Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

An accelerated-time simulation for traffic flow in a smart city[☆]

Q1 José L. Galán-García [*], Gabriel Aguilera-Venegas, Pedro Rodríguez-Cielos

Department of Applied Mathematics, University of Málaga, Spain

ARTICLE INFO

Article history:
Received 7 September 2013
Received in revised form 13 November 2013

Keywords:
Accelerated-time simulations
Smart cities
Car traffic
CAS

ABSTRACT

Traffic control is one of the most important problems related with urban development. Current trends for traffic control are based on the use of smart traffic lights and signals as a part of smart cities' projects. Different cities are currently involved in the design and implementation of smart traffic control. Since the cost of physically installing these systems is very high, in terms of both money and resources, accelerated-time simulations of traffic flow using smart traffic lights and signals significantly reduce these costs.

In this work we present a new model for accelerated-time simulations for traffic flow within. The philosophy of this model is based on previous works of the authors, where accelerated-time simulations for car traffic in a motorway or a roundabout and baggage traffic in an airport were developed. The philosophy of this model combines ideas from cellular automata and neural network theories, obtaining a mixed model.

This system was developed using a Computer Algebra System (CAS) called MAXIMA for mathematical computations and a JAVA based interface for graphical display. MAXIMA allows the system to support the use of ad hoc distribution functions for the different events dealt with in the simulations. The interface provides a friendly framework for entering input data and visualizing the simulations, providing also some statistical data.

1. Introduction

Information and Communication Technologies (ICTs) play a very important role in the development of sustainable cities. Smart city is a very broad concept which includes not only physical infrastructure but also human and social factors [2].

One of the main aspects of smart cities is a good control of the traffic flow within the city. Traffic jams or slow traffic are not only pollution and economic problems but also provoke frustration in drivers as well as pedestrians. The use of smart traffic lights and signals is one of the most important techniques that smart cities use to deal with these problems.

Smart traffic lights and signals are interconnected. Each sensor detects a different parameter of the traffic flow (speed of cars, density, waiting time, a traffic jam, etc). The system makes decisions according to the values of these parameters and gives the appropriate instructions to the lights and signals.

A good example of using smart traffic lights and signals was designed by Traffic21 project [3]. This project proved, among other things, that using smart traffic lights and signals reduced emissions by over 20%. To achieve this result, SURTRAC, a pilot implementation of an adaptive traffic signal control system, was implemented. This control system was installed for a nine-intersection road network in Pittsburgh, Pennsylvania (USA).

[☆] Expanded version of a talk with the same title presented at the *FEMTEC'2013* Conference (Las Vegas) and a later version presented at the *ACA'2013* Conference (Málaga) entitled *Simulating Car Traffic with Smart Signals using a CAS* Galán et al. (2013) [1]. Q2

[*] Corresponding author. Tel.: +34 95 2 13 27 64; fax: +34 95 2 13 27 66.

E-mail addresses: jl_galan@uma.es (J.L. Galán-García), gabri@ctima.uma.es (G. Aguilera-Venegas), prodriguez@uma.es (P. Rodríguez-Cielos).

0377-0427/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cam.2013.11.020

Please cite this article in press as: J.L. Galán-García, et al., An accelerated-time simulation for traffic flow in a smart city, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.11.020

Testing of a smart traffic light and signal system has a very high cost. These tests produce not only high costs in resources but also in traffic problems while physically implementing the system. Simulation techniques that help in the design of the system can drastically reduce these costs.

This work will deal with simulation techniques. Specifically, we introduce the ATISMART model, an accelerated-time model to simulate car traffic in a city using smart traffic lights and signals. The aim of the ATISMART model is to easily simulate a wide range of situations in order to test the system before its physical implementation. To achieve this goal, the ATISMART model has been programmed using a CAS that deals with exact and symbolic computations.

In Section 2 the main background related to this work is presented. Section 3 is the description of the ATISMART model while Section 4 shows its implementation. In Section 5, different results from using ATISMART in an example are provided. Finally, the conclusions and directions for future work are given in Section 6.

2. Background

Some of the foundations of the ATISMART model are based on ideas from: Cellular Automaton (CA) and Neural Network (NN) theories.

The starting point of CA gets us back to the decades of forties and fifties when Stanislaw Ulam and John von Neumann first introduced the idea [4]. But one of the best known example of CA is John Conway's game of "life" [5] which simulates the behaviour of a group of cells. The way that this group of cells evolve, is determined by the starting configuration and four basic rules. These four rules establish the state of each cell depending on its own and its neighbouring cells' state. Q3

An easy way to understand how a CA can be applied to simulate movement, is the rule 184 of Wolfram [6]. This rule establishes if a cell will or will not be occupied in the next step depending on the state of the current, previous and following cells. Specifically, there are 8 possible situations displayed in binary from 000 to 111, where 0 or 1 means that the corresponding cell is empty or occupied. The first, second and third bits correspond to the previous, current and following cells, respectively. The result of applying rule 184 to a three-bit number is a bit which tells if the central position will or will not be occupied in the next step. The following table shows how the rule is applied for the 8 different possibilities.

Input	111	110	101	100	011	010	001	000
Output	1	0	1	1	1	0	0	0

For example, after applying the rule 184 to 101, a 1 is obtained since the current position (the central one) is empty and the previous is occupied and the object in this cell wants to move one step to the right. After this one-step movement, the central position will be occupied. In a similar way, after applying the rule to 011, a 1 is obtained since the current position is occupied but cannot move to the right because the next position is occupied and, therefore, the cell remains occupied.

The name of the rule comes from the binary number obtained in the output row of the previous table, that is, 184 is 10111000 in binary.

A more sophisticated CA related with movement was given by Nagel and Schreckenber [7] (the NaSch model). They introduced a stochastic discrete model of freeway traffic.

Knospe et al. [8] stated that the NaSch model could properly simulate macroscopic characteristics of freeway traffic. However, they also stated that the comparison of simulation results with empirical data on a microscopic level is not satisfactory. To overcome this situation, they introduced a new model (the KSSS model) whose rules take into account characteristics such as random braking in addition to velocity, position and normal braking which were already considered in the NaSch model. Therefore, the KSSS model uses a finer discretization than that used in the NaSch model.

With regard to our own background on the topic of this paper, we introduced a new model for car traffic simulation [9]. This new model uses the KSSS model as its starting point but introduces new characteristics such as the use of car indicator signals and more than one traffic line. Our model also considered typical concepts from neural network theory such as the use of state vectors and minimization of the objective function. In this paper, the model is applied as an example of simulating car traffic behaviour in a roundabout with several traffic lines and different inputs and outputs.

In [10], the GRAM model was introduced as an extension of the previous model and was used to generate car traffic simulations in motorways as well as roundabouts. In that paper the description of the GRAM model was presented, together with a computer implementation of the model.

Another related work is the ATISBAT model [11] that simulates the baggage traffic in the handling system of an airport from the check-in desks to the aircraft.

The computer implementations of the GRAM and ATISBAT models were developed using a CAS in combination with graphical interfaces developed in Java. This blending of Java with a CAS was first used by us in [12] where we introduced a graphic interface for generating counterpoints for a given melody.

Other authors have also used accelerated-time simulations for other applications. Some of these simulations are described in [13,14]. The first deals with simulations for departing passengers' flow in airport terminals while the second introduces accelerated-time simulations for a dedicated freight double-track railway line.

Please cite this article in press as: J.L. Galán-García, et al., An accelerated-time simulation for traffic flow in a smart city, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.11.020

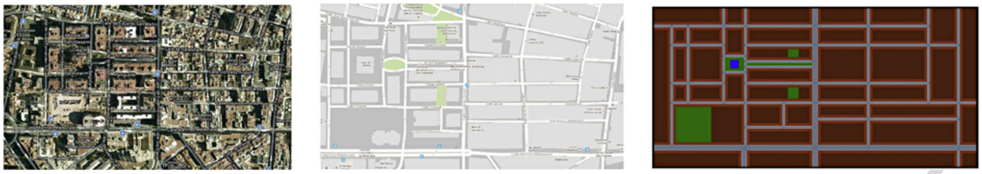

Fig. 1. Three maps of a zone in Málaga.

3. The ATISMART model

The ATISMART model has been designed to simulate car traffic in a city using smart traffic lights. Based on the traffic flow conditions, these smart signals can change the duration of the red and green periods. Furthermore, if needed, the direction of traffic flow on the streets can be changed in order to improve the overall traffic flow.

In order to describe the ATISMART model, we now consider three different factors: the city map, the cars and the smart signals.

3.1. The city map

The ATISMART model has been developed for a general use in any city. For example, it has been applied to a zone of the city of Málaga. The zone chosen can be seen in Fig. 1. The left side of the figure contains a picture taken from Google Earth; while in the middle, contains a drawing taken from Google Maps. The right side, contains an adapted map used for the graphic interface of the ATISMART implementation.

In order to store the information on the adapted map, a matrix M, with the data of the different elements of the map (lines, senses, intersections, traffic lights, entrances and exits) is used. However, the model uses a graph structure of the map to calculate the path of the cars.

Cars enter and exit the system through any of the different entrances and exits. Both, the entrances and exits are located at points on the border of the map. These points are entrances or exits depending on the sense of the streets. Q4

The initial senses assigned to the different lines on the map correspond to the those currently existing on the streets in the city. Also, traffic lights have been placed in the intersections as they are on the real map. Furthermore, the periods of the red and green lights are initially set to the existing periods. In any case, as will be shown later, both the senses of the streets and the red/green periods of each traffic light can be dynamically changed.

3.2. Cars

Each input has an associated specific probability distribution in order to simulate the entrance of cars in the system. By default, a Poisson distribution is assigned to each input. For input I_j, the corresponding λ_j-parameter has been set to the mean of cars accessing the map via entrance I_j. The ATISMART model is very flexible and it allows to dynamically change these λ_j-parameters. Furthermore, it is also possible to change the probability distribution from a Poisson to any of the classical well known distributions and even to choose an "ad hoc" distribution, defined by the user.

Once a car C_i enters the system, a random exit, O_k, is assigned. ATISMART also allows one to introduce a car by specifying its entrance and exit points. In any case, this information is stored in its state vector SV_i which contains the following information:

1. SV_{i1} is the X coordinate of C_i in M, the matrix of the map.
2. SV_{i2} is the Y coordinate of C_i in M.
3. $SV_{i3} = v_i$ is the velocity of C_i.
4. SV_{i4} is a flag which indicates if C_i has moved in the current step.
5. SV_{i5} is the assigned output for C_i.
6. SV_{i6} stores the colour of C_i to be shown in the graphical simulation.
7. SV_{i7} is a flag which indicates C_i has to recalculate its path.
8. SV_{i8} stores the next intersection identification where the car, C_i, is approaching.

Once the car, C_i, is in the system, it has to compute its path to get the location of the assigned exit (stored in SV_{i5}). In order to achieve the best path, Dijkstra's algorithm [15] is used. Since the street senses in the map can change (modifying the graph) while C_i is moving towards its exit, once the car reaches the next intersection, it recomputes its path. Therefore, C_i just only needs to store the location of next intersection.

The car C_i is always moving towards its exit unless the next position is occupied by another car or it reaches a red traffic light. Once the next position is empty or the traffic light changes to green, C_i starts moving again.

If more than one car reaches an intersection at the same time, the movement is produced according to the specific traffic rules of that intersection.

Please cite this article in press as: J.L. Galán-García, et al., An accelerated-time simulation for traffic flow in a smart city, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.11.020

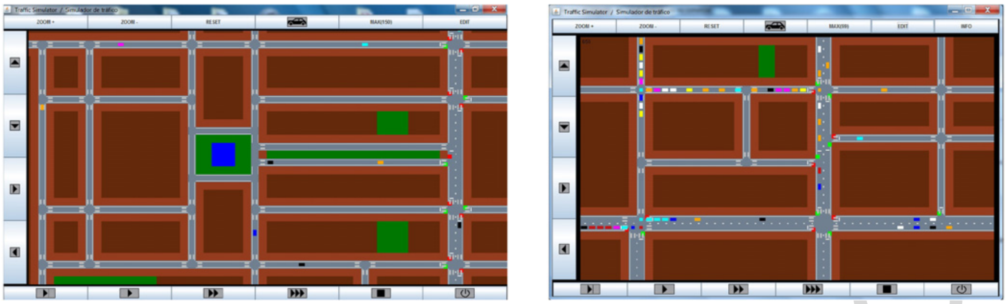

Fig. 2. Cars in the system.

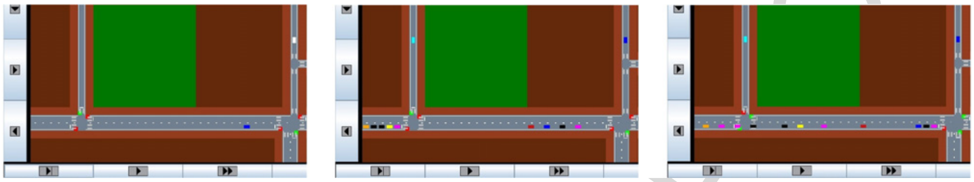

Fig. 3. Periods of traffic lights.

Since cars should not block intersections, if a car happens to remain stopped in an intersection for some predetermined time, it disappears from the system (simulating that a tow truck moved the car).

Once a car, C_i, reaches its exit, it leaves the system.

In the left picture of Fig. 2, some moving cars appear on the map while in the right picture, some cars are moving and others are stopped because of red traffic lights or occupied forward positions.

3.3. The smart signals

ATISMART allows both street sense changes and modifications of the period of red and green lights. Specifically, the sense of any part of a street between two intersections can be changed and the periods of red and green lights for any traffic light can be modified.

The three pictures in Fig. 3 have been taken in the same step of the simulation (step 42). In the first case, the picture on the left shows the state of the simulation in a normal situation, that is, values by default, so cars enter from the left with a Q5 Poisson distribution with $\lambda = 1$. No problems are seen in this picture. In the second picture, the one in the centre, λ has been increased to 2, so more cars enter the street from the left. A traffic jam can be seen in this picture. It is produced by the traffic light. In the third picture, λ remains equal to 2, but the period of the traffic light has been changed to give priority to the horizontal street instead of the vertical one. The result is that no traffic jam has occurred.

4. Implementing ATISMART

The ATISMART model has been implemented combining the ability of a CAS (specifically, Maxima) to do mathematical Q6 computations and the graphical power of a programming language such as Java. This idea of combining Java with a CAS was also used in previous works by the authors [10–12]. In these papers, a more detailed description on the convenience of this combination can be found.

In summary, the main reasons for using a CAS as the mathematical core of the model, together with Java for the Graphical User Interface (GUI) are:

- CASs are powerful tools to deal with mathematical computations, but all of them lack an application which allows the development of a GUI for help in visualizing the results and allowing the user to easily interact with the system. Java was chosen to overcome this deficiency.
- Both, Maxima and Java exist on several platforms and are publicly available. Furthermore, there exist different libraries which allow the communication between Maxima and Java. The one chosen for the implementation of the ATISMART model is Jacomax (https://www.wiki.ed.ac.uk/display/Physics/Jacomax). Therefore, by combining Java with Maxima, ATISMART can be run on different platforms.
- The authors had developed some packages in a CAS to generate random samples of distributions [16]. These packages are needed to simulate the random processes occurring in the system. For example, the way for a car to enter into the system is controlled by a probability distribution. The choice of a distribution is left to the users discretion, it can even be given “ad hoc”.

Please cite this article in press as: J.L. Galán-García, et al., An accelerated-time simulation for traffic flow in a smart city, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.11.020

- The use of a CAS is needed since some symbolic and exact computations are needed. For example, in order to generate a random value x from an "ad hoc" probability density function f, defined in the interval $[min, max]$, the inverse transform method (described in [16]) is used. This method obtains the random value, x, from the equation $x = F^{-1}(u)$ where F is the cumulative distribution function of f and u is a random value from the continuous uniform distribution $\mathcal{U}(0, 1)$.

  Specifically, these calculations needed are:
  - The cumulative distribution function F is calculated using:
  $$F(x) := \int_{\min}^{x} f(t)\mathrm{d}t.$$
  - Generate a random number u from a continuous uniform distribution $\mathcal{U}(0, 1)$.
  - Solve for x from the equation $F(x) = u$, that is, $x = F^{-1}(u)$. This value, x, is the randomly generated value using the "ad hoc" distribution given.

  In order to perform the inverse transform method, a CAS is needed for the following two symbolic operations: the antiderivative computation to obtain the cumulative function F from f and for finding the inverse value $F^{-1}(u)$ to obtain the quantile function F^{-1}.

The source code of both, the Java and the Maxima modules of ATISMART and a readme file with installation instructions can be downloaded without charge from http://www.matap.uma.es/jlgalan/ATISMART/.

5. An example of using ATISMART

In this section, an example of an accelerated-time simulation with and without smart facts and the corresponding results are described.

The map drawn in Fig. 1 corresponds to a zone close to "La Rosaleda", the stadium of the Málaga football club. Specifically, the stadium is located in the north of the given map. Therefore, when a football match is over, the traffic flow moving from the north greatly increases the normal traffic flow. As a result, many traffic jams occur. We want to examine whether a single change in the sense in a street, could lead to a reduction in the congestion.

To simulate this situation, λ-parameters of Poisson distributions have been increased to a value of 3 in each of the northern entrances of the map. In order to quantify the traffic congestion, we define that a traffic jam at an intersection occurs (in the following, we will say jam) when a car in the intersection cannot move given the present conditions but under the normal conditions the car can move to the next position. For example, if a car has a green light in front but the intersection is occupied, a jam occurs.

Two different simulations in the same conditions, except that a change in the sense in one street (see Fig. 4) have been done in the second simulation. With "same conditions" we will say that the simulations have been reproduced using the same steps, the same distributions and parameters (the north inputs following a Poisson distribution of parameter $\lambda = 3$) and the same random numbers. Therefore, all cars in the system have the same entrance and exit.

Fig. 5 displays the cumulative number of jams at intervals of ten steps of both simulations. Series 1 corresponds to the simulation with the real senses in the map while Series 2 shows the results when introducing the sense change in the street marked in Fig. 4.

It can be clearly seen that by introducing just a single sense change in one street, the number of jams has been significatively decreased. This visual first impression can be checked by doing the following small statistical analysis:

Let X be the random variable: "number of jams in 10 steps" for the first simulation (with no sense changes).

Let Y be the random variable: "number of jams in 10 steps" for the second simulation (with the single sense change stated before).

The values obtained for X and Y are:

$$\begin{aligned} X &= \{0, 0, 0, 0, 0, 0, 0, 0, 8, 3, 3, 15, 3, 5, 18, 4, 15, 8, 4, 16, 0, 6, 25, 14, 13, 16, \\ &\quad 8, 17, 7, 2, 25, 10, 5, 8, 0, 0, 9, 14, 7, 21, 2, 19, 22, 6, 13, 14, 11, 21, 16, 24\} \end{aligned}$$

$$\begin{aligned} Y &= \{0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 6, 3, 0, 10, 3, 10, 3, 0, 14, 12, 3, 16, 5, 8, 5, 8, \\ &\quad 27, 14, 12, 5, 3, 3, 14, 13, 6, 3, 0, 3, 14, 3, 4, 11, 5, 13, 11, 6, 14, 6, 8\} \end{aligned}$$

where x_i is the number of jams that has occurred in the first simulation between steps $10 \times (i-1) + 1$ and $10 \times i$ and y_j is the number of jams that occurred in the second simulation between steps $10 \times (j-1) + 1$ and $10 \times j$.

Let $Z = X - Y$. We want to check the normality of Z using the Shapiro–Wilk normality test with a significant level $\alpha = 0.05$. After running this test in the statistical software R (free software downloadable at http://www.r-project.org/), Z can be considered to follow a normal distribution since the p-value obtained is $p = 0.1919$ which is greater than α.

Once the hypothesis of normality has been verified, we now proceed to calculate the confidence interval for the mean μ_z with unknown variance for samples greater than 30 with a significant level $\alpha = 0.05$. Developing the computations in R, the confidence interval for μ_z is $[1.122937, 4.957063]$ with a confidence level of $1 - \alpha = 0.95$. Therefore, with a confidence level of 95%, we can conclude that μ_z belongs to this interval. Since 0 is outside of this confidence interval, we can conclude that $\mu_z > 0$ with $\alpha = 0.05$.

Hence, as expected by looking at Fig. 5, the results obtained in the second simulation are better than the ones from the first simulation at this level of significance.

Please cite this article in press as: J.L. Galán-García, et al., An accelerated-time simulation for traffic flow in a smart city, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.11.020

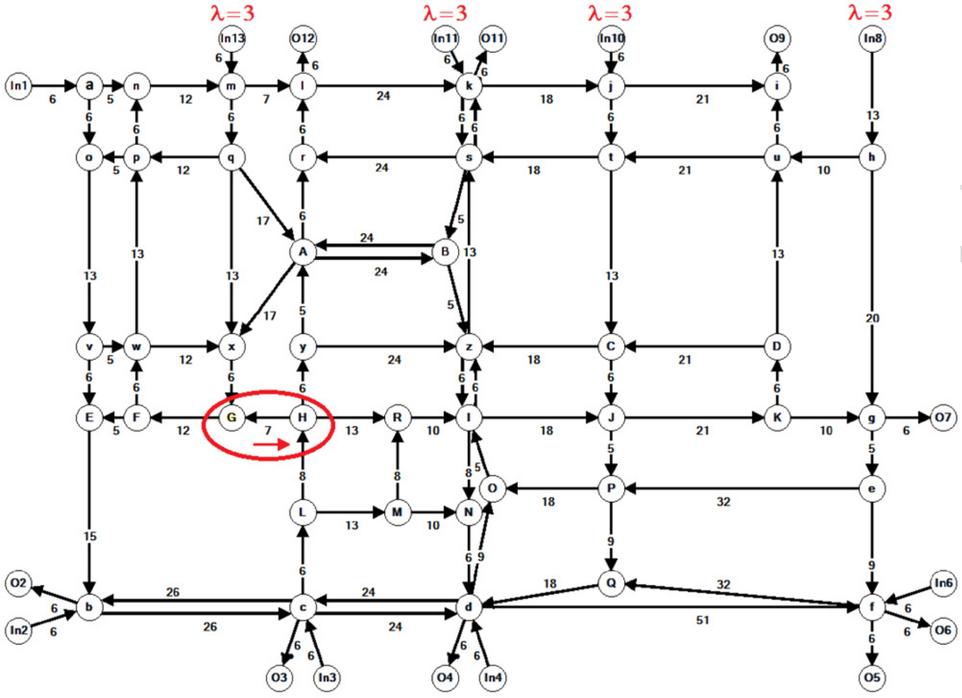

Fig. 4. Graph of the map.

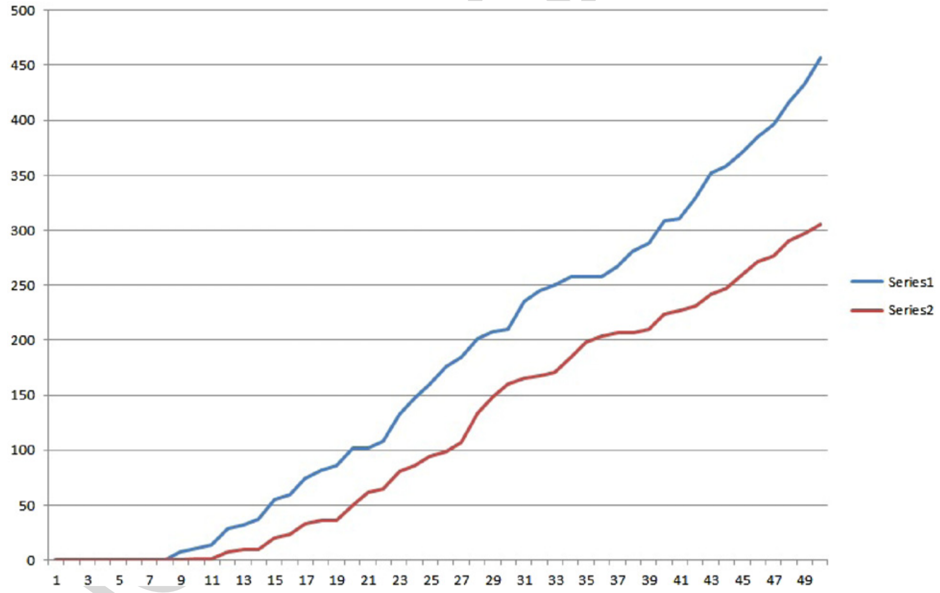

Fig. 5. Test.

6. Conclusions and ideas for future work

Some conclusions obtained after using the new ATISMART model are:

- The ATISMART model has been shown to be a flexible and easy tool to simulate traffic flow in a city using smart signals.
- Simulations developed using ATISMART reveal that both, changing the red/green period of traffic lights and reversing the sense of streets under different traffic conditions, can lead to a better traffic flow.
- Programming with a CAS allows one to deal with exact and symbolic computation. Therefore, not only numerical approximations can be done, but also exact methods.

Please cite this article in press as: J.L. Galán-García, et al., An accelerated-time simulation for traffic flow in a smart city, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.11.020

- The use of GUIs allows the user to interact dynamically with the system. The user can both, visually check what is happening in the simulation and immediately act in order to change some conditions and see the results.

Some ideas for related future work are:

- Adapt the ATISMART model for other accelerated-time simulations such as the design or the improvement of a city bus network that uses smart signals.
- Introduce fuzzy aspects when assigning the path using Dijkstra's algorithm in order to model situations in which the driver does not choose the optimal route.
- Introduce possible changes in the choice of an exit decision by some of the drivers.

References

[1] José Luis Galán, Gabriel Aguilera, José Carlos Campos, Pedro Rodríguez, Simulating car traffic with smart signals using a CAS (Abstract), in: J. L. Galán-García, G. Aguilera-Venegas, Pedro Rodríguez-Cielos (Eds.), Aplications of Computer Algebra ACA'2013 Proceedings. Málaga, 2013, pp. 183. Q7
[2] A. Caragliu, C. Del Bo, P. Nijkamp, Smart cities in Europe, J. Urban Technol. 18 (2) (2011) 65–82.
[3] S.F. Smith, G.J. Barlow, X.F. Xie, Z.B. Rubinstein, Smart urban signal networks: initial application of the SURTRAC adaptive traffic signal control system, in: Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling, Rome, Italy, 2013.
[4] J.v. Neumann, The general and logical theory of automata, in: L.A. Jeffress (Ed.), Cerebral Mechanism in Behavior—The Hixon Symposium, John Wiley & Sons, New York, 1951, pp. 1–31.
[5] M. Gardner, The fantastic combinations of John Conway's new solitaire game life, Sci. Am. 223 (1970) 120–123.
[6] S. Wolfram, A New Class of Science, Wolfram Media, 2002.
[7] K. Nagel, M. Schreckenber, A cellular automaton model for freeway traffic, J. Physique 2 (12) (1992) 2221–2229.
[8] W. Knospe, L. Santen, A. Schadschneider, M. Schreckenberg, Towards a realistic microscopic description of highway traffic, J. Phys. A: Math. Gen. 33 (48) (2000) 477–485.
[9] E. Mérida, G. Aguilera, J.L. Galán, P. Rodríguez, A new neural model for traffic simulation, Lecture Notes in Comput. Sci. 6254 (2010) 471–476.
[10] G. Aguilera, J.L. Galán, J.M. García, E. Mérida, P. Rodríguez, An accelerated-time simulation of car traffic on a motorway using a CAS, J. Math. Comput. Simul. (2012). http://dx.doi.org/10.1016/j.matcom.2012.03.010.
[11] G. Aguilera, J.L. Galán, E. Mérida, P. Rodríguez, An accelerated-time simulation of baggage traffic in an airport terminal, J. Math. Comput. Simul. (2013) submitted for publication. Q8 Q9
[12] G. Aguilera, J.L. Galán, R. Madrid, A.M. Martínez, Y. Padilla, P. Rodríguez, Automated generation of contrapuntal musical compositions using probabilistic logic in derive, J. Math. Comput. Simul. 80 (6) (2010) 1200–1211.
[13] E. Roanes-Lozano, L.M. Laita, E. Roanes-Macías, An accelerated-time simulation of departing passengers' flow in airport terminals, Math. Comput. Simul. 67 (2004) 163–172.
[14] A. Hernando, E. Roanes-Lozano, A. García-Álvarez, An accelerated-time microscopic simulation of a dedicated freight double-track railway line, Math. Comput. Modelling 51 (2010) 1160–1169.
[15] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271.
[16] G. Aguilera, J.L. Galán, M.A. Galán, Y. Padilla, P. Rodríguez, Generating random samples from continues and discrete distributions with derive, DERIVE Newsl. 75 (2009) 22–43.