Traffic Simulation Framework

Paweł Gora
Faculty of Mathematics, Computer Science and Mechanics
University of Warsaw
Warsaw, Poland
pawelg@mimuw.edu.pl

***Abstract*—In the paper, a software Traffic Simulation Framework is presented. This is an advanced tool for simulating and investigating real vehicular traffic in cities. The software is being developed for few years by the author of the paper and gained recognition and popularity among scientists all over the world. It can be used for conducting advanced experiments related to predicting, managing and optimizing vehicular traffic in cities. The paper introduces cellular automaton-based simulation model, TSF Model, inspired by well-known Nagel-Schreckenberg Model for simulating vehicular traffic on highways. The TSF Model was implemented as a core simulation engine in the TSF software. The paper also presents software's main features, functionalities, recently introduced improvements, plans for the further development and summarizes its previous applications.**

Keywords-**Vehicular traffic, traffic simulation, traffic modeling, intelligent systems.**

I. INTRODUCTION

Vehicular traffic is a very complex phenomenon which brings serious consequences to drivers, citizens and environment, especially in big agglomerations where the number of cars engaged in the city traffic increases every year. Traffic jams causes huge losses to budgets of many cities and countries and therefore it is very important to understand and learn how to cope with this phenomenon.

The most natural approach is, as in case of any other real, physical phenomenon, trying to gain as much as possible data about the traffic, analysing those data and drawing proper conclusions. However, in case of vehicular traffic it is still very difficult to obtain real, voluminous data about the phenomenon. Some producers of GPS navigations collect traffic statistics to improve their services ([7], [8], [9]), many municipalities use devices called Automatic Traffic Recorders (ATR [10], [11]) to collect traffic data from a number of selected road segments in the city and use it to make short-term and long-term predictions, for the purpose of driver navigation and urban planning (roadworks, new road construction etc.), as for example in the Autobahn Traffic project in the North Rhine-Westphalia, Germany [13]. Also, some web services display up-to-dated state of the road traffic, for example [14] and [15].

Collected data usually are not accessible to scientists, also they are often limited and it is difficult to make effective predictions when the traffic changes and the new traffic states haven't occured before. Also, we would like to learn how to manage and optimize the traffic effectively and not only to know the current and past states of the traffic. Therefore it is useful to apply mathematical models to describe the phenomenon. In the past, scientists were testing different mathematical models, usually searching for analogy between the traffic and other real, better known physical phenomenon, such as fluid dynamics, newtonian mechanics, kinetic gas theory and so [4]. The progress was made by introducing traffic simulators based on cellular automaton models. First such models were developed in the end of XX century and had property of realistic modeling vehicular traffic on highways [1], [2], [3]. Those simulation models were introduced for example in the Autobahn Traffic project in Germany [13].

During my research on modeling vehicular traffic in cities, I developed my own simulation model, also based on cellular automaton. The model (called TSF Model) was described in details in the paper [15] and implemented in the Traffic Simulation Framework software. Implementation of the software was a very challenging and difficult programming task. It required developing efficient algorithms for finding optimal and correct (compatible with reality) routes, conducting simulations, managing complex data structures with huge amount of data, integrating simulation engine with expanded interface, real maps and real road networks, manipulating input options and following simulations. The development process required great creativity, broad research on existing solutions and the whole domain of vehicular traffic.

The rest of the paper is organized as follows: the second section recalls fundamentals of the TSF Model and its connections with the Nagel-Schreckenberg Model. Sections III and IV present the software called Traffic Simulation Framework which implements the TSF Model. Section III describes features and functionalities of the TSF software, also its latest improvements, Section IV summarizes its previous applications. Section V concludes the paper and presents plans for the future development and research.

II. TSF MODEL

A. NAGEL-SCHRECKENBERG MODEL

Since TSF Model was inspired by previous models based on cellular automaton, it is required to recall the Nagel-Schreckenberg Model [1], which was a base to develop many other, more sophisticated simulation models [2], [3].

978-0-7695-4682-7/12 $26.00 © 2012 IEEE
DOI 10.1109/UKSim.2012.57

Nagel-Schreckenberg model emulates a single lane traffic. In this model space, time and velocities are discrete. The road is represented as a tape divided into identical cells. At any moment of the simulation, every cell may be empty or occupied by a single car. The state of the automaton in a time t may be represented by positions of cars. The car's velocity is its internal state and may take values from a finite set $\{0,\ldots, V_{MAX}\}$. The state of the automaton at time t +1 can be obtained from the automaton state at time t by applying the following rules to all cars at the same time:

1. Acceleration: $v_i(t+1) := \min(v_i(t)+1, v_{MAX})$
2. Braking: $v_i(t+1) := \min(v_i(t+1), d_i(t))$, where $d_i(t)$ is a number of empty cells in front of the car
3. Randomness: with probability p: $v_i(t+1) := \max(v_i(t+1) - 1, 0)$
4. Movement: car i moves forward $v_i(t+1)$ cells.

It is easy to see that all simulation steps correspond to the real actions taken during driving cars, the first step corresponds to acceleration, the second to braking. The third step introduces randomness which is crucial in the process of spontaneous jam formation [3]. It corresponds to random reduction of vehicle's speed due to weather conditions or driver's caution or inattention.

The experiments conducted using computer simulations showed that the NaSch model can reproduce traffic jams exactly in the same places as they occur in reality in case of single-lane highways [1], [3]. The model was broadly examined and generalized, for example to simulate traffic on 2-lane highways or in 2 directions [2].

B. *TSF Model*

This subsection introduces TSF Model which is extension of the NaSch model and enables conducting simulations on a real city network. The model inherits the most important properties of the NaSch model and introduces few interesting novelties:

- The road network is a directed graph (novelty)
- Every edge is divided into identical cells (as in the NaSch model)
- Every cell may be occupied by a single car or may be empty (as in the NaSch model)
- Time is discrete (as in the NaSch model)
- Space and velocities are not discrete, cars have also positions within the cell (novelty)
- Every car has its own start and destination points chosen from a given distribution (novelty) as well as time step in which the vehicle should start its drive (novelty)
- The route of a car is calculated using a routing algorithm, for example, Dijkstra algorithm or A* algorithm (novelty)
- Every edge may consist of more than 1 lane (novelty)
- The set of roads may be divided into separate classes, roads from the same class have the same distribution of vehicles speed and the number of lanes (novelty)
- Every driver has its own profile which influences maximal car's velocity (novelty)
- Some crossroads have traffic signals, signals located at the same crossroad are synchronized (novelty)
- Drivers always tend to increase the speed (as in the NaSch model)
- Drivers reduce vehicle's velocity before the crossroad. There are parameters which determine reduction in case of turning and going straight. Cars stop before the red light (novelty).
- Vehicles may change the lane (novelty)
- Vehicles must reduce the speed if they may collide with another car (as in the NaSch model)
- Drivers may randomly reduce the speed (as in the NaSch model)
- When the vehicle's speed is calculated and established, it moves similarly to the last step of the NaSch model (with possible turn, overtaking or switching lane)

More details about the TSF model can be found in [15] and [16].

III. TRAFFIC SIMULATION FRAMEWORK

Traffic Simulation Framework (TSF) is an advanced software for simulating and investigating vehicular traffic in cities. The software is implemented in the C# language using few important components of the .NET Framework technology [6] (i.e. Windows Forms) and maps from the OpenStreetMap project [5]. It has a very nice and intuitive GUI, but also there exists a command-line version without GUI which enables conducting advanced and time-demanding experiments. It can simulate motion of about 10^5 cars on the real road network in the real time on a standard desktop machine. Currently it implements simulation on a real road network of Warsaw.

Implementation of the TSF software was a very challenging and difficult programming task. It must have been preceded by a broad research in the domain of vehicular traffic on highways and in cities, modeling and simulating complex processes. The most difficult parts concerned designing efficient algorithms for generating realistic routes, simulating motion of realistic number (about 10^5) of cars, modeling road network as a cellular automaton. Also, designing intuitive graphical user interface and protocols for logging huge amount of simulated data was a very difficult and time-consuming part of the whole development process. The implementation uses sophisticated data structures, such as hash sets, dictionaries, priority queues, to store structure of the real road network or simulation data.

The software is still being developed. The next subsections cover the most important features and functionalities of the software and describe its latest improvements.

A. *Features and functionality*

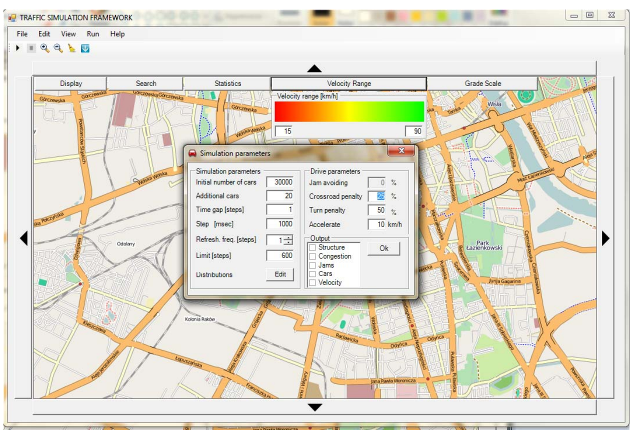

Figure 1: Editing simulation parameters and starting simulation

One of the most important and most impressive features of the TSF software is its Graphical User Interface which enables setting most of software parameters and following simulations. The GUI (presented on pictures Figure 1 and Figure 2) contains a city map taken from the OpenStreetMap project [5] which can be zoomed in and out and moved in four directions.

Through the GUI the user can see the road graph structure (nodes and edges), locations and states of traffic signals, distributions of start and destination points, positions and average velocities of cars and links monitored during the simulation. It is possible to see up-to-dated simulation statistics and specify velocity ranges.

The user can edit signal location and configuration, distributions of start and destination points, default street velocities, road links monitored during the simulation and simulation parameters such as: initial number of cars, number of cars which start move after every specified time period, duration of the simulation, relation between a single step and a real time, parameters which determine drivers' behavior before crossroads, parameters responsible for accelerating. It is also possible to specify type of data logged to external files during simulation (Figure 1). Values of all these parameters can be saved to the file or read from file at the GUI level.

The user can start, pause or stop simulation and observe it from 5 different zooming levels. It is also possible to generate and save new routes, to use it in the next simulation, as well as search for any street on a map by typing its name.

Most of these operations have been already described in details in the first paper about the TSF software [15].

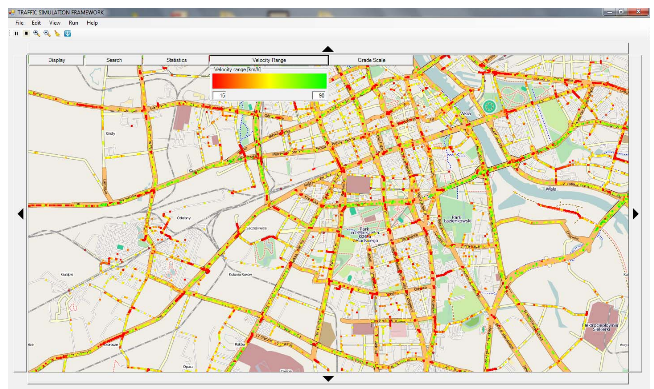

Figure 2: The main window of the TSF software

B. *Latest improvements*

The TSF software is still being developed and improved. In this section, we present the newest features, which haven't been earlier mentioned and published. The changes are consequences of many software applications which are described in the next section.

The biggest change is in type and way of logging data from the software during simulations to analyze it and investigate and predict vehicular traffic. Previously, it was possible to log simpler data, such as road congestions and average velocities. Currently it is possible to log to external files five different types of data:

- Road network structure:
  - Nodes identifiers, latitudes and longitudes
  - Edges: identifiers of beginning and end, length of the link, number of cells, number of lanes, average maximal velocities
  - Signals: identifiers of beginning and end, initial state, duration of green and red phases, switch of the phase
- Roads congestions:
  - Timestamp, edge identifiers, number of cars that passed the edge from the last timestamp

- Cars positions:
  - Timestamp, car identifier, edge identifiers, position on the edge (longitude, latitude and relative position – from beginning of the edge), car instantaneous velocity
- Jams formation:
  - Identifiers of edges on which traffic jams occur
- Average velocities:
  - Timestamp, edge name and identifiers, average velocity on the given edge from the last timestamp, number of cars used to calculate the average velocity

Those data are logged to external files, which can be specified from the GUI level or from the configuration file which is the next important improvement in the software. The configuration file is located in the software's main directory and can be edited before running the simulation. It contains default values of all crucial parameters (including name of input and output files) which are required to run the simulation or are optional parameters. By introducing this feature, it was possible to separate the GUI from the engine responsible for running simulations. Thanks to that, it is also possible to run simulations from the command line with parameters specified in the configuration file. The command line version of the software is more efficient and it is possible to run many instances of the software at the same time or to prepare a script which run many simulations one after another and with different parameters. The simulations may be longer and faster, logged data may be more voluminous.

The next improvement was made in selecting monitored road links - edges to log data from during simulations. Previously, selecting road links was possible only by marking rectangular region from which all edges were added to the set of monitored links, or by choosing (adding/removing) single edges manually. Currently, it is possible to specify any coherent route, from which all edges can be added to the set of monitored links.

The last significant improvement is related to changing distributions of start and destination points. This feature is very important, since it will be possible to simulate the real traffic throughout the whole day by running exactly one simulation. The user can specify many time periods with different distributions. Sequences of those periods may be saved to external file and read later for the need of another simulation.

There are also some minor improvements which make using the TSF software even simpler. For example, it is possible to log not only traffic data, but also state of the simulation. It helps tracing all important changes, finding and fixing possible inconsistencies.

IV. TSF APPLICATIONS

Traffic Simulation Framework is currently used by few scientists, whose research is related to modeling and optimization of vehicular traffic. Every new user causes new improvements which makes the software even better. So far, there were two important applications of the TSF software that should be emphasized:

- Generating data for the IEEE ICDM 2010 Contest on Traffic Prediction
- Testing genetic algorithms for optimization of vehicular traffic in cities

A. IEEE ICDM 2010 Contest: TomTom Traffic Prediction for Intelligent GPS Navigation.

The contest was held from June to September 2010. It was a data mining competition being a part of the IEEE International Conference on Data Mining 2010 (ICDM), which took place in Sydney, Australia, 14-17 December 2010. The contest was organized in cooperation with TunedIT platform [17], sponsored by TomTom, the world's leading provider of portable GPS and car navigation systems [9], held under the honorary patronage of the President of Warsaw. It attracted 575 participants, from whom over 100 submitted solutions, some of them are very respected scientists and specialists in data mining. The best and winning solutions were submitted for example by teams of scientists from IBM T.J. Watson Research Center, AT&T Labs Research, University of Warsaw, Duke University.

Participants had to solve 3 different, difficult tasks:

1. Traffic: the goal was to predict future road congestions based on historic ones.

2. Jams: the goal was to predict sequence of segments (road links) where next jams will occur in the nearest future, based on initial information about jams.

3. GPS: the goal was to predict average velocities on selected road segments based on historical notifications (locations and instantaneous velocities) from 1% of vehicles.

All data used by participants were generated using the TSF software. They were very voluminous and covered 1000 hours of simulations. It was a big challenge for the software and for organizers and a very good test of many software features. The contest was a success, it resulted in many interesting and efficient algorithms for traffic prediction. The best solutions were presented during the ICDM 2010 conference in Sydney in a special workshop dedicated to the contest and included in the conference proceedings [19], [20], [21], [22], [23], [24]. The whole contest was summarized in [25].

After the contest, the TSF software was made accessible to the public use to enable post-challenge research and attracted many other young scientists.

B. *Testing genetic algorithms for optimization of vehicular traffic in cities*

The second important application of the TSF software was testing a genetic algorithm for optimization of vehicular traffic in cities. Traffic lights configuration can be encoded as a genotype and evolutionary methods may be applied to find suboptimal solutions. In proposed approaches, the quality of genotypes is calculated using TSF software, which is the main novelty of the method. More precisely, to assess how good is the given traffic light configuration, it is sufficient to simulate traffic with this configuration applied to the city map and calculate and compare some key traffic features, such as average velocities or congestions on selected road links. The results of my research were presented in the paper [16].

V. CONCLUSIONS AND FUTURE WORK

I am planning to develop the TSF software further and work on methods for optimizing vehicular traffic in cities. My work on improving software will be directed to accelerate simulations and develop and test new methods for optimization of vehicular traffic in cities. I am developing efficient, scalable algorithms which integrate evolutionary approaches with adaptive strategies.

The next big challenge is conducting faster simulations. It can be achieved by introducing parallel or distributed calculations of crucial parts of code responsible for updating cellular automaton.

ACKNOWLEDGEMENT

The research presented in this paper has been partially supported by the research grant 2011/01/B/ST6/03867 from the Ministry of Science and Higher Education of the Republic of Poland and by the research grant 2011/01/D/ST6/06981 from the Polish National Science Centre. There are many people, especially from the Faculty of Mathematics, Computer Science and Mechanics of the University of Warsaw, who contributed in its creation by giving important ideas, advices on how to improve it and encouraged me to further work. Special thanks should go to my supervisor, Prof. Andrzej Skowron (University of Warsaw), Marcin Wojnarski (University of Warsaw & TunedIT), Dr Piotr Wasilewski (University of Warsaw), Dr Dominik Ślęzak (University of Warsaw), Prof. H. Son Nguyen (University of Warsaw), Prof. Marcin Szczuka (University of Warsaw) and many other people who have helped me at different stages of my work.

The software can be downloaded from my homepage: http://www.mimuw.edu.pl/~pawelg. Any questions and enquiries related to the software or my research should be addressed to: pawelg@mimuw.edu.pl .

REFERENCES

[1] K.Nagel and M.Schreckenberg, *A cellular automaton model for freeway traffic*, Journal de Physique 2 (1992), pp. 2221-2229.

[2] D.Chowdhury, A.Schadschneider, *Self-organization of traffic jams in cities: effects of stochastic dynamics and signal periods*, (1999), Physical Review E (Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics), Volume 59, Issue 2.

[3] A.Schadschneider (1999), *The Nagel-Schreckenberg model revisited, The European Physical Journal B*, Volume 10, Issue 3, pp. 573-582.

[4] A.Schadschneider (2000), *Statistical physics of traffic flow*, Physica, A285, p. 101-120.

[5] OpenStreetMap Project: An official OpenStreetMap Project website: http://www.openstreetmap.org.

[6] Microsoft .NET Framework: An official Microsoft .NET Framework website: http://www.microsoft.com/NET

[7] AutoMapa Traffic, http://www.automapa.pl/?PEI=210436&lng=PL

[8] Naviexpert, http://www.naviexpert.pl/

[9] TomTom Live Traffic, http://www.tomtom.com/livetraffic/

[10] Automatic Traffic Recorder, http://www.jamartech.com/ATRs.html

[11] Automatic Traffic Recorder, http://www.dot.state.mn.us/traffic/data/atr/atr.html

[12] Autobahn, http://www.autobahn-online.de/

[13] Google Maps, http://maps.google.com/

[14] Targeo.pl, http://www.targeo.pl/

[15] P.Gora, *Traffic Simulation Framework – a cellular automaton based tool for simulating and investigating real city traffic.* Recent Advances in Intelligent Information Systems, pp. 642-653, 2009

[16] P.Gora, *A genetic algorithm approach to optimization of vehicular traffic in cities by means of configuring traffic light,* ISMIS 2011, Emerging intelligent technologies in industry, Studies in Computational Intelligence, 2011, p. 1- 10

[17] TunedIT, http://tunedit.org

[18] ICDM 2010 Contest, http://tunedit.org/challenge/IEEE-ICDM-2010

[19] Carlos J. Gil Bellosta, *A convex combination of models for predicting road traffic*, 2010 IEEE International Conference on Data Mining Workshops, p. 1354-1356.

[20] B.Hamner, *Predicting Travel Times with Context-Dependent Random Forests by Modeling Local and Aggregate Traffic Flow*, 2010 IEEE International Conference on Data Mining Workshops, p. 1357-1359

[21] B.Hamner, *Predicting Future Traffic Congestion from Automated Traffic Recorder Readings with an Ensemble of Random Forests*, 2010 IEEE International Conference on Data Mining Workshops, p. 1360-1362.

[22] J.He, G.Swirszcz, Y.Kamarianakis, R.Lawrence, W.Shen, L.Wynter, *Ensemble-Based Method for Task 2: Predicting Traffic Jam*, 2010 IEEE International Conference on Data Mining Workshops, p. 1363-1365.

[23] L.Romaszko, *IEEE ICDM 2010 Contest: Traffic Prediction-Jams*, 2010 IEEE International Conference on Data Mining Workshops, p. 1366-1368.

[24] W.Shen, Y.Kamarianakis, L.Wynter, J.He, Q.He, R.Lawrance, G.Swirszcz, *Traffic Velocity Prediction Using GPS Data: IEEE ICDM Contest Task 3* Report, 2010 IEEE International Conference on Data Mining Workshops, p. 1369-1371.

[25] M.Wojnarski, P.Gora, J.Swietlicka, M.Szczuka, H.S.Nguyen, D.Zeinalipour, *IEEE ICDM 2010 Contest TomTom Traffic Prediction for Intelligent GPS Navigation*, 2010 IEEE International Conference on Data Mining Workshops, p. 1372-1374.