A Continuous Cellular Automata Approach to Highway Traffic Modeling

Emanuele Rodaro[1] and Öznur Yeldan[2]

[1] Departamento de Matemática, Faculdade de Ciências
Universidade do Porto, 4169-007 Porto, Portugal
emanuele.rodaro@fc.up.pt
[2] Dipartimento di Matematica, Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
oznur.yeldan@mail.polimi.it

1 Introduction

Traffic models are fundamental resources in the management of road network. A real progress in the study of traffic has obtained with the introduction of the models based on cellular automata (CA). CA models (CAMs) have the ability of being easily implemented for parallel computing because of their intrinsic synchronous behavior. They are conceptually simple, since a set of simple rules can be used to simulate a complex behavior. The traffic models based on CA are capable of capturing micro-level dynamics and relating these to macro-level traffic flow behavior. However, they are lack of the accuracy of other microscopic traffic models like the time-continuous car-following [1] ones where the behavior of a driver depends only on the leading vehicle. A basic one-dimensional CAM for highway traffic flow was first introduced by Wolfram, where he gave an extensive classification of CAMs as mathematical models for self-organizing dynamic systems [9, 10]. In 1992, Nagel and Schreckenberg proposed the first nontrivial traffic model (the NaSch model) based on CA for single-lane highway [5]. This paper gave rise to many other CAMs for traffic flow [3, 4, 6–8] whose common feature is that cells represent a piece of the road ("NaSch-type" models).

In this paper, we abandon the idea of dividing the road into cells and we introduce a new traffic model for highways using continuous cellular automata (CCA) to introduce the continuity in space. We consider a hybrid between the usual microscopic models (in general defined by means of a system of differential equations) which are very accurate in predicting general traffic behavior but computationally expensive, and the usual CAMs which are very efficient due to their simplicity and intrinsic parallelism making them natural to be implemented for parallel computing. This process of passing from the typical coarse-granularity of CAMs to the continuity in space is done assuming that cells represent vehicles. In this way, we obtain the immediate advantage of having less cells to compute. The continuity also gives us the possibility to refine the microscopic rules that govern the traffic dynamics using fuzzy reasoning to mimic different real-world driver behaviors. All parameters of the decision process of the drivers are modeled individually by means of fuzzy subsets, thus various types of drivers can be taken into consideration. This gives us the possibility to study how the heterogeneity influences the traffic macroscopically. The CCA model proposed in this paper is defined first for a single-lane road and then we extend the model to the multi-lane case where the extension is not as natural as "NaSch-type" models.

2 Overview of the Model

The single lane model is a CCA $\mathcal{SL} = (\mathbb{Z}, \Sigma, \mathcal{N}, \delta)$ where the lattice is the set of integers and the set of cell states $\Sigma = (K \times \mathbb{R}_0^+ \times \mathbb{R}_0^+ \times \mathbb{R} \times \{L, 0, R\} \times \{L, 0, R\}) \cup \{\perp\}$. A cell with the empty state $\perp$ represents a cell without a vehicle. The generic i-th non-empty cell is in the state $\sigma_i(t) = (k_i, x_i(t), v_i(t), s_i(t), d_i(t), d'_i(t))$ where

- k_i represents the kind of vehicles (seen as a unique entity driver/vehicle). It contains all the parameters such as: *the maximum velocity* (v_{max}), *the optimal velocity* (v_{opt}), *the length* (l_i), *the fuzzy membership functions*, *the maximum stress* (s_{max}), *the minimum stress* (s_{min}), *the probability functions of lane-changing to the right lane* ($P_R(x)$) and *to the left lane* ($P_L(x)$) (used in the multi-lane model).
- $x_i(t)$ is the position, $v_i(t)$ is the velocity, and $s_i(t)$ is the stress, a variable to keep track of how much the driver is above or below of his optimal velocity. In the single-lane model, $s_i(t)$ is introduced to implement a more realistic driver behavior since drivers usually tend to decelerate when they are moving with a velocity higher than their optimal velocity. However, this parameter is mainly used in the lane-changing process of the multi-lane model.
- $d_i(t)$ is the variable describing the desire for: lane-changing to the left "L", to the right "R" and staying on his own-lane "0", and $d'_i(t)$ is the variable showing from which lane the i-th vehicle is transferred: from the left lane "L", from the right lane "R" and not transferred "0"(these variables are used just in the multi-lane model).

$\mathcal{N}$ is a kind of one-dimensional extended Moore neighborhood defined by $\mathcal{N}(i) = (i-1, i, i+1, i+2)$, and $\delta : \Sigma^4 \rightarrow \Sigma$ is the local transition function defined componentwise. The space is updated by $x_i(t+1) = x_i(t) + v_i(t+1)$ (the unit of time is fixed to 1 sec.) and the velocity is updated by $v_i(t+1) = \min(v_{max}, \Delta x_i^+(t), \max(0, v_i(t) + A_i(t)))$ where $A_i(t)$ is the acceleration calculated using two sets of fuzzy IF-THEN rules (see Fig. 1) which take into consideration the distances and collision times of the back, front and next front vehicles, and the velocity. Besides the fuzzy rules to calculate $A_i(t)$, it is worth noting that in the case $A_i(t)$ is the stochastic function defined by $A_i(t) = 7,5\ m/s^2$ with probability p and $A_i(t) = 0$ otherwise, we essentially obtain the Nagel and Schreckenberg's first stochastic model [5] with the only difference that the space here is continuous. However, we have chosen to implement the decision of the acceleration using two fuzzy modules to mime driver behaviors more realistically.

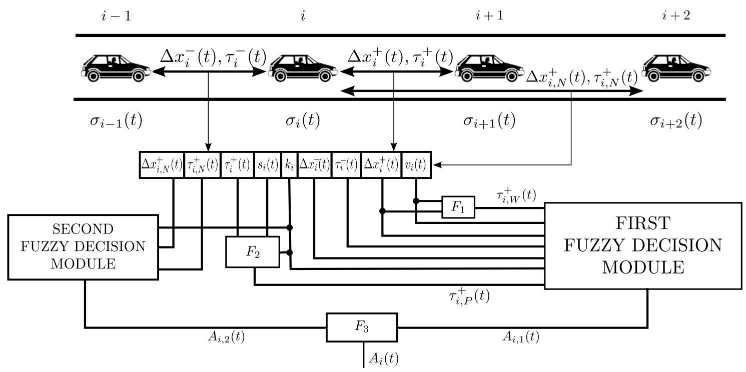

Fig. 1. A block diagram of the local transition function δ

The extension to the multi-lane case is not trivial as it is in the NaSch-type models where adding a lane simply means adding an array of cells and where the local transition function can naturally be extended. This is a consequence of having a clear physical interpretation of the model given by the fact that space is represented by cells. The most natural candidate is a union of interacting single-lane CCA where the interaction is given by a transfer operation. The process of transferring a vehicle from one lane to another depends on the desire of the vehicle to change lane (calculated using a stochastic process depending on the stress parameter) and the physical possibility of a vehicle to get transferred to a lane (depending on some safety constraints). Suppose that we have M-lanes represented by M-copies of the single-lane CCA $\mathcal{SL}$ in the configurations $c_1, \ldots, c_M$. We scan each lane starting from the left-most lane (in the configuration c_1) and we transfer the vehicles to the adjacent lanes. After this process, for each lane we apply the single-lane CCA model to update the configuration and this update is done by means of the global transition function of $\mathcal{SL}$. In this way, we obtain a new array of configurations $c'_1, \ldots, c'_M$, and this process represents 1 *sec.* of the simulation. Although this model is presented as an array of communicating CCA, we have proved that it is possible to define a CCA which actually simulates this model.

3 Conclusion

For a first test, we implement the model using Python with an object-oriented philosophy of programming. Using a questionnaire we set up two kinds of vehicles (long vehicles and passenger vehicles) which we have used to run a series of experiments. Analyzing the experimental results, we study the influence of different composition of vehicles on the macroscopic behavior of the traffic in order to observe the typical traffic flow phenomena (see Fig. 2). The code written in Python does not take advantage of the CA and its typical synchronous behavior.

For this reason, we also adapt the code using PyCuda to partially parallelize the multi-lane model on GPU's and we see that it is possible to boost the speed of execution by a factor of ~ 10, for instance, 1000 steps of the simulator with 5000 vehicles are run in 194 *sec*. instead of 1608 *sec*. (on a laptop equipped with a processor $i7$ intel and with a graphic card NVIDIA GeForce GT 555M).

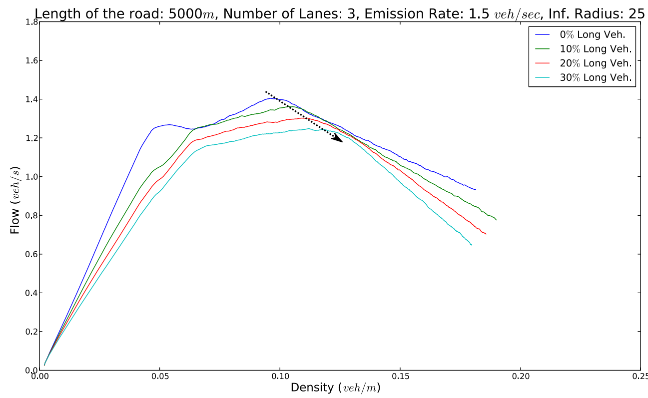

Fig. 2. The effect of heterogeneity on the fundamental diagram

References

1. Brackstone, M., McDonald, M.: Car-following: A historical review. Transportation Research Part F 2, 181–196 (1999)
2. Kari, J.: Theory of cellular automata: A survey. Theoretical Computer Science 334, 3–33 (2005)
3. Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M.: Disorder effects in cellular automata for two-lane traffic. Physica A: Statistical and Theoretical Physics 265(3–4), 614–633 (1999)
4. Maerivoet, S., De Moor, B.: Cellular automata models of road traffic. Physics Reports 419(1), 1–64 (2005)
5. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. Journal de Physique I 2(12), 2221–2229 (1992)
6. Nagel, K., Wolf, D.E., Wagner, P., Simon, P.: Two-lane traffic rules for cellular automata: A systematic approach. Phys. Rev. E 58(2), 1425–1437 (1998)
7. Rickert, M., Nagel, K., Schreckenberg, M., Latour, A.: Two lane traffic simulations using cellular automata. Physica A: Statistical and Theoretical Physics 231(4), 534–550 (1996)
8. Wagner, P., Nagel, K., Wolf, D.E.: Realistic multi-lane traffic rules for cellular automata. Physica A: Statistical and Theoretical Physics 234(3–4), 687–698 (1997)
9. Wolfram, S.: Theory and applications of cellular automata. World Scientific Press, Singapore (1986)
10. Wolfram, S.: A new kind of science, Wolfram Media (2002)