatically consider those changes [1]. Usually they are based in optimization approaches considered for a set of typical traffic densities and topologies, since the space of solutions to model every possible situation is very large.

Advanced Traffic Management Systems (ATMS) use learning methods to adapt the phases of traffic lights, normally using a central computer [6] or a hierarchical computer level [14]. However, high cost, their complexity and the use of proprietary methodologies that impede the migration from a particular methodology to another may be considered as drawbacks of present ATMS. Besides, they usually require maintenance done by specialists and provide different solutions particularized for every city.

As the first contribution, we have extended and improved an urban traffic scenario already provided in the NetLogo [17] distribution. NetLogo is a multi-agent [18] modeling environment for simulating natural and social phenomena.

We also present a self-organizing solution for Urban Traffic Control (UTC) based in the work described in [7]. The results provided in that paper are encouraging but the proposed implementation seems difficult, expensive or even technically unfeasible. So, the second contribution of this paper is a simpler history-based self-organized method (HB-SOTL), which is much easier to implement than SOTL in [7] with current technology, for instance by means of rubber bands or buried loop detectors.

This paper is organized as follows: Section 2 reviews some related work. Section 3 introduces some basic concepts. Section 4 presents our city model, and Section 5 explains our proposal for Urban Traffic Control by self-organizing traffic lights. Finally, Section 6 concludes the paper and draws some future work.

2 RELATED WORK

There are different approaches for traffic modeling in the literature. Reference [12] is one of the first studies. It takes concepts from chemical non-equilibrium systems and applies them to complex social systems. Reference [8] examines common phenomena and laws of the dynamic behavior of traffic, as well as traffic flow models. In [9] they use a gas-kinetic traffic model. Reference [10] presents an Enskog-like kinetic traffic flow equation and it derives fluid dynamic equations. In [4], the authors model vehicular traffic as a system of interacting "physical particles", which offers the possibility to study various fundamental aspects of true non-equilibrium systems.

The following references use cellular automata to model traffic effectively, and they are closer to our research. In [2], there is a simple model that describes traffic flow in two dimensions, with two phases: a low-density dynamic phase (in which all cars move at maximum speed) and a high-density jammed phase (in which they all get stopped). Self-organization effects are also studied and discussed. In [11] there is a stochastic discrete automaton model for freeway traffic. In [5], the authors propose a new cellular automata model for vehicular traffic in cities, by combining borrowed ideas from the two previous references.

More recent models employ multi-agent systems. In [3], the authors simulate the traffic in metropolitan areas, where each traveler is considered individually. Reference [16] describes and tests a multi-agent decision problem for optimal traffic light control in a green light district simulation. In [13], the applicability of autonomous intelligent agents to adapt traffic control to changing environments is investigated.

Concerning self-organization, the proposal in [7] treats traffic as an adaptation problem, where every traffic light adapts its green cycle to its current local conditions. This way, macro-level patterns emerge from the self-organized behavior of these elements. This article has been the basis and main motivation for our present work.

Interested readers can find an extended discussion and review in [1].

3 MAS, SELF-ORGANIZATION AND NETLOGO

3.1 Multi-Agent Systems (MAS)

Before introducing Multi-agent Systems (MAS), we need to define what we understand by an agent. Unfortunately, there is no general agreement in the research community about what an agent is. Therefore we cite a general description [18], and according to it, the term agent refers to a hardware or (more usually) software-based computer system characterized by the well-known properties: *autonomy*, *social ability*, *reactivity*, and *pro-activeness*. There are some other attributes that can be present, but usually they are not considered as a requisite: *mobility*, *veracity*, *benevolence*, *rationality* and *adaptability* (or *learning*) (see [18] for more details).

A system consisting of an interacting group of agents is called a Multi-Agent System (MAS), and the corresponding subfield of artificial intelligence (AI) that deals with principles and design of MAS is called distributed AI.

3.2 Self-Organization

The self-organization concept has been used in many areas with different meanings, see [7, 15], but an area specially interesting is multi-agent systems. We consider a system as self-organized if its elements *interact* to achieve a global behaviour that is not imposed by a single or few elements. Thus, this behavior *emerges* from the interaction among those elements, and allows the system to change its organization without explicit external interaction during its execution time.

As stated in the previous paragraph, the self-organizing concept is usually coupled with the concept of emergence: the situation in which great scale patterns *emerge* from the interaction of independent individuals. Emergence is a suitable context to design complex systems, that cannot be controlled in a centralized way, or systems that pretend to avoid such centralized control. Therefore, the idea is to design systems with suitable emergent behaviors, properties and/or functionalities.

3.3 NetLogo

We implemented our city model using NetLogo [17], a popular multi-agent modeling environment for simulating natural and social phenomena. It is particularly well suited for modeling complex systems.

NetLogo is a 2-D world made of agents that simultaneously carry out their own activity. There are three types of agents: *Patches* (stationary agents that make up the background or "world"), *Turtles* (agents that move around on top of the patches, and *The Observer* (that manages everything going on in the world).

NetLogo uses a simple scripting language to define the systems, and it also has a user-friendly graphical interface.

4 CITY MODEL

4.1 Previous Models

The NetLogo distribution [17] includes two simple traffic models. The first is named *Traffic Basic* and demonstrates how traffic jams can form even without any "centralized cause". Using *Traffic Basic* a small city with traffic lights is modeled in *Traffic Grid*, also included in the NetLogo distribution. It allows the user to explore traffic dynamics and develop strategies to improve traffic and to understand the different ways to measure the quality of the traffic.

Using the Traffic Grid model as a starting point, a more complex model is presented in [7], called Self-Organizing Traffic Lights (SOTL). Cars flow in a straight line, eastbound or southbound by default. Each crossroad has traffic lights that only allow traffic flow in one of the arteries that intersect it with a green light. Yellow or red lights stop the traffic. The light sequence for a given artery is green-yellow-red-green. Cars have the size of a patch and simply try to drive at a maximum speed of a "patch" per time step, but they stop when a car or a red or yellow light is in front of them. Time is discrete, but space is continuous. The user can change different parameters, such as the number of arteries or cars. Different statistics are shown: the number of stopped cars, their average speed, and their average waiting times. In this scenario, the author presents three self-organizing methods for traffic light control outperforming traditional ones, since the agents are "aware" of changes in their environment, and therefore they can adapt to new situations.

4.2 City Model Improvements

Given the previous model of the traffic in a city, described in [7], we have made some improvements to represent a more realistic scenario.

In the previous model, the roads have a single lane and direction, and there are only two directions by default, south and east, although we can obtain four by adding north and west. A car only changes road or direction depending on a probability (*prob-turn*), which means that the cars move at random. Also, in the previous model, a torus world was used by default, i.e., when a given car arrives at the end of the scenario, the same car will appear at the opposite side.

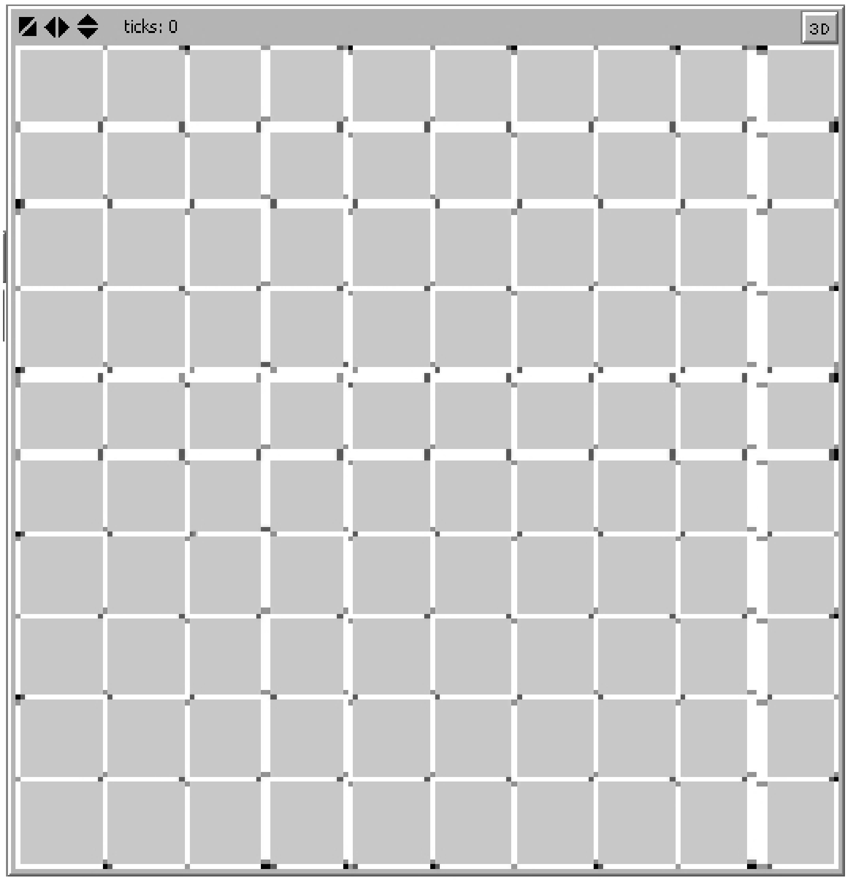

Fig. 1. The city model

To increase realism, we remove the torus and impose four directions (north, east, south and west). Also, a by-pass road is created to improve traffic, which is the outermost in the scenario.

We have also changed the car creation and elimination scheme. Now, for every car, we define a source (a random road patch) and a destination (another random road patch), such that every car is created at a source, and it moves (following the shortest path) to its destination, where it is eliminated. The source and destination may be outside the world. Besides, cars can park at a certain place and then drive to another destination from the parking place.

We have added the possibility of bidirectional roads and roads with two lanes in the same direction, controlled with a slider in the Netlogo interface.

In order to avoid deadlocks at the intersections, a deadlock algorithm has been implemented. If a given car at an intersection has not moved after a given time, it tries to change direction in order to keep moving and to try to exit the deadlock. This movement could affect other cars and help finishing the current deadlock.

Due to all these improvements, specially the possibility of an origin and a destination and bidirectional roads, a complex algorithm to guide the cars is needed. Whenever a car is in a patch that belongs to an intersection (it belongs simultaneously to a horizontal and a vertical road), it runs the guiding algorithm in order to know whether a change of direction is necessary, before moving on. If not, the car will keep the same direction, at least until the next intersection.

As seen in Figure 1, with these changes we obtain a scenario where we can notice the different widths of the streets, depending on whether they are bidirectional and single or dual-lane streets. We can also see the distribution of the traffic lights, and the by-pass road surrounding the city. This extended model has been used in the scenarios simulated by us and described in the next section.

5 SELF-ORGANIZING URBAN TRAFFIC CONTROL

5.1 Self-Organizing Traffic Lights (SOTL)

The proposal in [7] treats traffic as an adaptation problem, as traffic conditions change constantly, and every traffic light adapts to its current local conditions. This way, macro-level patterns emerge from the self-organized behavior of these elements.

In such model, the self-organization of traffic lights, to improve traffic flow, is based on a simple measure of local conditions: in every intersection, the number of cars approaching the red light (k_i) are counted (irrespective of their status or speed).

From this measure, two adaptive self-organized methods are proposed:

SOTL-request: when k_i reaches a threshold, lights change.

SOTL-phase: to prevent fast switching of lights, an additional condition is set: a minimum time must have elapsed since last change.

These two methods are compared with two non-adaptive methods:

marching: all green lights are either "vertical" or "horizontal".

optim (green wave): this method is an improvement of the marching one. It sets phases to traffic lights, so "green corridors" are created, i.e., a car driving at maximum allowed speed will find the following traffic lights in green.

The results in [7] show that *SOTL-request* presents the best performance for low traffic densities, whereas it is very inefficient for high traffic densities, due to constant switching. On the other hand, *SOTL-phase* performance is nearly as good as *SOTL-request* for low densities, and very similar to (*marching*) for high densities.

5.2 History-Based SOTL (HB-SOTL)

The results attained by the self-organized methods proposed in [7] are encouraging, but their implementation seems very hard, as counting the cars "approaching" the red light would be very expensive or even unfeasible.

So, we have devised another simpler self-organized method, in which the duration of the next green cycle of every light is directly proportional to the number of cars which crossed the intersection in the last green cycle. We have called it History-Based Self-Organized Traffic Lights (HB-SOTL).

More precisely, the mentioned duration of the next green cycle is proportional to the traffic density in the intersection, i.e. the quotient of the number of crossing cars in the previous green cycle and the duration of that cycle.

This measure is much easier to implement with current technology: for instance, by means of rubber bands (also known as pneumatic road tubes) or buried inductive loop detectors. The rubber band is installed perpendicular to the traffic flow direction. It sends a burst of air pressure along the rubber tube when a vehicle's tires pass over the tube. The inductive loop detector is used often for permanent installations and it consists of a coil with a few windings of copper wire which are installed under the road surface.

5.3 History-Based SOTL Results

We have implemented our method in NetLogo and then we compared it with the four methods described in [7], trying to keep the same simulation parameters, but with some changes to have a more realistic model:

- We have used our improved city model, with 25 % of the streets bidirectional, 25 % of the arteries (both belonging to unidirectional and bidirectional streets) have two lanes of traffic running in the same direction, 10 % of the cars have their origin out of the city, and 10 % of the cars have their destination out of the city.
- Instead of a grid of 10×10 arteries, we have used greater model: a grid of 19×19.

- We set the same number of vertical and horizontal streets, among the horizontal streets, the same number of east and westbound; and, among the vertical streets, the same number of north and southbound.

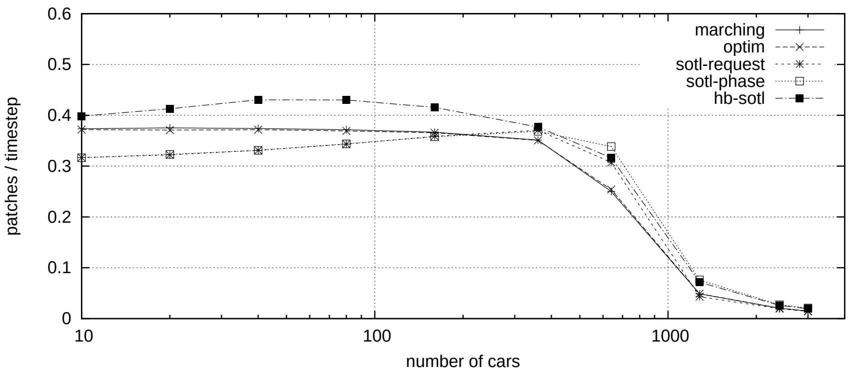

Fig. 2. Average speed of cars; 19 × 19 grid

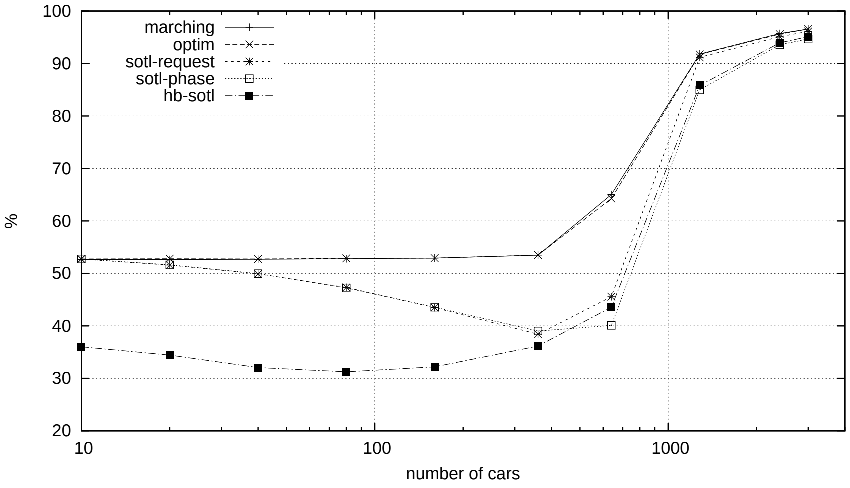

Fig. 3. Average stopped cars; 19 × 19 grid

The results show that the performance of HB-SOTL is similar or better than the other self-organized or classical approaches. In particular, compared to SOTL-phase:

- As shown in Figure 2, the average speed of cars is higher (43 % of maximum speed vs. 34 % in an 80 cars scenario) for low traffic densities, while it is slightly lower (7.2 % vs. 7.6 % in a 1 280 cars scenario) for high traffic densities.

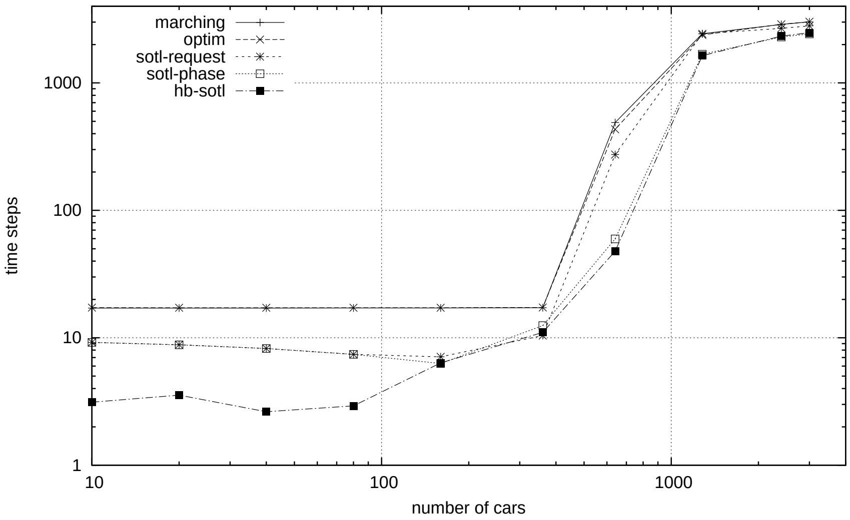

Fig. 4. Average waiting time of cars; 19 × 19 grid

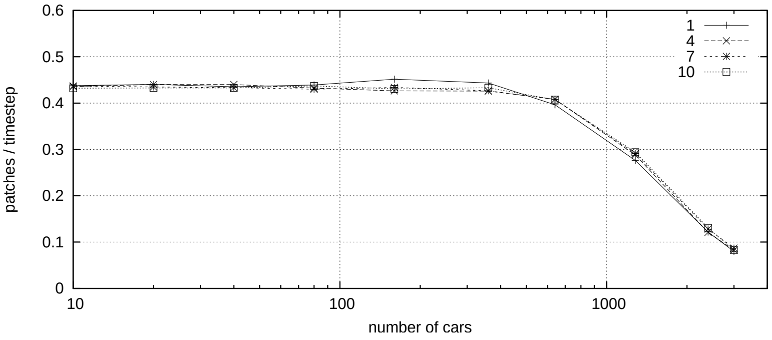

Fig. 5. Average speed of cars of HB-SOTL using several history cycles; 19 × 19 grid

- The average number of stopped cars is significantly lower (31 % of the cars vs. 47 % in an 80 cars scenario) for low densities, and only slightly higher (86 % vs. 85 % in a 1 280 cars scenario) for high densities (Figure 3).
- The average waiting time of cars is much better (3 time steps vs. 7 in an 80 cars scenario) for low densities and very similar for high densities (1 636 vs. 1 680 in a 1 280 cars scenario), as seen in Figure 4.

We also consider the effect of counting not only the history over the last cycle, but the effect of using the average number of cars over the last n cycles. Figure 5 considers several simulation runs, of the HB-SOTL method, using a history record from 1 to 10 cycles, and their influence over the average speed. As the reader can see, it has no observable effect. It means that using a very simple device, able to count the number of cars over the last cycle, is enough to obtain good results. This can be done very easily with the present technology.

6 CONCLUSIONS

In this paper we have extended and improved a previous model for describing Urban Traffic Control (UTC) using a multi-agent system and applying a self-organizing methodology. We represent a typical city with horizontal and vertical roads, uni- or bidirectional, and with different numbers of lanes. We model cars as agents that wish to arrive to a destination point from a starting point, via the shortest path. We have tested our city model in a multi-agent simulator developed in NetLogo.

We also present a history-based self-organizing solution (HB-SOTL) for managing urban traffic, which is cheap and easy to implement with current technology. We compare our proposal with some other classical ones and with self-organized approaches. The results are better in practically all the conditions measured.

As future work, we plan to include improvements in the city model, like empty areas in the scenario, slanting streets and roundabouts. Concerning our HB-SOTL proposal, we want to refine the parameters considered for the traffic lights cycle and include the modeling of pedestrians.

We are also adapting our MAS city model to manage car-to-car communications (C2C) using the IEEE 802.11 and IEEE 1609 standards. The application to C2C communications will allow us to analyze the characteristics and performance of information sharing among vehicles in a city environment.

REFERENCES

[1] Bazzan, A. L. C.: A Distributed Approach for Coordination of Traffic Signal Agents. Autonomous Agents and Multi-Agent Systems, 2005, No. 10, pp. 131–164.

[2] Biham, O.—Middleton, A.—Levine, D.: Self-Organization and a Dynamical Transition in Traffic-Flow Models. Phys. Rev. A, Vol. 46, 1992, No. 10, pp. 6124–6127.

[3] Cetin, N.—Nagel, K.—Raney, B.—Voellmy, A.: Large-Scale Multi-Agent Transportation Simulations. Computer Physics Communications, Vol. 147, 2002, pp. 559–564.

[4] Chowdhury, D.—Santen, L.—Schadschneider, A.: Statistical Physics of Vehicular Traffic and Some Related Systems. Physics Reports, Vol. 329, 2000, No. 4, pp. 199–329.

[5] Chowdhury, D.—Schadschneider, A.: Self-Organization of Traffic Jams in Cities: Effects of Stochastic Dynamics and Signal Periods. Phys. Rev. E, Vol. 59, 1999, No. 2, pp. 1311–1314.

[6] Federal Highway Administration: Traffic Control Systems Handbook. U. S. Department of Transportation. 1998.

[7] Gershenson, C.: Self-Organizing Traffic Lights. Complex Systems, Vol. 16, 2004, pp. 29.

[8] Traffic and Granular Flow. In: Helbing, D.—Hermann, H.—Schreckenberg, M.—Wolf, D. E. (Eds.), Springer-Verlag, 1999.

[9] Helbing, D.—Treiber, M.: Gas-Kinetic-Based Traffic Model Explaining Observed Hysteretic Phase Transition. Phys. Rev. Lett., Vol. 81, 1998, No. 14, pp. 3042–3045.

[10] Klar, A.—Wegener, R.: Enskog-Like Kinetic Models for Vehicular Traffic. Journal of Statistical Physics, Vol. 87, 1997, No. 1–2, pp. 91–114.

[11] Nagel, K.—Schreckenberg, M.: A Cellular Automaton Model for Freeway Traffic. Journal de Physique I, Vol. 2, 1992, No. 12, pp. 2221–2229.

[12] Prigogine, I.—Herman, R.: Kinetic Theory of Vehicular Traffic. American Elsevier Pub. Co, 1971.

[13] Roozemond, D. A.: Using Intelligent Agents for Pro-Active, Real-Time Urban Intersection Control. European Journal of Operational Research, Vol. 127, 2001, No. 2, pp. 293–301, availaible on: `http://ideas.repec.org/a/eee/ejores/v131y2001i2p293-301.html`.

[14] TYCO Integrated Systems: The Sydney Co-ordinated Adaptive Traffic System (SCATS). 2008, available at: `http://www.traffic-tech.com/pdf/scatsbrochure.pdf`.

[15] di Marzo Serugendo, G.—Gleizes, M. P.—Karageorgos, A.: Self-Organisation and Emergence in MAS: An Overview. Informatica, Vol. 30, No. 1, 2006, pp. 45–54.

[16] Wiering, M.—Vreeken, J.—van Veenen, J.—Koopman, A.: Simulation and Optimization of Traffic in a City. IEEE Intelligent Vehicles Symposium, pp. 453–458, June 2004.

[17] Wilensky, U.: NetLogo: Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL. Availaible on: `http://ccl.northwestern.edu/netlogo/`, 2007.

[18] Wooldridge, M.—Jennings, N. R.: Intelligent Agents: Theory and Practice. Knowledge Engineering Review, Vol. 10, 1995, No. 2, pp. 115–152.

[19] Bekiaris, E.—Nakanishi, Y. (Eds.): Economic Impacts of Intelligent Transportation Systems: Innovations and Case Studies. Elsevier, 2004.

Juan C. Burguillo-Rial received the M. Sc. degree in 1995 and the Ph. D. degree (cum laude) in 2001 from the University of Vigo, Spain; both in telecommunication engineering. He is currently Associate Professor at the Department of Telematic Engineering at the University of Vigo. He has participated in several R & D projects in the areas of telecommunications and software engineering, and has written over fifty papers in international refereed journals and conference proceedings. His research interests include intelligent agents, distributed artificial intelligence and optimization, and telematic services.

Pedro S. Rodríguez-Hernández received the M. Sc. degree in 1989 from Polytechnic University of Madrid, Spain, and the Ph. D. degree (cum laude) from University of Vigo, Spain, in 1998. He is currently an Associate Professor at the Department of Telematic Engineering at University of Vigo. He has written several papers in international journals and conferences. He has participated in diverse R & D projects in the areas of telecommunications and software engineering. His current research interests include intelligent agents, distributed optimization, real-time systems and high-performance embedded systems.

Enrique Costa-Montenegro received the M. Sc. degree in 2000 and the Ph.Ḋ. degree (cum laude) in 2007 from the University of Vigo, Spain; both in telecommunication engineering. He is Assistant Professor since 2002 at the Department of Telematic Engineering, at the ETSE de Telecomunicación University of Vigo, Spain. His research interests include wireless networks, car to car communication technologies, multi-agent systems and peer-to-peer systems. He is author of several articles in international journals and conferences and has participated in diverse R & D projects in these areas.

Felipe Gil-Castiñeira received the M. Sc. degree in 2002 and the Ph. D. degree (cum laude) in 2007 from the University of Vigo, Spain; both in telecommunication engineering. He is Assistant Professor since 2003 at the Department of Telematic Engineering of the ETSE de Telecomunicación, University of Vigo, Spain. His research interests include wireless and car to car communication technologies, embedded systems, nomadic devices, wireless sensor networks and ubiquitous computing. He is author of several articles in international journals and conferences and has participated in diverse R & D projects.