DIGITAL LABORATORY OF AGENT-BASED HIGHWAY TRAFFIC MODEL*

Anna T. Lawniczak

Department of Mathematics and Statistics University of Guelph
Guelph, Ontario N1G 2W1, Canada

Bruno N. Di Stefano

Nuptek Systems Ltd., Toronto, Ontario M5R 3M6, Canada

(Received March 1, 2010)

We describe our microscopic model of highway traffic and its implementation as digital laboratory. We discuss the range of experiments that can be conducted using this laboratory. Software implementation details are discussed together with the model because the implementation affects the model and delimits what can be modeled. This is seldom described in the literature, but lack of this knowledge often affects the ability of the reader to replicate the research. We model the expressway as a number of adjacent lanes, where each lane is divided into cells. Each cell is assumed to be 7.5 m. The most innovative aspect of our model is that we model multiple lanes as a single 1-D automaton. By extending the "Cellular Automata" (CA) paradigm to the "Global Cellular Automata" (GCA) paradigm we can represent the multilane highway with a single 1-D GCA, which can be implemented to execute faster than a traditional 2-D CA implementation and than a multi 1D CA implementation. We present selected simulation results and outline our plan for future work.

PACS numbers: 89.40.Bb, 89.20.Kk, 89.20Bb

1. Introduction

We developed a microscopic agent-based model of highway traffic and implemented it as a digital laboratory. In this paper, we describe our model, the resulting digital laboratory, and the range of experiments that can be conducted. We present selected simulation results.

This model is meant for practical traffic engineering applications, to estimate travel time between two access ramps, an entry ramp and an exit ramp, once certain highway traffic parameters are known at certain points

* Presented at the Summer Solstice 2009 International Conference on Discrete Models of Complex Systems, Gdańsk, Poland, June 22–24, 2009.

of the highway. Our concern is primarily with effects of flow and congestion through a long highway on travel time. The difference between our work and published research previously conducted by others is that we model much longer highways, *e.g.* at least 500 km, and a much higher number of vehicles, *e.g.* realistic traffic conditions over several days.

Traditional macroscopic models of the 50s, such as the LWR model (Lighthill, Whitham, Richards), [1] and [2], are characterized by a large number of parameters without an immediately intuitive equivalent when conducting empirical investigations. Microscopic models based on cellular automata (CA) such as the one of Cremer and Ludwig, [3], and the Nagel Schreckenberg model, [4], have solved this problem. The Nagel Schreckenberg model is the starting point of a very fruitful stream of research, see for instance [5–8], and [9]. However, most of this work has been conducted from the point of view of physics and statistical physics, to investigate dynamical aspects of highway traffic, see for instance [10] and [11]. Most such models do not have features that are very important from an engineering point of view, such as the ability to track individual vehicles during their trip, from entry ramp to exit ramp.

Elementary Cellular Automata (ECA) Rule 184 is often called "*the traffic rule*" and has been used successfully for many road traffic models, *e.g.* see [12, 13], and [14]. However, also models based on ECA 184 lack the ability to track individual vehicles during their trip, from entry ramp to exit ramp. Agent-based modeling provides a solution to this problem.

When speaking about agent-based modeling, we define the "agent as an autonomous entity capable of autonomously interacting with its environment and other agents". According to this definition, each car is an agent and the road is the environment. Thus, if properly designed, each car is capable of moving on the road according to the rules of traffic and the laws of physics, avoiding accidents with other cars (*i.e.*, other agents) and with obstacles on the road, *e.g.* bridge pillars, trees, *etc.* (*i.e.*, the environment). This type of model can be implemented in many ways, including models based on heuristics. As new information emerges while testing and validating the model, a large number of "*If ... Then*" and "*If ... Then ... Else*" statements sneak stealthily into the code with many function calls. Consequently, the code becomes hard to verify for correctness and even harder to modify. Because of this, care must be taken to make sure that all situations to be accounted for can be expressed analytically as much as possible and that proper formalism of motion and navigation rules and of data structures is maintained. To avoid this situation we have developed an agent-based model that has been evolved directly from ECA Rule 184, by extending ECA Rule 184 to model realistic highway traffic, *i.e.* traffic with entry and exit ramps, multiple lanes, variable acceleration and speed, obstacle avoidance,

driver's prior knowledge and anticipation (*e.g.*, response to brake lights). The complexity of the process being modeled does not allow writing a single analytical expression capable of describing motion and navigation, but it is possible writing a set of analytical expression, each describing motion and navigation at different instants in time during the simulation.

Software implementation details are discussed together with the model because the implementation affects the model and delimits what can be modeled. This is seldom described in the literature, but lack of this knowledge often affects the ability of the reader to replicate the research.

2. Our model

We model the expressway as a number of adjacent lanes, where each lane is divided into cells. Each cell is assumed to be 7.5 m in length as in most of the literature, *e.g.* [4] and [15]. This has been chosen because it corresponds to the space occupied by the typical car plus the distance to the preceding car in a situation of dense traffic jam. The traffic jam density is given by 1000/7.5 m approximately equal to 133 vehicles per km. As a unit of time, we have the "time step" duration of 3 seconds. The minimum speed of a vehicle advancing by one cell at each time step is equivalent to 9 km/h (that is, $7.5 \times 3600/3 = 7.5 \times 1200 = 9000$ m/h). This allows representing most realistic and legal speeds observed in Canadian highways, with a vehicle advancing by a maximum of 11 cells per time step, that is, 99 km/h, as the speed limit is at 100 km/h.

Probably, the most innovative aspect of our model is that we model multiple lanes as a single 1-D CA.

Fig. 1 shows a small section of a 3 lane highway. Inside each cell we show two digits separated by a comma. This first digit represents the lane number while the second digit represents the cell number. Fig. 2 shows a trailing

2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9

Fig. 1. Small section (*i.e.*, 75 m) of 3 lane highway.

vehicle, at time $t = t_0$, in cell $\langle 0, 3\rangle$ about to pass a leading vehicle in cell $\langle 0, 4\rangle$. Fig. 3 shows the trailing vehicle, at time $t = t_1$, in cell $\langle 1, 3\rangle$ after having changed lane. This can be implemented with a 2-D CA just moving to an adjacent cell, see for instance [16]. Alternatively, one could implement each lane with a 1-D CA, hopping from one CA to another when changing lane. The solution of the 2-D CA requires two space loops, one for rows

and columns, with double of the number of implied end of loop tests. The multiple 1-D CA is cumbersome to parameterize if one wants to implement a model with a variable number of lanes.

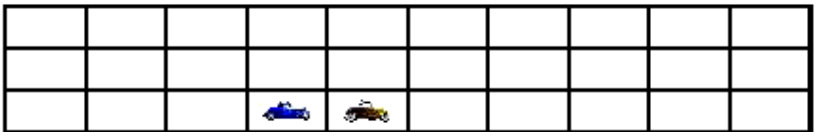

Fig. 2. Small section of 3 lane highway with trailing vehicle about to change lane (time $t = t_0$).

Fig. 3. Small section of 3 lane highway with trailing vehicle that has just changed lane (time $t = t_1$).

By extending the CA paradigm to the "Global Cellular Automata" (GCA) paradigm we can represent the multilane highway with a single 1-D CA, indeed a single 1-D GCA, see [17, 18], and [19]. In a GCA each cell is a neighbour of all other cells, its "global" neighbours. Any entity moving in a GCA can jump from any cell to any other cell, as long as the rules of motion within the GCA are applied in the same way for all cells. Fig. 4 shows the same small section of highway as Fig. 1 mapped into a 1-D GCA. Each cell

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9

Fig. 4. Small section of 3 lane highway represented as a GCA.

shows one digit indicating the lane number on one line, then a comma on the next line as a separator, and the cell number (within the lane) on the last line. Lane 0 occupies the first 10 cells from the left. Lane 1 occupies the next 10 cells and lane 2 occupies the last 10 cells. Fig. 5 shows the same scenario as Fig. 2, *i.e.* a trailing vehicle, at time $t = t_0$, in cell $\langle 0, 3\rangle$ about to pass a leading vehicle in cell $\langle 0, 4\rangle$. Fig. 6 shows the same scenario as Fig. 3, *i.e.* the trailing vehicle, at time $t = t_1$, in cell $\langle 1, 3\rangle$ after having changed lane. While Fig. 6 can be logically mapped to Fig. 3, as it is done in the GUI of the software implementing our model, in reality, the physical location is different.

Fig. 5. Small section of 3 lane highway represented as a GCA with trailing vehicle about to change lane (time $t = t_0$).

Fig. 6. Small section of 3 lane highway represented as a GCA with trailing vehicle that has just changed lane (time $t = t_1$).

We started the description of our model emphasizing the detail of how we implement multiple lanes, because this detail affects the implementation of the entire model. We model multilane highway traffic with two loops, an "external" time loop and an "internal" space loop. To be able to model different traffic behaviour the internal space loop is replaced by various consecutive space loops, each one of them required to perform a single operation. After all space loops have been completed, if the program operates in graphic mode, the display is updated and the processing moves to the next time step. However, using a 1-D GCA reduces the need from two loops, *i.e.* row loop and column loop, to one loop only, thus reducing the machine cycles by the end of loop checking condition that is actually performed at each time step before incrementing the loop counter. This comes with no substantial penalty, because moving an object to a distant cell is only a bit more costly, in machine cycles terms, than moving it to an adjacent cell.

At the beginning of each time step vehicles are generated at each entry ramp according to a predefined vehicle generation probability that can be specified individually for each entry ramp. After all vehicles have been generated for each entry ramp, they are queued and placed on the ramp data structure (a first-in-first-out queue). Each space loop scans the 1-D GCA from the highest numbered cell to the lowest numbered cell, *i.e.* cell 0. If a cell is empty, no action takes place. If a cell is occupied a loop specific algorithm is applied. The following loops are executed in sequence. If the current cell is occupied by a car,

- that car changes lane to the right if possible (we assume that we are modeling traffic of a country where cars drive on the right side of the road and where cars are on the rightmost lane unless they have to pass a slower vehicle or avoid an obstacle),
- that car changes lane to the left if needed to pass a slower vehicle or avoid an obstacle,
- we apply a modified Nagel Schreckenber algorithm, [4], to that car,

- we check if that car is expected to exhibit "erratic behaviour" (*i.e.*, sudden acceleration and/or deceleration) and, if applicable, we move probabilistically the car within a range of ± 3 cells, *i.e.* 1, 2, or 3 cells ahead or 1, 2, or 3 cells back, as it may apply.

If the program is executing in graphic mode, we update our display. At this point all space loops have been executed.

We compare the predefined destination (exit ramp) of each vehicle with a neighbourhood of the cell where the vehicle is currently located. If the vehicle is "next to the exit ramp", we remove the vehicle from the expressway and update all data structures. Exit ramps are listed in the input configuration file without any other parameter than a keyword and the number corresponding to the cell where the exit ramp is located. Exit ramps are always on lane number 0 except the "end of highway" exit ramps, which coincide with the highest numbered cell for each lane. Time is incremented.

2.1. Software implementation

We implemented our model using the C++ programming language. The execution can be in graphic mode and in non graphic mode. For graphic mode we used GLUT, the OpenGL Utility Toolkit, [20]. We tested the graphic mode operation under MS Windows XP and MS Windows Vista. We tested the non graphic mode operation under MS Windows XP, MS Windows Vista, and Linux using SHARCNET, see [21]. Almost all storage data structures used in the code are vectors as defined in the C++ Standard Template Library. All code written by us is compatible with *ISO/IEC 14882:1998*. We used a pseudorandom number generator previously developed by us in the course of a different research, [23] and [24]. We implemented the pseudorandom number generator as a class because the sequence of each instance used in the program must be independent of the others. "We are not content with one sequence of random numbers in the simulation system because we use them for different purposes", see [24]. We called this implementation of our model Freeway.exe.

FreewayCA.exe can run in "non graphic mode", for fast execution, and in "graphic mode", to provide an intuitive understanding on how the simulated traffic evolves.

FreewayCA.exe simulates traffic faster than in real time, *i.e.* the execution time of a simulation of a given configuration will come to conclusion faster than the events being simulated. The actual execution time will depend on:

- On the computer used for the simulation.
- The length of the expressway and the number of lanes being simulated.

- The number of entry and exit ramps being simulated.
- The traffic density being simulated.
- A number of other case specific parameters.

However, the simulation time will always be much faster that the real time of the event being simulated. Typically, to simulate 24 hours of traffic for an expressway long 500 km will require minutes on a UNIX/Linux grid computer and about 2–3 hours on a modern notebook computer, depending on notebook CPU speed and available RAM.

FreewayCA.exe allows for multiple degrees of freedom, *i.e.* all simulation parameters can be set independently of each other. Simulations by means of FreewayCA.exe can be configured to run user defined experiments. Currently FreewayCA.exe can simulate expressways with user-defined:

- Length (actually tested for more than 1,000 km).
- Number of lanes (actually tested with up to 5 lanes).
- Max simulation time (actually tested for 7 days of traffic).
- Global (expressway wide) maximum speed (in all cases when no other speed limit is in effect).
- Global car creation probability for all entry ramps for which no explicit car creation probability is defined.
- Global last exit ramp destination probability for all vehicles for which no explicit definition of destination probability is defined.
- Complex probability distribution of drivers exhibiting erratic behaviour violating traffic rules.
- Any probability distribution of probe vehicles.
- Any number of entry ramps.
- A time varying independent traffic generation probability distribution at each entry ramp.
- Any number of exit ramps.
- Any number of obstacles, deterministically present during user defined time intervals.

Particular flexibility is associated to the entry ramps. For each entry ramp it is possible to define, valid for a specific time interval:

- An independent vehicle generation probability distribution (*e.g.*, from midnight to 7:00 am a 0.01 generation probability, from 7:01 to 9:30 am a 0.75 generation probability, from 9:31 am to 3:30 pm a 0.25 generation probability, from 3:31 to 6:30 pm a 0.75 generation probability, from 6:31 pm to midnight a 0.10 generation probability).

- An independent probability distribution of having the last exit ramp as a destination.

- A maximum velocity and an associated probability that a given vehicle will reach that maximum velocity.

- An independent probability that a given vehicle is a probe vehicle.

Because of the flexibility, of the degrees of freedom, associated to this program, it is impossible to calculate a priori the execution time when simulating a specific scenario. However, approximate predictions are possible. A considerable number of experiments are required. The work of establishing sufficient heuristics for execution time prediction is currently under way. At this stage we can only say that FreewayCA.exe simulates traffic significantly faster than it takes to travel in real time.

2.2. Software invocation

The implementation of our model, the software package FreewayCA.exe, is invoked, at command line prompt, with the following command:

```
FreewayCA.exe <input data file> <output data file> <output data file2>
```

where:

- <input data file> is the configuration file describing the expressway and the experiment to be conducted,

- <output data file> is the output file tracking every car at every time step,

- <output data file2> is the output file reporting aggregate data about the number of cars on the road and the number of cars backed-up on entry ramps at every time step.

While the input data file is normally very small, typically few kilobytes, the output data file, which is a function of the size of the expressway and of the duration of the simulation, may be extremely large, even several gigabytes in our experiments. The size of the second type of output file varies with the experiment being conducted. Some experiments do not create any backup at the entry ramps and the file is nearly empty, while some other

experiments generate considerable back up (*e.g.*, when an accident with long lasting road obstruction is modeled) and the resulting file can be a couple of hundred kilobytes.

File names are not fixed and can be arbitrarily chosen by the user.

2.3. Configuration file

A real expressway can be modeled by either manually generating a configuration file by using an ASCII text editor (*e.g.*, notepad or vi) or by converting a different description (*e.g.*, a digitized map) into the configuration file by means of a suitable software utility. Fig. 7 shows the list of the configuration commands currently defined and tested.

```
DISPLAY_MODE 0  // non graphic mode for fast execution
DISPLAY_MODE 1  // graphic mode for all cars

CELL_NUM <NUMBER_OF_CELLS_PER_LANE>
LANE_NUM <NUMBER_OF_LANES>
MAX_TIME <MAXTIME>
MAX_SPEED <CELLS_PER_TIME_STEP>
ENTRY_CAR_PROB <CREATION_PROB>
LAST_DEST_PROB <LASTEXIT_PROB>

ENTRY <LANE> <CELL> <STARTTIME> <ENDTIME>
<CREATIONPROB> <LASTCELLPROB> <MAXSPEED>
<MAXSPEEDPROB> <PROBEPROB>

EXIT <CELL>

OBSTACLE <LANE><CELL><START_TIME><END_TIME>

ERRATIC_PROB <ERRATICPROB> <1AHEADPROB> <2AHEADPROB>
<3AHEADPROB> <1BACKPROB> <2BACKPROB> <3BACKPRB>
<1LEFT> <1RIGHT>
```

Fig. 7. List of configuration commands currently defined and tested.

Most commands are self explanatory. All commands accepting the pair <START_TIME> and <END_TIME> as parameters can be repeated multiple times in the configuration file. For instance, the configuration file could contain the fragment shown in Fig. 8.

```
.....
OBSTACLE 0 760  1200  2400
OBSTACLE 0 760  5000  7000
OBSTACLE 0 760  9000 10000
.....
```

Fig. 8. Sample fragment of code containing various "OBSTACLE" statements.

This command means that we want to model a temporary obstacle in lane 0, cell 760, and that this obstacle will be present for three different time intervals, *i.e.* from time step 1200 to time step 2400, from time step 5000 to time step 7000, and from time step 9000 to time step 10000. At all other times, there is no obstacle at lane 0, cell 760, and traffic can flow normally. This could, for example, be the way of modeling the effect of temporary maintenance work.

Fig. 9 shows a fragment of code that could be used to change traffic creation probability at an entry ramp (*i.e.*, lane 0, cell 600) at different times of the day. For instance between time step 0 and time step 7200 the creation probability is 0.10, while between time step 7201 and time step 10800 the creation probability is 0.80, *etc.* In this way, rush hours and low traffic periods can be modeled.

```
.....................................................................
ENTRY 0  600      0  7200 0.10 0.5  11 0.9 0.0
ENTRY 0  600   7201 10800 0.80 0.5   8 0.8 0.0
ENTRY 0  600  10801 18000 0.50 0.5  10 0.8 0.0
ENTRY 0  600  18000 21600 0.80 0.5  10 0.8 0.0
.....................................................................
```

Fig. 9. Sample fragment of code containing various "ENTRY" statements.

The command ERRATIC_PROB allows to specify a probability that a given vehicle exhibit an erratic behaviour (*e.g.*, abrupt accelerate and/or decelerate) and a different probability of moving abruptly 1, 2, or 3 cells ahead or 1, 2, or 3 cells back, or one lane to the right or one lane to the left as it may apply.

The final destination probability and the maximum speed probability define not only the obvious probabilities implied by their names, but also the values of the complementary probabilities. In other words if Pd is the probability that the vehicle is instantiated with last cell as its destination, (1-Pd) is the probability that the vehicle will go elsewhere, to other exit ramps. The specific exit ramp is assigned randomly. Similarly, if Pvmax is the probability that the vehicle will be instantiated with the specified maximum speed, (1-Pvmax) is the probability that the vehicle will be instantiated with a different speed. The specific different speed will be assigned randomly.

To facilitate the creation of the configuration file we developed a configuration builder that can be run under MS Windows XP and MS Windows Vista, Fig. 10. The user simply enters locations in terms of cell number and of lane number and time as time step number. Once the configuration has been entered, by pressing the "save" button, the configura-

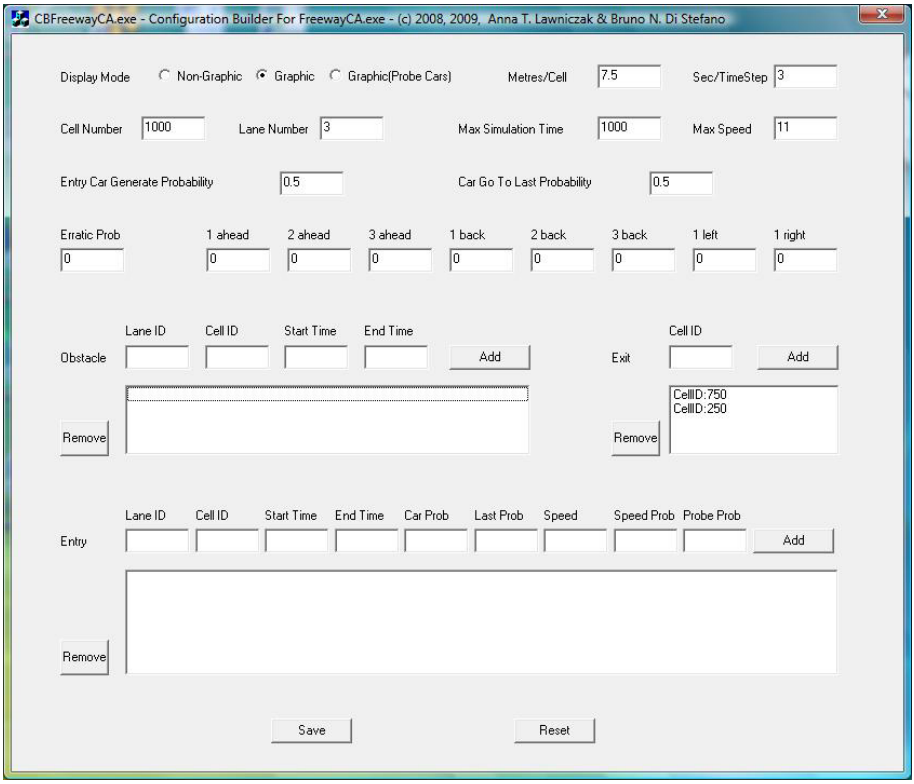

Fig. 10. Dialog box for configuration input.

tion file is created with a default name reflecting the configuration being modeled. As an example, let us assume that the file generated is D1L3C16000T7200V11PC0.80PL0.50.txt. In this case, the configuration file is meant for a simulation:

- in graphic mode (*i.e.*, D1),
- with 3 lanes (*i.e.*, L3),
- with 16000 cells (*i.e.*, C16000),
- to be run for 7200 time steps (*i.e.*, T7200),
- with maximum velocity for all vehicles, unless individually specified, 11 cells per time step (*i.e.*, V11),
- with probability of vehicle creation, in all cases when other creation probability is not specified, equal to 0.80 (*i.e.* PC0.80),
- with probability of vehicle heading to the last cell, *i.e.*, the end of the highway, in all cases when other last cell destination probability is not specified, equal to 0.50 (*i.e.*, PL0.50).

Each configuration file name represents a class of possible configurations and does not provide any information about number and location of entry ramps, exit ramps, obstacles, and potential erratic behaviour being modeled. A file name containing these details would be too long.

2.4. Output data files

The first output data file consists of as many records as the number of cars that have transited in the highway during the experiment. Each record consists of two lines, which can be processed to extract both global aggregate information about the experiment and vehicle specific information.

The first line of each record contains: Car ID, Creation Time, Start Time, Exit Time, Maximum Speed, "Probe or Not", Destination. The Car ID is a unique identifier of the car, a bit like the plate number or the serial number of the car chassis. "Creation Time" is the time, measured as absolute time step, when the vehicle has been instantiated. An instantiated vehicle exits and is placed on a waiting queue at an entry ramp. "Start Time" is the time, measured as absolute time step, when a vehicle actually leaves the entry ramp and enters the expressway. "Exit Time" is the time, measured as absolute time step, when the vehicle actually leaves the expressway via its destination exit ramp. "Maximum Speed" is the maximum speed allowed for the specific vehicle. This allows conducting experiments about vehicle violating the speed limit, as done by Makowiec in [22]. "Probe or Not" is a variable used to conduct experiments with so called "Probe Vehicles", vehicles that traffic engineers probe, *i.e.* check wirelessly during the trip, to verify their location. "Destination" is the cell number identifying the exit ramp.

The second line of each record contains a sequence of fields called "Lane Number" and "Cell Number". Each field represents the exact position of the vehicle at a given time step. The number of these fields (*i.e.*, Lane Number and Cell Number pairs) is equal to "Exit Time" minus "Start Time". By comparing the last Lane Number and Cell Number "Destination" from the first line of the record we know if the vehicle is still on the road, has arrived to destination, or has missed the exit ramp.

Aggregate information is output to the second output data file where at each time step we store: current time step number, total number of vehicles instantiated, total number of vehicles on the road, and total number of vehicles delayed in entry ramps. Thus, it is possible to infer how many vehicles have exited at this time.

3. Example of experiment: modeling effects of erratic vs. normal driver's behaviour on travel time

As an example of the use of Freeway.exe, we run it with the following parameters:

- `CELL_NUM = 16000 (corresponding to 120 km),`
- `LANE_NUM = 3,`
- `MAX_TIME = 2400,`
- `MAX_SPEED = 11 (99 km/h),`
- `ENTRY_CAR_PROB = 0.5,`
- `LAST_DEST_PROB = 1.`

To simplify our experiment, we do not consider any intermediate entry or exit ramp. We simulate the following case:

- No erratic drivers,
- probability 0.05 of erratic drivers,
- probability 0.10 of erratic drivers,
- probability 0.20 of erratic drivers,
- probability 0.30 of erratic drivers.

We make the hypothesis that the drivers are erratic in an aggressive way. Regardless what is the probability of behaving in an erratic way we set the other parameters as follows: probability 0.5 of jumping one cell ahead, probability 0.5 of jumping two cells ahead, probability 0.5 of jumping three cells ahead. All probabilities of slowing down abruptly are set to 0.0. The probability of changing lane to the right is 0.0, while the probability of changing lane abruptly to the left is 0.5.

Table I shows the effect of erratic *vs.* normal driver's behaviour on travel time. The scenario that we have investigated is very unusual, because it is a 3 lane stretch with no entry or exit ramp before the end of the expressway. It is almost akin to a racetrack. One would expect that aggressive drivers should do well in this scenario. However, their presence slows the average driver down, at least when their number is noticeable (*i.e.*, creation probability is 0.20).

We would like to emphasize that this experiment has been presented only as an example of the capability of the model and of the resulting digital laboratory. This is not a complete study of the effect of erratic *vs.* normal driver's behaviour over travel time. A much larger number of experiments are required to reach conclusions, accounting for different expressway topologies (*i.e.*, number of entry and exit ramps), different traffic densities (*i.e.*,

TABLE I

Effect of erratic *vs.* normal driver's behaviour on travel time.

Erratic behaviour probability	Total distance traveled by each car	Total number of cars	Average travel time (time steps)	Average velocity (cells/time step)
0.00	15999	661	1625	9.85
0.05	15999	753	1647	9.71
0.10	15999	710	1618	9.89
0.20	15999	729	1639	9.76
0.30	15999	736	1642	9.74

vehicle creation probability at each ramp) and different type of erratic behaviour. We are actually conducting these experiments and plan to discuss them elsewhere.

4. Future work

We are currently validating our model with information from traffic engineers. We plan on using this model for practical traffic engineering applications, to estimate how some technological innovations affects travel time between two access ramps, an entry ramp and an exit ramp, once certain highway traffic parameters are known at certain points of the highway. Our concern is primarily with effects of flow and congestion through a long highway on travel time. The technological innovations include wireless communication from roadside transmitters to vehicles, wireless communication among vehicles, *etc.* We are considering parallelizing our code for execution under SHARCNET, a consortium of Canadian academic institutions sharing a network of high performance computers, see [21].

A.T. Lawniczak acknowledges partial financial support from the Natural Science and Engineering Research Council (NSERC) of Canada. B.N. Di Stefano acknowledges full financial support from Nuptek Systems Ltd. A.T. Lawniczak acknowledges support from SHARCNET in the form of computing facilities and resources for the Linux implementation and testing of the software. The authors thank The Fields Institute for Research in Mathematical Sciences for providing hospitality and Prof. Danuta Makowiec and Prof. Rolf Hoffmann for providing inspiring conversation.

REFERENCES

[1] M.J. Lighthill, G.B. Whitham, *Proc. R. Soc.* **A229**, 317 (1955).

[2] P.I. Richards, *Operations Research* **4**, 42 (1956).

[3] M. Cremer, J. Ludwig, *Mathematical and Computers in Simulation* **28**, 297 (1986).

[4] K. Nagel, M. Schreckenberg, *J. Phys.* **I2**, 2221 (1992).

[5] W. Knospe, L. Santen, A. Schadschneider, M. Schreckenberg, *Phys. Rev.* **E70**, 016115 (2004).

[6] W. Knospe, L. Santen, A. Schadschneider, M. Schreckenberg, *J. Phys.* **A33**, L477 (2000).

[7] W. Knospe, L. Santen, A. Schadschneider, M. Schreckenberg, *Phys. Rev.* **E65**, 056133 (2002).

[8] M. Schreckenberg, A. Schadschneider, K. Nagel, N. Ito, *Phys. Rev.* **E51**, 2939 (1995).

[9] P. Wagner, K. Nagel, D. Wolf, *Physica A* **234**, 687 (1997).

[10] D. Chowdhury, L. Santen, A. Schadschneider, *Phys. Rep.* **329**, 199 (2000).

[11] D. Helbing, *Rev. Mod. Phys* **73**, 1067 (2001).

[12] H. Fukś, *Phys. Rev.* **E55(3)**, R2081 (1997).

[13] B. Chopard, M. Droz, *Cellular Automata Modelling of Physical Systems*, Cambridge University Press, Cambridge 1998.

[14] N. Boccara, *Modeling Complex Systems*, Springer-Verlag, New York 2004.

[15] S. Maerivoet, B. De Moor, *Phys. Rep.* **419**(1), 1 (2005).

[16] A.T. Lawniczak, B.N. Di Stefano, *Automata 2008: Theory and Applications of Cellular Automata*, Eds. A. Adamatzky, R. Alonso-Sanz, A. Lawniczak, G. Martinez, K. Morita, T. Worsch, Luniver Press, 2008, p. 527.

[17] R. Hoffmann, K.-P. Volkmann, S. Waldschmidt, Theoretical and Practical Issues on Cellular Automata, Proceedings of the Fourth International Conference on Cellular Automata for Research and Industry (ACRI 2000), Eds. S. Bandini, T. Worsch, Karlsruhe 4–6 October, 2000, Springer-Verlag, London 2000.

[18] R. Hoffmann, K.-P. Volkmann, S. Waldschmidt, W. Heenes, *Lect. Notes Comput. Sci.* **2127**, 66 (2001).

[19] R. Hoffmann, K.-P. Volkmann, W. Heenes, Proceedings of International Parallel and Distributed Processing Symposium (IPDPS 2003), Nice, France, April 22–26, 2003, IEEE Comp. Soc., 2003.

[20] `http://www.opengl.org/resources/libraries/glut/`

[21] `https://www.sharcnet.ca/`

[22] D. Makowiec, W. Miklaszewski, *Lect. Notes Comput. Sci.* **3993**, 256 (2006).

[23] A.T. Lawniczak, A. Gerisch, B. Di Stefano, Proceedings of IEEE CCECE'2003, Montreal, Quebec, Canada, May 4–7, 2003.

[24] A.T. Lawniczak, A. Gerisch, K.P. Maxie, B. Di Stefano, IEEE Proceedings of "HPCS 2005: The New HPC Culture The 19th International Symposium on High Performance Computing Systems and Applications", Guelph, May 15–18, 2005, p. 9.