VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

CALIBRATION OF THE HIGH-SPEED TRAFFIC MICROSIMULATION

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE Ing. PAVOL KORČEK
AUTHOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

KALIBRACE VYSOKORYCHLOSTNÍ MIKROSIMULACE DOPRAVY

CALIBRATION OF THE HIGH-SPEED TRAFFIC MICROSIMULATION

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE Ing. PAVOL KORČEK
AUTHOR

VEDOUCÍ PRÁCE Prof. Ing. LUKÁŠ SEKANINA, Ph.D.
SUPERVISOR

BRNO 2016

Abstract

This dissertation thesis is focused on the high-speed simulation of road traffic and its precise calibration with different types of traffic data. The thesis surveys the state of the art in traffic simulation, with a special focus on traffic simulators and the data they utilize. A new traffic microsimulation model inspired by existing cellular automaton-based model is proposed and its features and parameters are discussed. The proposed acceleration techniques allowed the utilization our microsimulation model beyond the real-time simulation even for large scale systems. The complex process of quality assurance is tightly associated with the model calibration. A brand new calibration methodology that is based on the evolutionary algorithms is proposed. It was found that the proposed method does not require a sensitivity analysis. The method is also capable of determining the parameters that are hard to estimate with other well know techniques. The quality of the resulting optimized models was experimentally analyzed and discussed at both the macroscopic and microscopic traffic levels. The method is robust enough to deal with missing samples and other imperfections of the real data. Properties of the proposed model such as the scalability and accuracy were evaluated with respect to the existing solutions, showing a considerable improvement of the state of the art. The software tools that were created within the research are briefly presented. Finally, the thesis also outlines further steps of research and development, especially the use of the proposed tools in practice.

Keywords

microsimulation, traffic, cellular automaton, calibration, evolutionary algorithm.

Citation

KORČEK, Pavol. *Calibration of the high-speed traffic microsimulation*, PhD thesis, Department of Computer Systems, Faculty of Information Technology, Brno University of Technology, Brno, CZ, 2016.

Abstrakt

Tato disertační práce je zaměřena na vysokorychlostní simulace dopravy a jejich přesnou kalibraci pomocí různých typů dopravních dat. Práce se po úvodním popisu motivace pro samotný výzkum nejdříve věnuje současnému stavu poznání, a dále rozdělení simulátorů dopravy, zejména podle typu dat, se kterými se v nich pracuje. Úpravou existujícího řešení je navržen vlastní mikrosimulační model, který je založen na principu celulárního automatu. S tímto novým modelem je pak experimentováno, zpočátku z pohledu rychlosti simulace a další rozšiřitelnosti. Je navržena a popsána technika, kterou je možné navržený model významně akcelerovat a následně provádět simulace rychleji než v reálném čase i pro rozsáhlá území. Práce dále přistupuje k samotné kalibraci modelu, ke které byl využit evoluční přístup. Je představena metoda pro efektivní způsob optimalizace parametrů mikrosimulačního modelu, která nevyžaduje citlivostní analýzu a je schopná nalézt jinak obtížně nastavitelné parametry modelu. Kvalita získaných optimalizovaných modelů byla analyzována jak pomocí makroskopických, tak i mikroskopických dopravních dat, a to i s ohledem na jejich reálné vlastnosti, tj. chybějící vzorky. Dále jsou zhodnoceny výkonnostní a jiné kvalitativní parametry vlastního přístupu v porovnání s existujícími řešeními, přičemž bylo dosaženo významného zlepšení. Nakonec jsou představeny nástroje, které v rámci řešení vznikly. Na závěr je uvedeno další zaměření výzkumu, a to zejména s ohledem na využití výsledků práce v praxi.

Klíčová slova

mikrosimlulace, doprava, celulární automat, kalibrace, evoluční algoritmus.

Citace

KORČEK, Pavol. *Calibration of the high-speed traffic microsimulation*, disertační práce, Ústav počítačových systémů, Fakulta informačních technologií Vysokého učení technického v Brně, Brno, CZ, 2016.

Calibration of the high-speed traffic microsimulation

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně pod vedením Prof. Ing. Lukáše Sekaniny, Ph.D. a školitele specialisty Doc. Dr. Ing. Otto Fučíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

....
Pavol Korček
19. května 2016

Contents

Chapter 1

Introduction

This chapter starts with the motivation for the whole research presented in this thesis. Then, the research area and objectives of the thesis are formulated. An outline of the thesis is given at the end of the chapter.

1.1 Motivation

A modern world without cars or road transport in general is something we are simply unable to imagine. Cars affect everyone on a daily basis and many industries are strongly dependent upon them.

The history of modern road transport, or more specifically vehicles with combustion engines, dates back to the end of the 19th century. Modern cars developed from the era of self-propelled carriages propelled by steam engines. The cars were initially utilized by individuals or small groups of fans, while the masses simply didn't believe or even accept the concept at that time. The society began to change their attitude over time and thanks to persons such as *Karl Benz, Gottlieb Daimler, Wilhelm Maybach, Armand Peugeot* and *Henry Ford*, the car soon became engrained as an integral part of everyday life [29].

Road transport has brought forward an acute need for rules as the cars were growing in popularity. This was the impetus for creating the first road traffic regulations and laws. Then it became necessary to manage traffic at the busiest road connections – at the crossroads. The police initially took responsibility for this task, but were later replaced by light signals and other automated traffic control systems. Road transport today continues to develop dynamically. The latest technologies are developed and deployed, in the areas of transport infrastructure, navigation systems and traffic management and even in cars themselves. All these aspects are integrated under a notion of *Intelligent Transportation Systems*, or *ITS*. These systems monitor road traffic to a fine level of detail and can even intervene when needed.

The number of vehicles continues to grow in all segments, including both passenger cars and trucks or other types of vehicles. Technological advances have been the primary driver of such an increase and have resulted in safer and less expensive vehicles that are now available to a larger group of people. The car ownership is now the standard in advanced countries. However, the number of vehicles continues to rise at a rate that is much faster than the pace at which existing roads are expanded or new roads are built. While the overall road network has only been expanded by a few kilometres per year (there were a total of 55,747.6 km of roads in the Czech Republic in 2014, an increase of only 240 km over

2005), the number of cars has increased by nearly 25% over the same period of time (again in comparison with 2005, with 4.83 million registered cars in 2014) [26]. Investments into transport infrastructure also continue to drop year-on-year, with the amount invested in 2014 only 40% (less than half) of the amount invested in 2005. Conversely, costs continue to rise for the road network repairs and maintenance due to the need to maintain older roads as they age [30].

This situation results in an unfortunate increase in road traffic density, which has an impact on transport itself. The denser traffic increases the probability of accidents and other collisions while the busiest areas of roads and crossroads suffer from gridlock and congestion [1, 21]. This phenomenon is regularly occurring at some locations and is a daily problem.

Secondary problems also co-exist. The cars stuck in long traffic jams move slowly, wasting fuel and spewing large amounts of polluting exhaust gases into the atmosphere [7]. These contain hazardous substances including carbon dioxide and nitrous oxides that have a negative environmental impact. There are also economic consequences. Traffic, which causes delays, thereby results in an overall increase in transport costs. Last but not least, congestion actually extends the time needed to transport people. All these aspects have mobilised countries to organise and make greater efforts to find solutions to the current problems faced by transport. These issues can be resolved in three ways in a general sense:

(a) mitigating the negative impacts of transport,

(b) media campaigns and official regulations, and

(c) modernisation and the construction of new roads and/or vehicles.

While efforts to mitigate the negative impacts of transport (a) have begun to bear fruits recently, for instance, modern technology has been applied to vehicles to ensure they comply with standards [28], the same cannot be said about media campaigns and official regulations (b). Media campaigns and official regulations seek to convince and even force the general public to act in a more ecological and efficient manner when it comes to transport. This primarily involves media campaigns in support of public transport. Recent statistics actually show a decrease in the number of public transport passengers, while nearly all of those public transport passengers have moved to personal cars [26, 30]. The final measure (c) could be of assistance in light of the current situation where excessively high road traffic density is responsible for traffic congestion to a large extent. Modernisation and construction of new roads is much more financially intensive and is not always acceptable. The open space needed to build or expand roads is not often available. The new construction would lead to additional devastation of the countryside in many rural areas. The inadequacy of this solution is confirmed by the fact that financing for transport infrastructure continues to drop [26]. The situation is better in the case of construction of new vehicles. These are nowadays integrated with modern communication features known as *Vehicle-to-infrastructure (V2I)*, *Infrastructure-to-vehicle (I2V)* and/or *Vehicle-to-vehicle (V2V)* that can perhaps help to avoid some kind of collisions. However, such a technology is only in its early adoption stages.

1.2 Research Area

There is another way to efficiently increase the network capacity without constructing new roads. Detailed traffic analysis conducted on a specific road segment may generate an optimum traffic control strategy that increases the capacity and decreases the traffic density [38, 4, 21]. The gradual integration of a number of roads, cities and towns may eventually drive the optimisation of the entire transport network in a city, district or region. The optimisation includes many targets such as modifying speed limits, changing rules with respect to the right-of-way, modifying the timing at crossroads, implementing dedicated lanes of travel and much more. And not only the static or permanent optimisations, but also adaptive changes during operations are possible by the deployment of modern ITS (traffic reports, roadway information signs, portable signs, etc.). This means using different configurations (e.g. speed limits noted above) to adapt a specific road segment to different needs at different times. This dynamic road traffic management can be seen, for example, in the article from 2013 [11].

Moreover, road traffic analysis tools can be also utilized in so-called *Advanced Traveller Information Systems - ADT* (e.g. online traffic applications used in smartphones and other portable devices). This tools can help drivers to make the decision making process significantly easier on the basis of their own knowledge and the information provided from such systems. Drivers informed, e.g. about collisions, may decide to bypass a critical place when there is still time to do it and thus avoid additional traffic problems. Overview of such systems can be found in the separate article from 2015 [36].

Road traffic simulators are used to assist in the decision making process when selecting a specific strategy. Two fundamental prerequisites are critical with respect to traffic simulators used for these purposes. The first is that the simulator must be able to simulate the given road section at a speed that is faster than real time for the needs of forecasting. Sometimes the simulations must be conducted at a much faster speed to test a number of optimisation strategies and identify the best solution in a real time. The second and clearly logical prerequisite is that such a simulator must be very accurate. Both these prerequisites, however, counterbalance one another. More accurate simulators are capable of modeling traffic at a detailed microscopic level (see below). These simulators define very detailed relationships between the individual entities involved in transport. As there are many such entities, these simulators are very demanding in terms of computing resources.

Every simulation model must be calibrated (and validated) against relevant data, ideally the data coming from the specific road section, before its deployment in the real world. A problem may occur with respect to transport where the data is not available at the required level. Such situation occurs when monitoring sensors used to generate the data are not available for the given road section. Another problem connected with the use of the real data is that despite deploying suitable sensors, the resulting data samples may be lacking or missing for a specific period of time. Inclement weather (camera fogging, etc.) or a simple hardware or software fault in the sensor may be responsible for such a situation.

1.3 Research Objectives

The main research objective for this thesis is to

„develop a fast and accurate traffic simulator, capable of multi-real-time simulations of complex traffic areas.“

As indicated, such a simulator must meet two basic requirements. It must be able to perform simulations faster than in real time. Moreover, it must be able to do multiple in real time simulations due to the necessity of various road traffic scenarios simulation.

The second requirement is that it must be very precise, e.g. the accuracy should be better than current state of the art. Finally, its calibration must be supported for several traffic facilities and different traffic data as they occur in the real world (e.g. the operation with missing data has to be supported).

The main research objective can then be re-formulated in several sub-goals. The first sub-goal is *to find a proper road traffic simulation model* and *to develop a basic version of a simulator based on the chosen model*. Then, it will be needed *to accelerate the simulator* in the case it is too slow and unable to perform the multiple in real time simulations. For the quality assurance, *a calibration methodology needs to be proposed and evaluated*, not only on macroscopic data, but also on microscopic data sets. For a real deployment of the road traffic simulator, *a methodology has to be developed to deal with incomplete field data sets*.

1.4 Thesis Outline

The thesis is composed as a collection of papers. The research contribution of this thesis is thus constituted by the five peer-reviewed research papers, in their original publication format. The thesis is organised as follows: Chapter 1 (this chapter) gives an introduction to the thesis. Chapter 2 surveys the state of the art, and presents relevant background information for the research. Chapter 3 summarises the research process and gives an overview over the papers constituting the research contribution. And finally, Chapter 4 presents conclusions and proposes future research directions.

Chapter 2

Survey of the State of the Art

This chapter surveys the state of the art and gives the relevant background information needed for a proper understanding of the work presented in the thesis.

2.1 Road Traffic Simulation

A road traffic simulation model represents a highly dynamic system, where the time constitutes a primary independent variable. In general, simulation models describe how each entity of a system reacts to stimuli and thus develops its state in time. Discrete simulation models are representative of real-world systems in such a way that the conditions suddenly change right at designated time points. There are two types of these models [25, 33, 5]. Models with a discrete time divide the time into intervals of a fixed length. These models calculate an activity in each interval (e.g. for vehicle acceleration, deceleration) that changes the state of an entity. On the other hand, some systems don't activate the entities until the moment when their actual change of the state is planned. Here we speak about models with discrete events where the events are triggered.

Due to the fact that this event triggering can effectively be scheduled, it is possible to save computing power. However, it does not apply to traffic simulators, because the number of triggered events is always high. Microscopic and macroscopic models are basic road traffic simulation models. They are described in detail in next sections.

2.1.1 Microscopic Models

Microscopic traffic models describe each entity and each interaction individually. For every vehicle in the road network, it is separately calculated how it will behave in the following step on the basis of the surrounding conditions. As these models deal with every vehicle on the road section, they manage to be very accurate.

On the left-hand side of Figure 2.1, on the vertical axis x, the position of the vehicle with label a can be seen. The vehicle positioned in front of this vehicle is described as $a+1$. Since both vehicles are moving on the road, their positions are time dependent. Both positions are therefore valid at the time t_0 denoted on the horizontal time axis t.

The position of a vehicle monitored during a certain period of time is called the trajectory. Because every vehicle has a certain length, a certain point of the vehicle, for example the rear bumper, is used as the reference point of this trajectory. The length of the vehicle itself is the constant L_a. It is also clear that different types of vehicles have a different length L. In Figure 2.1 the trajectories of both vehicles a and $a+1$ are marked with bold

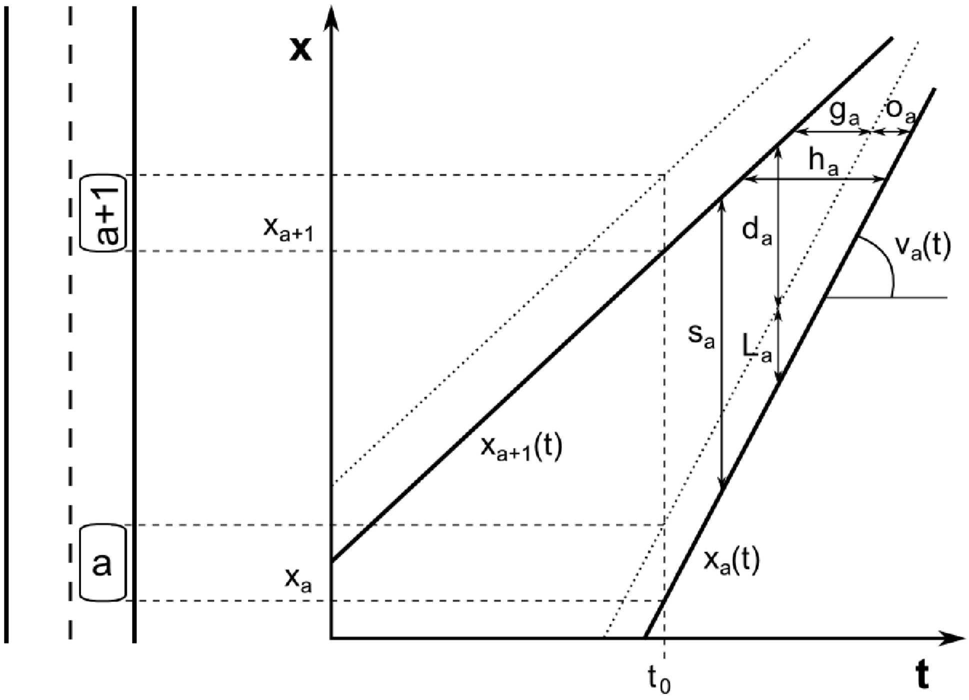

Figure 2.1: Microscopic level of modeling.

lines. The space, which the vehicles occupy, is bordered with dot lines. If the two trajectories intersect, the passing of a vehicle occurred at this point. The speed of vehicle a is defined by its trajectory derivation (Eq. 2.1):

$$v_a(t) = \frac{dx_a(t)}{dt} \tag{2.1}$$

The acceleration a_a of a given vehicle is obtained through the second derivative, whereas with an accelerating vehicle this value is positive, and oppositely, with a slowing vehicle (braking) this value is negative (Eq. 2.2):

$$a_a(t) = \frac{d^2x_a(t)}{dt^2} \tag{2.2}$$

Each vehicle occupies a certain part of the road section. However, this value does not represent only the vehicle length (L), but it also includes the space that each driver maintains behind the preceding vehicle. For the vehicle a the total area s_a consists of vehicle size L_a and distance d_a, which is maintained between vehicles, and is also time-dependent (Eq. 2.3 and Eq. 2.4):

$$s_a(t) = x_{a+1}(t) - x_a(t) \tag{2.3}$$

$$s_a(t) = d_a(t) + L_a \tag{2.4}$$

Each vehicle takes some (travel) time to pass through the given section. This interval, known as h (headway) consists of the time g (gap) necessary to overcome the space between the vehicles and time o (occupancy) needed to overcome its own vehicle length L (Eq. 2.5):

$$h_a = g_a + o_a \tag{2.5}$$

If we assume a constant speed, or if we are just not interested in the acceleration of the vehicle, it is possible to write (Eq. 2.6):

$$o_a = \frac{L_a}{v_a} \tag{2.6}$$

The speed difference of two successive vehicles can then be expressed as (Eq. 2.7):

$$\Delta v(t) = v_{a+1}(t) - v_a(t) = \frac{ds_a(t)}{dt} \tag{2.7}$$

Many of the above mentioned parameters can be obtained by monitoring real traffic conditions. For example, two aerial images of the same area taken successively can give us the position x, speed v and intervals h, g and o. More feasible sources of information are, compared to the aerial images, inductive detectors and detection cameras. For example, by using inductive detectors placed on the roads, it is possible to obtain velocity v, space s, and with the cooperation of detection cameras, the length of vehicle L can be recognized. Moreover, it is possible to obtain headways (i.e. travel time for every vehicle) using (at least) two detection cameras with a vehicle plate recognition feature integrated. Such cameras have to be placed at the beginning and the end of the monitored segment.

Roughly a decade ago there was a huge increase of interest in the traffic models, which looked at the traffic situation on the microscopic level [3]. This was mainly due to the increase in the processing power of personal computers, which enabled performing microscopic simulation in a quite reasonable time. Currently, there are many microsimulation transport tools: from small, educational programs, where the entry and exit into the simulator is commonly obtained via the command line, to academic projects and commercial simulators. These latter ones are often based on successful academic projects, and have, among other things, also a greatly reworked graphic user interface. Examples of traffic microsimulators include (without reference) *AIMSUN, CORSIM, DRACULA, FREESIM, HUTSIM, INTEGRATION, MITSIM, NETSIM, PADSIM, PAMINA, PARAMICS, SITRAS, TRANSIMS* or *VISSIM*. More details regarding the microsimulation models can be found in specialized literature (e.g. [3, 20, 47]).

There are models (called *submicroscopic*) that try to model the reasons for which individual simulated entities change their behavior. These may be the breaking performance, engine speed, speed of gear-change, and many others. Some of the above mentioned models are complemented by such submicroscopic properties. Therefore, these models require even more time for the simulation than the pure microscopic ones.

2.1.2 Macroscopic Models

Macroscopic models describe traffic on a very high level with just a few equations [47]. Instead of individual entities, aggregated values are modeled and, therefore, we are not interested in vehicles as separate entities. The macroscopic level of description of traffic is a dynamic description of the entire monitored section and is designated especially for strategic purposes (planning new roads, etc.). Macroscopic variables are always analyzed over a particular area of interest (Figure 2.2). This area (or even interval) means a certain space in the graph of trajectories. It is possible to define three such areas, along their characteristic variables.

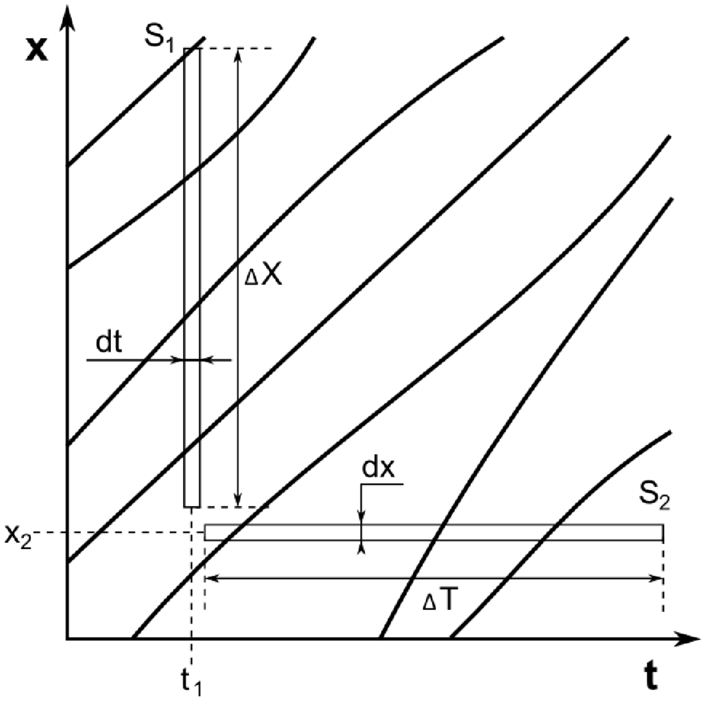

Figure 2.2: Macroscopic level of modeling.

Oblong area S_1 covers the road section with a length of ΔX during the infinitely small time interval dt. The interval corresponds to the state of section ΔX exactly at the moment in time t_1 and gives us information about the position of n vehicles at a given time in the given section. Each such vehicle will be assigned index i in the following text. The area S_1 may be obtained by the aerial image of the road section. The characteristic variable is the density k, which reflects the number of vehicles on a given road section. In the given location x_1, time t_1 and in a certain interval defined by the area S_1, the density is defined as (Eq. 2.8):

$$k(x_1, t_1, S_1) = \frac{n}{\Delta X} \tag{2.8}$$

The total occupied space by n vehicles is ΔX, and, therefore, for density k it applies that (Eq. 2.9):

$$k(x_1, t_1, S_1) = \frac{n}{\sum\limits_{n} s_i} = \frac{1}{\bar{s}}, \tag{2.9}$$

where the vehicle in an average occupied (in the section ΔX) space $\bar{s}$ is defined as (Eq. 2.10):

$$\bar{s} = \frac{1}{n} \sum_{n} s_i \tag{2.10}$$

Through a generalization of expression (Eq. 2.8) by extension for the infinitely small time dt, we obtain (Eq. 2.11):

$$k(x_1, t_1, S_1) = \frac{n \cdot dt}{\Delta X \cdot dt} \tag{2.11}$$

Another area highlighted in Figure 2.2, labeled S_2, is the infinitely small road section dx during time interval ΔT. Again, this infinitely small road section can be seen this time as a single position marked as x_2. The m trajectories, created by m vehicles, intersect through the position x_2. Each vehicle will be assigned index j. In order to obtain area S_2, it is necessary to have induction detectors and detection camera systems to record the vehicles in time at two points on the road section.

A typical variable for area S_2 is traffic flow – q. It represents the number of vehicles passing a certain road point at a given time. In a given location x_2, time t_2 and in a certain interval defined by area S_2, this flow is defined as (Eq. 2.12):

$$q(x_2, t_2, S_2) = \frac{m}{\Delta T} \tag{2.12}$$

where m represents the number of vehicles crossing the road point x_2 during interval ΔT. On the other hand, this interval is the sum of the time needed for m vehicles to make the transition to the position of the previous vehicle – h. On the basis of the value $\bar{h}$, the flow is defined as (Eq. 2.13):

$$q(x_2, t_2, S_2) = \frac{m}{\sum\limits_m h_j} = \frac{1}{\bar{h}} \tag{2.13}$$

where the average value $\bar{h}$ is (Eq. 2.14):

$$\bar{h} = \frac{1}{m} \sum_m h_j \tag{2.14}$$

The maximum possible traffic flow q is also called capacity c of the given road section. Through a generalized expression 2.12 in the form of the extension for the infinitely small time interval dx, we obtain (Eq. 2.15):

$$q(x_2, t_2, S_2) = \frac{m \cdot dx}{\Delta T \cdot dx} \tag{2.15}$$

The last of the possible areas is shown in a separate Figure 2.3 as S_3. It defines an area in the final time and space. Several trajectories are crossing this area. The distance X_3, that the vehicle goes through in the area, is the projection of the trajectory on the axis x. Conversely, time T_3, during which the vehicle was located in the monitored area, is a projection of the trajectory on the timeline t. The last of the traffic fundamental variables is the average speed u. It is defined by means of the aforementioned variables. It is a function of the place, time and area (Eq. 2.16):

$$u(x_3, t_3, S_3) = \frac{q(x_3, t_3, S_3)}{k(x_3, t_3, S_3)} \tag{2.16}$$

Finally, expression 2.16, rewritten to formula 2.17, is seen as a *fundamental relationship* of the traffic flow theory:

$$q = k \cdot u \tag{2.17}$$

On the border of microscopic and macroscopic traffic simulation models, mesoscopic ones can also be found. They analyze traffic at the level of small groups of vehicles, but not at the level of individual vehicles (as it is with microscopic ones) or the whole road section (as it is with macroscopic ones).

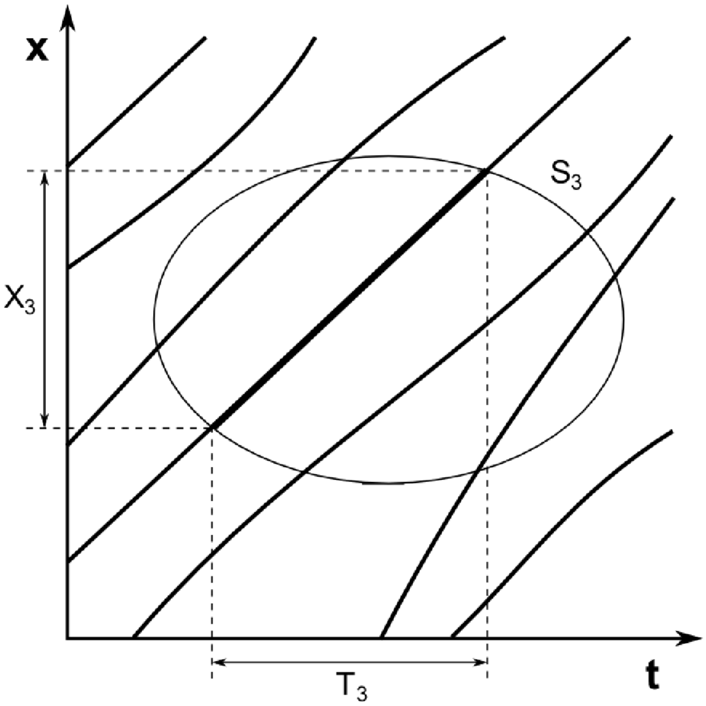

Figure 2.3: Trajectories of the vehicles in time and space for macroscopic modeling.

2.1.3 Other Properties

Some traffic simulators do not include any random components and, therefore, each entity is modeled using an exact relationship expressed by mathematical equations. Such a model becomes deterministic. Conversely, models that contain a random component are called probabilistic traffic models. Some actions in these models take place with a certain probability and the resulting behavior is therefore not possible to determine as precisely as with the models of the previous type. Probabilistic models are useful in the case where we do not have a detailed information about the modeled area or even when we cannot find out all the information from modeling entities.

Another way the traffic simulation models can be categorized is according to the geographical scope of the model. Some models are intended only for specific purposes, for example, for the analysis of the utilization of a particular intersection, or a particular roundabout. Their impact is, therefore, local. Other models are instead focused on a wider area and they even analyze whole towns. It is evident that for the simulation models capturing more extensive road sections, performance problems appear on account of a greater number of modeled entities. On the other hand, modeling a larger area is usually preferred to get a better overview of the traffic.

Finally, traffic simulators also differ with respect to entities they model (in the case of microsimulation models). Most simulators are devoted to personal or commercial vehicles. However, it is also possible to meet with specific simulators which include pedestrians, cyclists and other entities in their models. The simulation model then becomes very complicated as well as the requirements for computing resources are increasing.

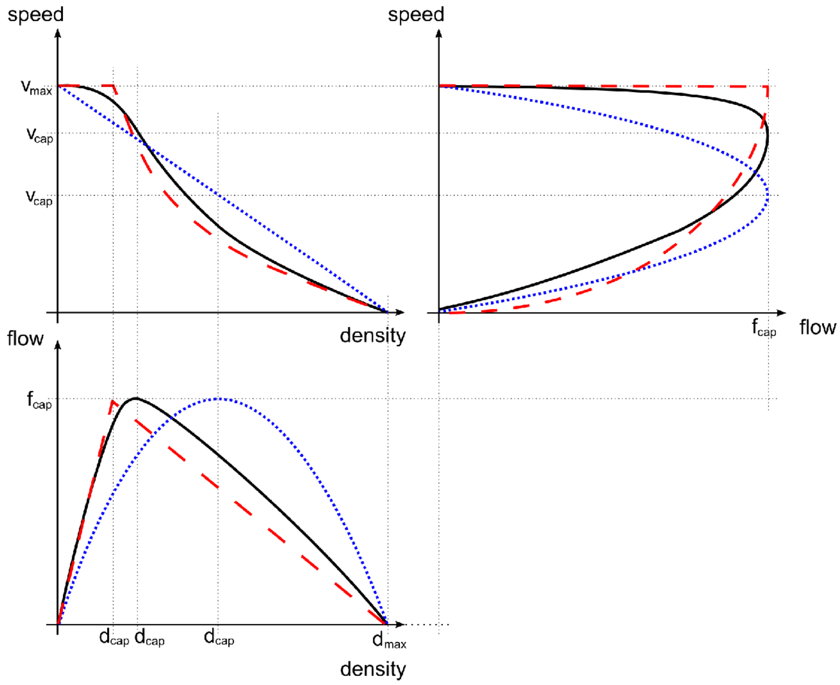

Figure 2.4: Three fundamental diagrams: Greenshield's *(blue dotted lines)*, Pipes's *(red dashed lines)* and Van Aerde's *(black full lines)* models.

2.1.4 Quality of Simulation Models

Any simulation model must be calibrated prior to its deployment and usage. Model calibration can be defined as a process in which the input parameter values are being tuned in order to reflect modeled traffic conditions at required level of precision. A common practice is to utilize a real data obtained directly from the modeled sections [3, 20, 47]. A close attention to this process is needed in order to perform it correctly [2]. Many guidelines have been created by international authorities [17] and other entities for these purposes [18, 45]. However, since it is not always possible to get a real data, *traffic fundamental diagrams* are often used for the process of calibration. These diagrams are reflecting the fundamental equation (see Eq. 2.17).

Greenshield was the first, who in 1934 came up with a formulation of mathematical relationships between traffic variables (see blue dotted lines in Figure 2.4). As the original formulation was based on a very small sample of data, the model was too oversimplified [46]. It has been shown that some of the relationships in his fundamental diagram do not correspond to reality. Despite this fact, this model is still used for the traffic simulation models calibration and/or validation because of its simplicity. *Pipes* came up with a different formulation for a fundamental diagram in 1953 (see red dotted lines in Figure 2.4). His model accepts a variety of traffic behaviors with and without congestion [46]. However, other negative characteristics of the original model were retained. The last, and probably the most accurate model from Van Aerde that seeks to address the previous problems was created in 1995 [46] (see black lines in Figure 2.4). In this model, the speed of the vehicle reacts to the current density. This is beneficial for many road sections (tunnels, slip roads, etc.). As majority of these models, this model has been developed primarily for freeways.

Because traffic simulation is not generally intended for high speed road sections, some of the current traffic simulators have begun to solve problems with their quality through the so-called online simulation [6]. They use the principle of vehicle injection at particular points of the road network. If the results of the simulation in a given road section differ from the real state, they either simply „subtract“ the vehicle at the given location, or, conversely, dynamically „add“ it. In this case, the simulation model does not need to be very precise. On the other hand, the actual injections, the number of which can be relatively high, significantly restrict the use of the simulator.

2.1.5 Scalability and Speed of Simulation Models

One of the problems of microscopic traffic simulators is their performance. None of them has primarily been designed for long-term forecasting. Hence, the models are not suitable for integration to ITS for traffic state predictions [10, 5]. Some of them were accelerated in order to achieve higher performance and to deploy them in larger areas and to enable multiple real time execution. However, as the authors claimed, the obtained results have never achieved a sufficient quality [22].

The acceleration was enabled by specialized equipment [55] or parallelization by means of threads which can be effectively mapped into a multicore processor (e.g. [27]). However, the concept is not primarily designed for accelerating the simulation. The primary goal was to expand capabilities for simulating larger areas. The so-called modular approach is another main direction in the acceleration of traffic microsimulation programs. In this case, the entire simulation network is divided into several sub-networks which are simulated independently [13].

The presented techniques can be employed to accelerate other types of simulation models [10, 5]. The possibility of the parallel microsimulation in such a way that the used parallelization technique can be implemented on a multicore system, should be an inherent feature of modern microsimulation models [12]. It means that a good traffic simulation model has to be constructed with the aim of parallel processing instead of making some effort to accelerate an inappropriate model. The acceleration of microsimulation model is seen as a quite complicated task and extra effort has to be invested to it. As a result, some new methodologies and strategies are gradually emerging [42].

2.2 Cellular Automata in Traffic Microsimulation

Cellular automaton (CA) is a computational model inspired in biology and developed to model massively parallel, locally interacting and self-replicating systems [49]. As defined, for example in [50], it is a d-dimensional grid of finite automata (known as cells) that operate according to their *local transition functions*. The cells work either synchronously, where each state of every cell is calculated from its previous state and the previous states of cell's neighbors synchronously with a synchronization signal, or asynchronously, where the cells can change their state independently of each other (e.g. [54]). By a configuration of the cellular automaton we mean the state of all the cells at a given moment. The global behavior is captured in the *global transition function*, which defines a transformation from one configuration to the next configuration of the cellular automaton. If all cells operate according the same local transition function (rule), the automaton is called *uniform*. Otherwise, it is called *non-uniform*. This type of CAs was introduced, because it can show a more complicated behavior than uniform ones. The rules can be deterministic or probabilistic. In general, the CA model supposes that the number of cells is infinite. However, in the case of practical applications, the number of cells is finite. Then it is also necessary to define the *boundary conditions*, i.e. the setting of the boundary cells.

A one-dimensional binary non-uniform cellular automaton with a finite number of cells can be defined as the 7-tuple [50]: $\mathsf{A} = (Q, N, R, z, b_1, b_2, c_0)$, where $Q = \{0, 1\}$ is a binary set of states, N denotes a neighborhood ($N \subseteq \mathbb{Z}$), z denotes the number of cells, b_1 and b_2 are boundary values, c_0 is an initial configuration, and a mapping $R : S \to (Q^N \to Q)$ assigns to each cell of the grid $S = \{1, 2, \ldots, z\}$ a local transition function $\delta_1, \ldots, \delta_z$, where $\delta_i : Q^N \to Q, i \in S$.

A configuration of A is a mapping $c \in Q^S$ which assigns a state to each cell of A. If only a single neighborhood $N = \{$-1, 0, 1$\}$ is considered, then the *global transition function* $G : Q^S \to Q^S$ is defined as:

$$G(c(i)) = \begin{cases} \delta_i(c(i-1), c(i), c(i+1)) & \text{for } i = 2 \ldots z-1, \\ \delta_1(b_1, c(1), c(2)) & \text{for } i = 1, \\ \delta_z(c(z-1), c(z), b_2) & \text{for } i = z, \end{cases}$$

where c_i denotes the cellular automaton configuration in a step i. G is used to define a sequence of configurations c_0, c_1, c_2, ... such that $c_j = G(c_{j-1})$, for $j \geq 1$. This sequence represents the *computation* of A.

Cellular automata have been applied in many scientific areas, especially for the modeling of complex (biological, chemical, computational, etc.) phenomena. They enable the modeling of discrete dynamic systems, in particular those in which the interaction of a given element is limited to its local surrounding. The global behavior is caused locally by many interacting elements. It can be observed that microscopic traffic models share many properties with CA.

The first microsimulation traffic model based on CA was introduced in 1992 and used to simulate a one-directional cyclically bounded section of the road [40]. In this traffic simulation model, each cell represents a certain part of the road section at a constant length (7.5 meters). The local transition function of the cell defines a new state based on the current status and availability of several neighboring cells. Each cell maintains the information about presence of a vehicle. If a vehicle is present, its parameters (e.g. vehicle speed in the original model [40]) are given. The distance, which the given cell (i.e. driver in the vehicle) can monitor, is defined by the number of neighboring cells. A CA cell may also define the type and, for example, the limitations of the road section, which it represents. The local transition function can be formulated, provided that each vehicle in the cell has a certain speed in the defined range and it is possible to determine the free space (the number of free cells), in front of (and in the event of overtaking also behind) the occupied cell.

Reflecting the criticisms, the CA based traffic simulation models have been modified in many different ways [32, 48, 16, 58, 57]. As the model parameters can be tuned it is possible to change properties of the model. It was shown that CA-based models can show a non-trivial and realistic behavior, while also being capable of reproducing all the phenomena that occur in traffic [40, 32, 48, 16]. Because of their simplicity, they are capable of performing several million updates per second even on an ordinary computer. This means, among other things, that they can be used to simulate very large traffic networks or they can be used for simulation of the areas with a very high traffic density (the number of simulated entities). They are, therefore, very useful in the area of the traffic microsimulations. Their simplicity predisposes them to parallel implementations on multicore computers or, for example, for implementations directly in specific integrated circuits [55].

2.3 Evolutionary Algorithms for Model Calibration

Evolutionary algorithms (EAs) are stochastic optimization algorithms inspired by biological evolution observed in nature [9]. The main branches are genetic algorithms, genetic programming, evolution strategies and evolutionary programming.

Candidate solutions, also called *individuals*, are usually coded as strings of symbols – the so-called *chromosomes*. At the starting point, a collection of individuals *(population)* is generated randomly. In order to evaluate the solutions encoded in the population, a mapping (or decoding) process is usually performed firstly. This transforms every chromosome to a candidate solution – the so-called *phenotype* which can be then evaluated.

The *fitness function* gives a measure of how good a given solution is for the target task. If the fitness of candidate solution is measured with regard to multiple objectives, the algorithm is called the *multi-objective optimization*.

There are several ways how to select individuals for creating a new population. A common way is to use fitness-proportionate selection, where an individual is given a probability of being drawn for reproduction proportional to its fitness score. An alternative to this is *tournament selection* where a number of individuals are drawn randomly and the individual with the highest fitness score of this selection is chosen for reproduction. The selected individuals are subject of various operations before they are inserted into the population of individuals constituting the new generation. The most common form of the variation operators used is called *mutation*. By applying this operator, a small change in the chromosome is performed. Recombination is done using the *crossover* operator, which swaps and exchanges some parts of two parent chromosomes. A common way of speeding up the whole process of evolution is to include *elitism*. This performs transferring the fittest individual of a given generation to the next one. However, this usually depends on the context and there may be situations where the elitism is not advantageous. A new population is established from newly generated individuals and some members of the previous population.

Finally, a skeleton of a typical EA can be described with the following steps:

1. Randomly generate a population of individuals.
2. Perform a genotype–phenotype mapping process on each individual.
3. Evaluate the obtained phenotypes with regard to the fitness function and assign scores to individuals.
4. If the termination criterion is reached, terminate the algorithm.
5. Select individuals according to their fitness scores.
6. Apply genetic operators (crossover, mutation, replacement) on selected individuals to create a new population.
7. Repeat from step no. 2

Genetic algorithm (GA) is one of the most popular variants of EAs [23]. In the case of GAs, the chromosome consists of a string of consecutive genes, often bits, which can be mapped into a phenotype. In this case, the mutation means inverting one or few bits. On the other side, it is also possible to operate on integers, floating point numbers and other entities by means of more advanced mutation operators. The crossover operator has several variants of exchanging parts of chromosomes.

In the context of this work, GA has been utilized in two scenarios connected with the traffic simulators. The first one is in the sensitivity analysis for the proper selection of optimized parameters [35, 34, 51]. Here, GA replaces standard techniques of probabilistic sensitivity analysis (e.g. *Monte Carlo*) with quite good results. With the proper optimization parameters chosen, the whole simulation model can be optimized and thus deployed significantly faster (compared to the „manual tuning process“) [24].

Other usage of GA is for optimization of all model parameters (or at least those ones chosen on the basis of the sensitivity analysis). However, the authors commonly use only speed-flow-density relations in the traffic fundamental diagram instead of the real microscopic data. For example, in [31], the authors proposed a generic traffic microsimulation parameter optimization tool that uses genetic algorithms. Their framework was included into *PARAMICS* traffic simulation tool and utilized in Port Area network in downtown Toronto, Canada for estimating speed-flow relations. Another example of calibration of microscopic traffic simulation models using GA has been presented in [53]. The authors estimate the origin-destination (O-D) flows jointly with the behavioral parameters in *MITSIMLab* model. Even in more recent research (2016), the microsimulation models are still calibrated to macroscopic data (e.g. *AIMSUM* [14]). Moreover, all mentioned research is focused especially on one type of facilities – the freeways.

2.4 Challenges Addressed in the Thesis

The research presented in this thesis extends the previous research in several ways. Instead of the macroscopic data we also employ the microscopic data in order to perform the calibration. Microscopic data (e.g. travel times) can be compared one-to-one (e.g. vehicle-by-vehicle or driver-by-driver) and thus the results can easily be compared to other calibrated simulators. The simulation results based on the macroscopic data have been previously compared by means of their plots in time. For those situations where the macroscopic data are still desired (e.g. due to planning purposes where the behavior of elementary entities is not so important), we propose a methodology for a precise mathematical comparison of given results. Furthermore, the thesis primarily deals with the simulation of different facilities, i.e. not only freeways.

Chapter 3

Research Summary

This chapter summarises the research conducted in the thesis. First, a brief overview of the research process is given. Then motivation and abstracts for each included paper are presented. Finally, the all other relevant publications produced during the research are listed.

3.1 Overview

The research presented in this thesis started with a detailed analysis of all traffic phenomena that must be reproduced in any realistic traffic simulation model. Then a suitable CA-based microsimulation model has been chosen and analysed in a greater detail in Paper I. Next, our focus was shifted to the model acceleration to support the multiple in real time simulations. Some explorations were undertaken in Paper I and a more scalable solution was presented in Paper II. As a suitable scalable solution has been found, the work was then directed to model validation and calibration. The first results were published in Paper III for the macroscopic data and an evolutionary calibration has been utilized for the first time. At this stage, the focus was also shifted towards evaluation of the competitiveness of the proposed system. Paper IV included a basic comparison with the current state-of-the-art systems. In Paper V, improved calibration process was introduced including the data used to address real sensors behavior.

3.2 Papers

This section presents relevant details, motivation and contributions for each paper together with the paper abstract.

3.2.1 Paper I

A Scalable Cellular Automata Based Microscopic Traffic Simulation

As the traffic modeling for the real-world applications was one of the main goals of the research, a deep analysis of all traffic phenomena was desirable. Such analysis is covered within this paper. Moreover, this paper focuses on selection of an appropriate model that should be used in the next research. As none of existing models has been perfectly suitable for precise and very fast simulations, a new model (based on the cellular automaton) has been developed. The corresponding simulator has been validated and tested with the

standard symmetric multiprocessor (SMP) architecture.
This work resulted in the basic implementation of the simulator based on the brand new microsimulation model that is suitable for an acceleration by means of common parallel architectures.

Abstract

This paper presents a new model for simulations of very large scale traffic networks. The proposed model is based on microscopic cellular automata (CA) extended to eliminate unwanted properties of ordinary CA based models, such as stopping from maximum speed to zero in one time step. The accuracy of the model has been validated by comparisons with various fundamental diagrams. A parallel implementation developed using the proposed model allows for an almost linear speedup. This allows to run a simulation multiple in real-time, that the traffic state of very large scale networks can be precisely predicted, for example, with various scenarios.

3.2.2 Paper II

Cellular Automata Based Traffic Simulation Accelerated on GPU

The main motivation for the research covered in this paper is in the need for even faster multiple in real time simulations for the real ITS deployment. A faster simulation will enable us to perform and evaluate more different strategies and scenarios. As modern graphic processing units (GPUs) began to contain more and more powerful (SMP) processing cores, the paper was focused on their utilization in our simulator.
This work resulted in a scalable and effective solution for previously proposed simulator which can now be accelerated on GPUs. Also some architecture depending limitations have been investigated.

Abstract

Intelligent transportation systems become more and more important with the increasing traffic densities and safety requirements. A reasonably good traffic prediction can be obtained using microscopic traffic simulation models witch distinguish and trace every traffic entity. However, microscopic simulation requires considerable computing resources. In this paper, we propose to accelerate a cellular automata based microscopic traffic simulator using graphic processing units (GPU). The proposed accelerator provides speed-up of 204.65 with respect to a single core solution for problems instances containing 170 mil. cells, equivalent to 935 000 km of traffic network. This solution is sufficient to predict traffic simulations multiple in real-time.

3.2.3 Paper III

Evolutionary Approach to Calibration of Cellular Automaton Based Traffic Simulation Model

The proposed scalable solution presented in the previous papers has to be calibrated and validated before the real usage. However, due to missing real data in our early research, suitable mathematical models such as Van Aerde model (operating with the macroscopic data) are utilized for the calibration. For the first time, we have utilized a evolutionary

algorithm in the calibration process in the way that traffic expert is not forced to set any of the model parameters.
This work resulted in a brand new methodology for the comparison of simulation results based on macroscopic variables. A new approach for the calibration of the model has been proposed.

Abstract

Microscopic traffic simulation models have become very popular in the evaluation of transportation engineering and planning practices in the past few decades. To achieve high fidelity and credibility of simulations, a model calibration and validation must be performed prior to deployment of the simulator. In this paper, we proposed an effective calibration method of the microscopic traffic simulation model. The model is based on the cellular automaton, which allows fast large-scale real-time simulation. For its calibration, we utilized a genetic algorithm which is able to optimize different parameters much better that a human expert. Furthermore, it is possible to readjust the model to given field data coming from standard surveillance technologies such as loop detectors in our case. We have shown that the precision of simulations can be increased by 20 % with respect to a manually tuned model.

3.2.4 Paper IV

Calibration of Traffic Simulation Models Using Vehicle Travel Times

In our previous work we have used the data on the macroscopic level of detail. As some types of truly microscopic data are becoming more available and we are able to get the microscopic data from our partners, we could use them in our research. The whole calibration process was updated in order to exploit these microscopic data.
As a result of this work, we were able to use our calibration process to find out a very competitive solution that is stable for different microscopic data sets. Moreover, it was shown that our automated process of calibration is able to find parameter values that are hard to determine or even estimate with the other common techniques.

Abstract

In this paper, we propose an effective calibration method of the cellular automaton based microscopic traffic simulation model. We have shown that by utilizing a genetic algorithm it is possible to optimize various model parameters much better than a human expert. Quality of the new model has been shown in task of travel time estimation. We increased precision by more than 25 Moreover, we were able to calibrate some model parameters such as driver sensitivity that are extremely difficult to calibrate as relevant data can not be measured using standard monitoring technologies.

3.2.5 Paper V

Advanced Approach to Calibration of Traffic Microsimulation Using Travel Times

The motivation for the final paper was to deal with real properties of traffic sensors, e.g. when some data are often missing in data sets. Moreover, we begin to use the raw data

from the sensors instead of their statistical properties.
This research and deployment resulted in a complete system for calibration of microscopic traffic simulators based on cellular automata which is not only scalable and precise, but also competitive with other similar systems. Moreover, the proposed system can cope with incomplete data sets, and thus it can be adapted to the real world situations.

Abstract

An effective calibration method of the cellular automaton based traffic microsimulation model is proposed in this paper. It is shown that by utilizing a genetic algorithm it is possible to calibrate different parameters of the model much better than a traffic expert. Moreover, using this process it is also possible to find several model parameters that are extremely difficult to calibrate as relevant data can not be measured using standard monitoring technologies or complete data sets are often not available. The quality of the new calibrated models is discussed in the task of vehicle travel time estimation. The precision of simulations is increased over three times compared to a manually tuned model. The average error rate is 10.75.

3.3 List of Publications

Papers Included in Thesis

I KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. A Scalable Cellular Automata Based Microscopic Traffic Simulation. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2011 (IV11).* Baden-Baden: IEEE Intelligent Transportation Systems Society, 2011, pp. 13-18. ISBN 978-1-4577-0889-3. [70 %]

II KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. Cellular Automata Based Traffic Simulation Accelerated on GPU. In *Proceedings of the 17th International Conference on Soft Computing (MENDEL2011).* Brno: Institute of Automation and Computer Science FME BUT, 2011, pp. 395-402. ISBN 978-80-214-4302-0. [70 %]

III KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. Evolutionary Approach to Calibration of Cellular Automaton Based Traffic Simulation Model. In *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC2012).* Anchorage: IEEE Intelligent Transportation Systems Society, 2012, pp. 122-129. ISBN 978-1-4673-3062-6. [70 %]

IV KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. Calibrating Traffic Simulation Model Using Vehicle Travel Times. In *Proceedings of the 10th International Conference on Cellular Automata for Research and Industry (ACRI 2012).* Berlin: Springer-Verlag, 2012, vol. 7495, pp. 807-816. ISBN 978-3-642-33350-7. [70 %]

V KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. Advanced Approach to Calibration of Traffic Microsimulation Models Using Travel Times. In *Journal of Cellular Automata (JCA).* Philadelphia: Old City Publishing, Inc., 2013, vol. 8, no. 6, pp. 457-467. ISSN 557-5969. [70 %]

Other Relevant Papers

- KORČEK Pavol, SEKANINA Lukáš a FUČÍK Otto. Towards Scalable and Accurate Microscopic Traffic Simulation Using Advanced Cellular Automata Based Models. In: *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems Workshops.* Madeira Island: IEEE Intelligent Transportation Systems Society, 2010, s. 27-35. ISBN 978-972-8822-20-0. [70 %]

- KORČEK Pavol, SEKANINA Lukáš a FUČÍK Otto. Microscopic Traffic Simulation Using CUDA. In *Advanced Computer Architecture and Compilation for High-Performace and Embedded Systems (ACACES 2011) Poster Abstracts.* Fiuggi: Academia Press, 2011, s. 207-210. ISBN 978-90-382-1798-7. [70 %]

- PETRLÍK Jiří, KORČEK Pavol, FUČÍK Otto, BESZÉDEŠ Marián a SEKANINA Lukáš. Estimation of Missing Values in Traffic Density Maps. In *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC2012).* Anchorage: IEEE Intelligent Transportation Systems Society, 2012, s. 632-637. ISBN 978-1-4673-3062-6. [15 %]

Chapter 4

Discussion and Conclusions

This chapter discusses the approach followed in the thesis. In addition, the achieved results are summarised and discussed. Finally, conclusions and possible directions for future work are presented.

4.1 The Approach

An analysis of recent scientific and commercial traffic simulation tools and a study of the current acceleration, calibration and validation techniques have been performed within this research. Some of the studied tools are reported in the introduction to this thesis. In order to validate an optimize the proposed methods, various contacts were accordingly established with research and public institutions which enabled to acquire real data from traffic sensors at both macroscopic and microscopic level of detail. The data were obtained from universities (University of Žilina – The Faculty of Operation and Economics of Transport and Communications, Czech Technical University in Prague - The Faculty of Transportation Sciences and University of Pardubice - The Jan Perner Transport Faculty) and commercial companies operating in the transport sector (Camea, spol. s r.o., Road and Motorway Directorate of the Czech Republic and Technical management of roads in Prague), but also from several publicly available international sources such as the Research Data Exchange (RDE) database maintained by the U.S. Department of Transportation, Federal Highway Administration[1]. All concrete research steps are presented in description of published scientific papers mentioned earlier in this thesis (see Chapter 3).

4.2 Software Outcomes

During the research, several software tools were developed and actively used. When calibrating a simulator, process can fail when not all proper data are available, for example, when a road segment is not covered by sensors. In order to calibrate for macroscopic data (e.g. traffic flow), software *TRAffic SEnsor LOcation* (TRASELO)[2] capable of tracking such missing data was developed. On the basis of a given traffic network topology (where the nodes represent intersections and the edges are roads, see Fig. 4.1), with several traffic sensors embedded, this program calculates and generates the remaining values. Moreover, *TRASELO* suggests the sites where new sensors should be installed in order to enable the

[1]Available at: www.its-rde.net
[2]Available at: www.fit.vutbr.cz/research/view_product.php.en?id=297

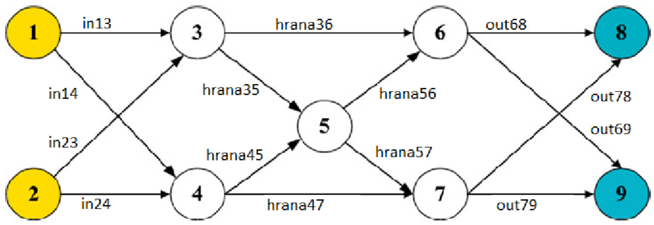

Figure 4.1: TRASELO network example.

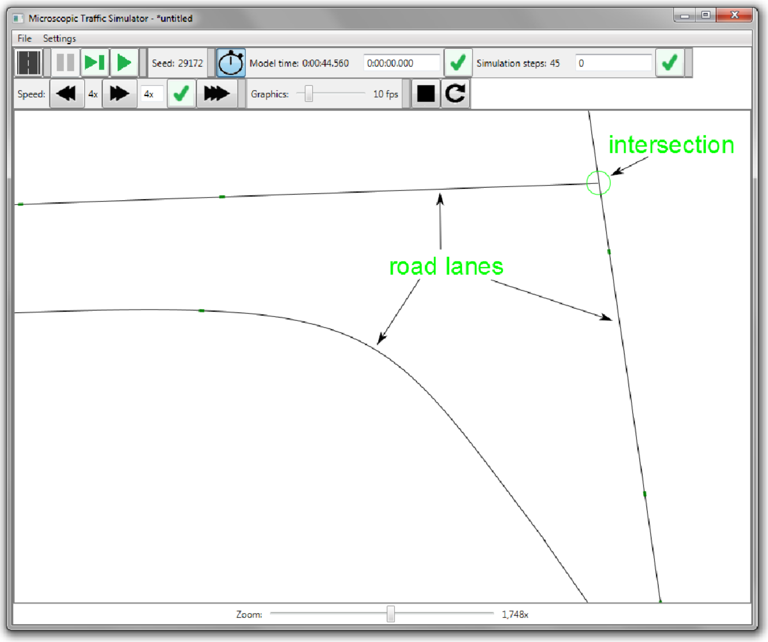

Figure 4.2: Simulation of a given topology in the microsimulation.

calculation of the remaining and greatest possible number of missing values. The method used in this program is based on the linear algebra (matrix operations) and it is inspired by paper [41].

In addition, a simple road traffic microsimulator[3] with a graphical user interface was developed. It is based on the principle of cellular automaton, as presented in this work, and intended for an experimental evaluation of certain traffic phenomena produced with the aid of simulation. Its user interface (see Figure 4.2) enables the road topology to be constructed with several lanes and intersections. This software supports modeling of the unmarked intersections, major-minor road intersections and intersections with traffic lights, and can even be used for roundabouts. The macroscopic data are generated and graphically displayed during the simulation, while the microscopic data can be exported and imported using the most recent version of the tool. Changing the local transition function and possibly other parameters (see Figure 4.3) allows the model to be adapted to the results of the calibration process.

[3]Available at: http://www.fit.vutbr.cz/research/view_product.php.en?id=314

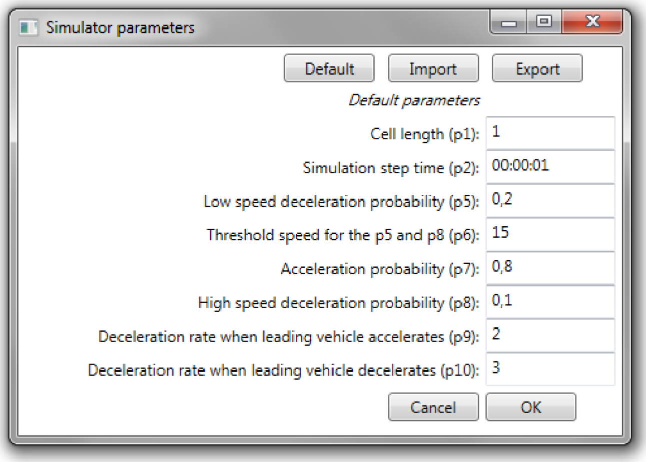

Figure 4.3: Setting the parameters of the microsimulation.

4.2.1 Scalability

The problem of scalability seen in the microsimulators involves not just the modeled topology dimensions, but also the number of simulated entities (vehicles, etc.). The demand for computing performance is growing as the traffic density increases. For a fair comparison of our model under different conditions, *TRANSIMS*[4] microsimulator was selected. *TRANSIMS* is more powerful than the commercially used *VISSIM*[5] in a simple parallelization by means of the job distribution among 16 computers [39], something which today can be replaced with a single multicore processor. Due to this, as well as due to its unavailability (without a graphical extension), the *VISSIM* was not chosen for the comparison. Instead, *MATSIM*[6] - a multi-agent traffic simulation tool was selected, with its latest performance modifications [56]. Finally, a command-line version of *SUMO*[7] microsimulation tool was also incorporated into our comparison. The choice was made with respect to the availability and practical use of these tools. Moreover, an assessment by the scientific community was taken into account as well [19].

The whole simulation took place in a synthetic testing environment created with mentioned tools under the same traffic conditions. Namely, a 30-kilometer-long road network and a random generation of incoming vehicles with the same vehicle arrival probability distribution at different filling densities were considered. An average runtime out of ten runs for one-hour modeled traffic situation with 10% traffic density was taken as the reference. The runtime for every selected microsimulator was also calculated in the same manner for other densities (20%, 30% and 40%). Table 4.1 shows the time multiple necessary to simulate the same segment compared to the reference time (with 10% traffic density). The simulations were carried out on a machine with two processors Intel® Xeon® E5-2650 v3 (25M Cache, 2.30 GHz, 10 cores and 20 threads per processor) with 32 GB of RAM in total. There was no specialized acceleration routine implemented and the models used were not calibrated, except using the same initial settings (the vehicle/driver type and parameters).

[4] Available at: https://sourceforge.net/projects/transims/
[5] Available at: http://vision-traffic.ptvgroup.com/en-uk/home/
[6] Available at: http://www.matsim.org/
[7] Available at: http://sumo.dlr.de/wiki/Main_Page

	Our model	**TRANSIMS**	**MATSIM**	**SUMO**
10 % density	*1.00*	*1.00*	*1.00*	*1.00*
20 % density	*2.02*	*2.01*	*3.74*	*3.32*
30 % density	*3.00*	*3.02*	*4.58*	*6.98*
40 % density	*4.01*	*4.00*	*5.31*	*12.33*

Table 4.1: Scalability comparison of selected tools: the time multiple necessary to simulate the same segment compared to the reference time (with 10% traffic density).

Manually tuned	**Set 1**	**Set 2**	**Set 3**
23.44	*13.96*	*17.80*	*14.80*

Table 4.2: An average error in % on macroscopic datasets (from the paper included in Chapter 3.2.3).

As can be seen from Table 4.1, the simulator based on our model is very similar in terms of performance to *TRANSIMS*. For higher densities, both tools require a proportional increase in the simulation time necessary to finish, which was not the case in the remaining simulators. The reason is that both simulators are based on the same principle of a well-scalable cellular automaton. However, the *TRANSIMS* publicly available sources indicate that it has not been modified or tested to run effectively on other acceleration platforms such as graphics cards (GPGPU). This was also confirmed in an opinion provided by an author whose paper in 2013 [52] referenced our paper discussed in Chapter 3.2.1. The author points out that the GPGPU parallelization approach seems to be extremely suitable because GPGPUs are constantly elevating their power. It should also be noted that both tools are capable of performing the simulations several times faster than in real time, which is not true for the other chosen microsimulation tools.

4.2.2 Accuracy and Robustness

In the papers included into this thesis, the quality of our model and the simulator based thereon was compared on different data sets. Results on test data sets, both macroscopic and microscopic, have been already published. For a basic comparison, the accuracy of the initial and manually tuned models has always been presented. These results are briefly summarized in Table 4.2, Table 4.3 and Table 4.4.

A comparison with the other existing models and microsimulation tools is not straightforward because every author shows the quality on its own data. Such data are not often publicly available due to the exclusive ownership of the providers. If the relevant model was available then it would be possible to compare these models with either our data or publicly available data as described earlier.

Manually tuned	**Set 1**	**Set 2**
32.3	*19.23*	*15.21*

Table 4.3: An average error in % on macroscopic datasets derived from the microscopic data (from the paper included in Chapter 3.2.4).

Manually tuned	Set 1	Set 2	Set 3	Set 4
34.31	*11.19*	*9.89*	*11.27*	*10.64*

Table 4.4: An average error in % on pure microscopic data sets (from the paper included in Chapter 3.2.5).

	Set 1	Set 2	Set 3	Set 4
Our model	*11.19*	*9.89*	*11.27*	*10.64*
TRANSIMS	*15.13*	*12.19*	*18.45*	*21.25*
MATSIM	*85.25*	*89.31*	*91.18*	*89.67*
SUMO	*19.88*	*13.76*	*21.39*	*23.52*

Table 4.5: An average error in % on pure microscopic data sets (according to the paper included in Chapter 3.2.5).

For such a comparison, the tools mentioned in Chapter 4.2.1 were chosen and calibrated according to the procedures set out specifically for each tool [44, 8]. Only the *MATSIM* tool was used „as-is“ because there was no calibration method appropriate for the selected data, namely the microscopic data prepared in our previous research. In this data, some values are also missing as described in the paper included in Chapter 3.2.5. The results are summarized in Table 4.5.

It is evident that our calibration procedure is able to calibrate the proposed microsimulation model significantly better than other tools. The method scores better in terms of several percent points on used data. Our calibrated model has also been employed as a reference model for other authors [59].

An evolutionary calibration method for traffic microsimulators appears to be robust in several aspects. First, there is no need for a sensitivity analysis, which is quite complicated in models with many parameters, as has been demonstrated in [15]. It is due to the basic principle of the evolutionary approach which chooses the parameter values by its own. And even for different traffic facilities it is able to optimize different model parameters. This is also confirmed in a recent study published in 2015 [37] and related work, which among other things analyzed the possibility of employing several calibration techniques for the purpose of calibration of traffic simulators. The authors included our work for the reference. Furthermore, the method is robust regarding the missing data (see data set 3 and 4 in Table 4.5) because in such a case an appropriate setting of the microsimulation model parameters capable of precise simulations can be found. In this case, the other simulators are not precise enough.

4.3 Conclusions

The main contributions of this thesis can be summarized as follows:

- Development of a brand new traffic microsimulation model that is scalable and thus applicable for a) large-scale areas, b) high density areas and c) high-speed multiple in real time scenarios.

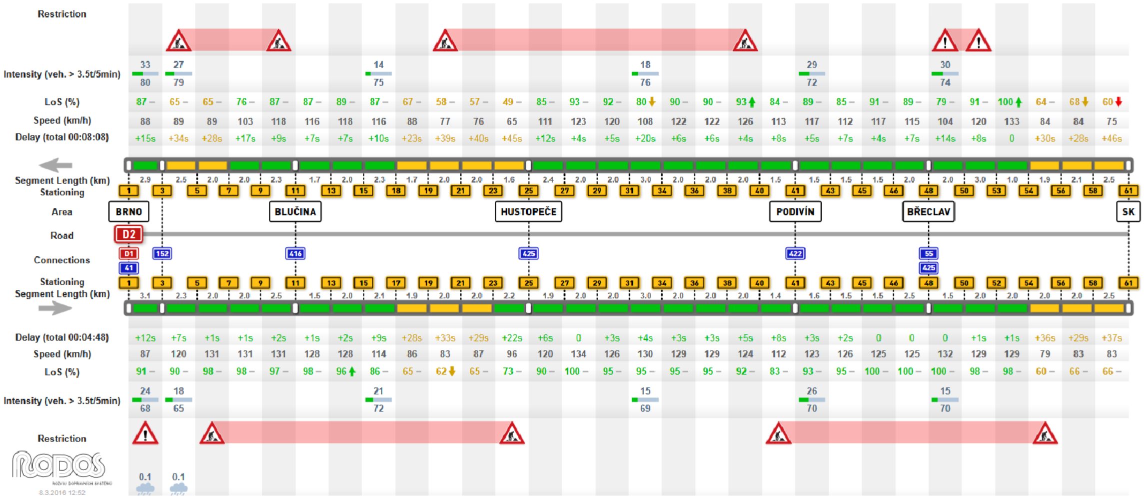

Figure 4.4: RODOS online system preview. Source: http://rodos.vsb.cz.

- Design of a brand new calibration procedure that allows to perform highly precise simulations a) without the need for sensitivity analysis, b) capable of utilizing both micro- and macroscopic data and c) capable of adopting to real-world sensors conditions.

It can be concluded that the thesis has contributed to the initial goal of developing a fast, scalable and accurate or highly precise traffic simulator, capable of multi-real-time simulations of complex traffic areas.

4.4 Future Work

Traffic engineers and other specialists are continuously improving ITS. There is a growing interest in deployment of the microsimulation tools in the traffic states forecasting as stated at the beginning of this thesis. It also happens in the Czech Republic and, therefore, the *RODOS* centre has been established in 2012. It is currently the biggest active subject in the field of applied transport research with the main focus on road transport monitoring and control as well as its funding. It consists of three biggest technical universities, one public research institution and six commercial companies that rank among leading players in the fields of IT, software, data collection and practical implementation of ITS on the Czech market. The strategic objective of the Centre is to create a complex information „superstructure“ for road transport by means of new transport informatics tools and to integrate it into existing ITS.

RODOS is developing *Dynamic Mobility Model of the Czech Republic* (DMM). It integrates dynamic models of passenger, vehicle and goods mobility and related information in the entire area of the Czech Republic. The model and its operation will be utilised extensively not just in transport and other network industries but also in processes of the state and public administration [43].

RODOS DMM uses databases and several predictive features build upon them. These databases consist of several data streams coming from various sensors and so the traffic data needed for the proper microsimulation are also present. The scalable microsimulation model with a highly precise calibration method created in this work will be integrated to this system. It can be deployed to allow performing precise short and long term traffic states forecasting when calibrated for various facilities. It will be a big opportunity to try our model in simulating unexpected traffic events (such as accidents and other types of collisions). This will need further experiments with the whole system and it will be definitely interesting and challenging continuation of the work started in this thesis.

Bibliography

[1] M. A. Abdel-Aty and A. E. Radwan. Modeling traffic accident occurrence and involvement. *Accident Analysis and Prevention*, 32(5):633 – 642, 2000.

[2] V. Alexiadis, J. Sloboden, G. Cordahi, and R. VanGorde. Guidance on the Level of Effort Required to Conduct Traffic Analysis Using Microsimulation. [online] http://www.fhwa.dot.gov/publications/research/operations/13026/13026.pdf, 2014. 96 pages.

[3] S. Algers, E. Bernauer, M. Boero, L. Breheret, C. di Taranto, K. Dougherty, M.and Fox, and J Gabard. Simulation Modeling Applied to Road Transport European Scheme Tests (SMARTEST) - Review of Micro-Simulation Models, March 1998.

[4] Ashley, F., et al. Tackling congestion by making better use of England's motorways and trunk roads. Technical report, National Audit Office, London, UK, 2004. 50 pages.

[5] J. Barcelo. *Fundamentals of Traffic Simulation.* Springer, San Francisco CA, 2010.

[6] R. Barlovic, J. Esser, K. Froese, W. Knospe, L. Neubert, M. Schreckenberg, and J. Wahle. Online traffic simulation with cellular automata. *Traffic and Mobility*, pages 117–134, 1999.

[7] M. Barth and K. Boriboonsomsin. Real-World Carbon Dioxide Impacts of Traffic Congestion. *Transportation Research Record: Journal of the Transportation Research Board*, 2058:163 – 171, 2008.

[8] M. Behrisch and D. Krajzewicz. SUMO Tutorials/Calibration. [online] http://sumo.dlr.de/wiki/Main_Page.

[9] P. J. Bentley. *Evolutionary Design by Computers.* Morgan Kaufmann, San Francisco CA, 1999.

[10] S. A. Boxill and L. Yu. An Evaluation of Traffic Simulation Models for Supporting ITS Development. Technical report, Texas, 2000. 116 pages. Technical report, Number 167602-1.

[11] J. Brügmann, M. Schreckenberg, and W. Luther. Real-Time Traffic Information System Using Microscopic Traffic Simulation. In *Modelling and Simulation (EUROSIM), 2013 8th EUROSIM Congress on*, pages 448 – 453, 2013.

[12] P. Byungkyu, B. L. Smith, J. Lee, D. Pampati, M Ben-Akiva, and R. Balakrishnan. Evaluation of DynaMIT–a Prototype Traffic Estimation and Prediction System, 2007. Report No. UVACTS-15-11-71.

[13] D. Charypar, A. Horni, and K. W. Axhausen. A Parallel Framework for Large Spatial Microsimulations. [online] http://www.fhwa.dot.gov/publications/research/operations/13026/13026.pdf, 2010. Report.

[14] S. Chiappone, O. Giuffre, A. Grana, R. Mauro, and A. Sferlazza. Traffic simulation models calibration using speed-density relationship: An automated procedure based on genetic algorithm. *Expert Systems with Applications*, 44:147–155, 2016.

[15] B. Ciuffo and L. C. Azevedo. A Sensitivity-Analysis-Based Approach for the Calibration of Traffic Simulation Models. *IEEE Transactions on Intelligent Transportation Systems*, 15(3):1298 – 1309, 2014.

[16] A. Clarridgea and K. Salomaab. Analysis of a cellular automaton model for car traffic with a slow-to-stop rule. *Theoretical Computer Science*, 411(38-39):3507 – 3515, 2010. Implementation and Application of Automata (CIAA 2009), ISSN: 0304-3975.

[17] R. Dowling, A. Skabardonis, and V. Alexiadis. Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Software. [online] http://ops.fhwa.dot.gov/trafficanalysistools/tat_vol3/, 2004. 146 pages.

[18] R. Dowling, A. Skabardonis, J. Halkias, G. McHale, and G. Zammit. Guidelines for Calibration of Microsimulation Models: Framework and Applications. *Transportation Research Record: Journal of the Transportation Research Board*, 1876:1 – 9, 2004.

[19] Allan. D. F. and A. M. Farid. A Benchmark Analysis of Open Source Transportation-Electrification Simulation Toolst. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 1202 – 1208, 2015.

[20] K. Fox. New tools to evaluate Intelligent Transportation Systems - the SMARTEST project, 2006. Report.

[21] N. Gartner, C. J. Messer, and A. K Rathi. *Traffic Flow Theory : A state-of-the-Art Report*. 2001. 385 pages, Revised.

[22] P. E. Grubb, B. Hellinga, and J. J Mallett. Using micro-simulation tools to improve transportation decisions: case studies. *Presented at the AQTR-ITE Conference*, 1999. 16 pages.

[23] J. H. Holland. *Adaptation in Natural and Artificial Systems*. A Bradford Book, Reprint edition, 1992.

[24] Y. Hollander and R. Liu. The principles of calibrating traffic microsimulation models. *Transportation*, 35(3):347–362, 2008.

[25] S. P. Hoogendoorn and P. H. L. Bovy. State-of-the-art of Vehicular Traffic Tlow Modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4):283 – 303, 2001.

[26] O. Kastlová and M. Brich. Transport Yearbook Czech Republic. [online] www.sydos.cz/cs/rocenka_pdf/Rocenka_dopravy_2014.pdf, 2014. 173 pages, ISSN: 1801-3090.

[27] M. Killat, F. Schmidt-Eisenlohr, H. Hartenstein, Ch. Rossel, P. Vortisch, S. Assenmacher, and F. Busch. Enabling Efficient and Accurate Large-scale Simulations of VANETs for Vehicular Traffic Management. In *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks*, pages 29 – 38, New York, NY, USA, 2007. ACM.

[28] S. Kollamthodi, D. Kay, I. Skinner, C. Dun, and S. Hausberger. The potential for mass reduction of passenger cars and light commercial vehicles in relation to future CO2 regulatory requirements. Technical Report 3, European Commission, Harwell, Oxford, UK, 2015.

[29] A. Kuba and J. Hausman. *Malé dějiny auta*. Albatros, Praha, 1 edition, 1973. 229 pages.

[30] J. Lüttmerdingová, I. Hamanová, and M. Pálinkásová. Yearbook of Transport, Posts adn telecommunications in 2014. [online] http://www7.statistics.sk, 2014. 131 pages, ISBN: 978-80-8121-328-1.

[31] T. Ma and B. Abdulhai. Genetic Algorithm-Based Optimization Approach and Generic Tool for Calibrating Traffic Microscopic Simulation Parameters. *Transportation Research Record: Journal of the Transportation Research Board*, 1800:6–15, 2002.

[32] S. Maerivoet and B. De Moor. Cellular automata models of road traffic. *Physics Reports*, 419(1):1 – 64, 2005.

[33] S. Maerivoet and B. De Moor. Traffic Flow Theory. *ArXiv Physics e-prints*, July 2005.

[34] T. Mathew and P. Radhakrishnan. Calibration of Microsimulation Models for Nonlane-Based Heterogeneous Traffic at Signalized Intersection. *Journal of Urban Planning and Development*, 136:59 – 66, 2010.

[35] S. Menneni, C. Sun, and P. Vortisch. Microsimulation Calibration Using Speed-Flow Relationships . *Transportation Research Record: Journal of the Transportation Research Board*, 2088:1 – 9, 2008.

[36] U. Mori, A. Mendiburu, M. Álvarez, and A. J. Lozano. A review of travel time estimation and forecasting for Advanced Traveller Information Systems. *Transportmetrica A: Transport Science*, 11(2):119 – 157, 2015.

[37] U. Mori, A. Mendiburu, M. Álvarez, and A. J. Lozano. Robust calibration of macroscopic traffic simulation models using stochastic collocation. *Transportation Research Part C: Emerging Technologies*, 59:358 – 374, 2015. Special Issue on International Symposium on Transportation and Traffic Theory.

[38] M. R. Morris, N. J. Pedersen, and R. E Skinner. *Highway Capacity Manual*. Transportation Research Board, Washington, D.C., 4 edition, 2010. 1650 pages, ISBN: 978-0-309-16077-3.

[39] K. Nagel and M. Rickert. Parallel implementation of the {TRANSIMS} micro-simulation. *Parallel Computing*, 27(12):1611 – 1639, 2001. Applications of parallel computing in transportation.

[40] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *J. Phys. I France*, 2(12):2221 – 2229, 1992.

[41] ManWo Ng. Synergistic sensor location for link flow inference without path enumeration: A node-based approach. *Transportation Research Part B: Methodological*, 46(6):781 – 788, 2012.

[42] A. Ni, Z. Juan, and J. Yang. *Method and Strategy for Parallelizing Microscopic Traffic Simulation*, chapter 174, pages 1600 – 1610. 2010.

[43] Team of authors. Rodos transport systems development centre. march 2016. Online: http://www.rodos-it4i.cz/.

[44] B. Park, B. Brian, and J. Kwak. Calibration and validation of TRANSIMS microsimulator for an urban arterial network. *KSCE Journal of Civil Engineering*, 15(6):1091 – 1100, 2011.

[45] B. Park and H. Qi. Development and Evaluation of a Procedure for the Calibration of Simulation Models. *Transportation Research Record: Journal of the Transportation Research Board*, 1934:208 – 217, 2005.

[46] H. Rakha and B. Crowther. Comparison of Greenshields, Pipes, and Van Aerde Car-Following and Traffic Stream Models. *Transportation Research Record: Journal of the Transportation Research Board*, 1802:248 – 262, 2002.

[47] N. T. Ratrout and S. M. Rahman. A comparative analysis of currently used microscopic and macroscopic traffic simulation software. *The Arabian Journal for Science and Engineering*, 34(1):121 – 133, 2009.

[48] A. Schadschneider. Cellular automata models of highway traffic. *Physica A: Statistical Mechanics and its Applications*, 372(1):142 – 150, 2006. ISSN: 0378-4371.

[49] J. L. Schiff. *Cellular Automata: A Discrete View of the World (Wiley Series in Discrete Mathematics & Optimization)*. 2008.

[50] L. Sekanina. *Evolvable Components - From Theory to Hardware Implementations.* Springer Verlag, Berlin, 2003.

[51] S.M.P Siddharth and G. Gitakrishnan Ramadurai. Calibration of {VISSIM} for Indian Heterogeneous Traffic Conditions . *Procedia - Social and Behavioral Sciences*, 104:380 – 389, 2013. 2nd Conference of Transportation Research Group of India (2nd CTRG).

[52] A. Siromascenko and I. Lungu. A Massive Multilevel-parallel Microscopic Traffic Simulator with Gridlock Detection and Solving. *Studies in Informatics and Control*, 22(3), 2013.

[53] T. Toledo, M. Ben-Akiva, D. Darda, M. Jha, and H. Koutsopoulos. Calibration of Microscopic Traffic Simulation Models with Aggregate Data. *Transportation Research Record: Journal of the Transportation Research Board*, 1876:10–19, 2004.

[54] M. Tomassini and M. Venzi. *Parallel Problem Solving from Nature — PPSN VII: 7th International Conference Granada, Spain, September 7–11, 2002 Proceedings.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[55] J. L. Tripp, H. S. Mortveit, A. A. Hansson, and M. Gokhale. Metropolitan road traffic simulation on FPGAs. In *Field-Programmable Custom Computing Machines, 2005. FCCM 2005. 13th Annual IEEE Symposium on*, pages 117 – 126. IEEE, 2005.

[56] R. A. Waraich, M. Charypar, D.and Balmer, and K. W. Axhausen. *Computational Approaches for Urban Environments*, chapter Performance Improvements for Large-Scale Traffic Simulation in MATSim, pages 211 – 233. Springer International Publishing, 2015.

[57] M. Zamith, P. R. Célia, E. Clua, E. M. Toledo, and G. V.P. de Magalhaes. A new stochastic cellular automata model for traffic flow simulation with drivers behavior prediction. *Journal of Computational Science*, 9:51 – 56, 2015. Computational Science at the Gates of Nature.

[58] M. Zamith, P. R. Célia, M. Kischinhevsky, E. Clua, D. Brandao, A. Montenegro, and E.B. Lima. A probabilistic cellular automata model for highway traffic simulation. *Procedia Computer Science*, 1(1):337 – 345, 2010. {ICCS} 2010.

[59] Y. Zheng, Y. Zhang, J. Hu, L. Li, and X Pei. Simulating coherent flow with cellular automaton model. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 546 – 551, 2012.

Papers

I KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. A Scalable Cellular Automata Based Microscopic Traffic Simulation. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2011 (IV11).* Baden-Baden: IEEE Intelligent Transportation Systems Society, 2011, pp. 13-18. ISBN 978-1-4577-0889-3.

II KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. Cellular Automata Based Traffic Simulation Accelerated on GPU. In *Proceedings of the 17th International Conference on Soft Computing (MENDEL2011).* Brno: Institute of Automation and Computer Science FME BUT, 2011, pp. 395-402. ISBN 978-80-214-4302-0.

III KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. Evolutionary Approach to Calibration of Cellular Automaton Based Traffic Simulation Model. In *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC2012).* Anchorage: IEEE Intelligent Transportation Systems Society, 2012, pp. 122-129. ISBN 978-1-4673-3062-6.

IV KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. Calibrating Traffic Simulation Model Using Vehicle Travel Times. In *Proceedings of the 10th International Conference on Cellular Automata for Research and Industry (ACRI 2012).* Berlin: Springer-Verlag, 2012, vol. 7495, pp. 807-816. ISBN 978-3-642-33350-7.

V KORČEK Pavol, SEKANINA Lukáš and FUČÍK Otto. Advanced Approach to Calibration of Traffic Microsimulation Models Using Travel Times. In *Journal of Cellular Automata (JCA).* Philadelphia: Old City Publishing, Inc., 2013, vol. 8, no. 6, pp. 457-467. ISSN 1557-5969.

Paper I

A Scalable Cellular Automata Based Microscopic Traffic Simulation.

2011 IEEE Intelligent Vehicles Symposium (IV)
Baden-Baden, Germany, June 5-9, 2011

A Scalable Cellular Automata Based Microscopic Traffic Simulation

Pavol Korček, Lukáš Sekanina and Otto Fučík

Abstract—**This paper presents a new model for simulations of very large scale traffic networks. The proposed model is based on microscopic cellular automata (CA) extended to eliminate unwanted properties of ordinary CA based models, such as stopping from maximum speed to zero in one time step. The accuracy of the model has been validated by comparisons with various fundamental diagrams. A parallel implementation developed using the proposed model allows for an almost linear speedup. This allows to run a simulation multiple in real-time, that the traffic state of very large scale networks can be precisely predicted, for example, with various scenarios.**

I. INTRODUCTION

MICROSCOPIC traffic simulation models, in contrast to macroscopic models, distinguish and trace every single vehicle and driver. The models typically employ characteristics such as vehicle lengths, speeds, accelerations, time and space headways, vehicle and engine capabi-lities, as well as some rudimentary human characteristics that describe driving behavior [1], [2].

In the past, the use of microscopic models has been limited in scope and scale of the problem. The applications were restricted to small networks with a limited number of vehicles. With the advent of fast computers a large number of microscopic simulation models have been developed. For an illustration, the SMARTEST report (1999) identified 58 microscopic simulation models of which 32 were analyzed [7]. Some of them are true microscopic simulators in the sense that they model the behavior of each individual vehicle in the traffic flow. Another and most recently short overview of microscopic traffic models (2004) can be found, for example, in [6].

The need for more precise and realistic traffic models resulted in a complicated comparison and validation schemes which are also integrating field measurements into the model [19], [20]. Field data are used for model initialization, calibration and lastly for validation. These simulation models are so called "online" traffic simulators [20]. An important condition for both online simulators and for traffic state forecasting is to simulate the traffic in networks faster than real time (multiple in real time). Hence the acceleration of microscopic models continues to grow in importance. Cellular automata (CA) based modes have been recognized as quite useful in this area, mainly because they can easily be accelerated on parallel computers or directly in hardware.

The goal of this paper is to present our extended CA based model which has been proposed in order to model road traffic more realistically and also quite efficiently. Results from this type of simulations can be used in various ITS, for example, in driver assistance or collision avoidance systems. Our approach attempts to extend the simulation beyond a few roads and intersections to large areas (e.g. we are going to model and simulate very large networks such as the whole Czech Republic with approximately 55 000 km of road segments [8]. Moreover, the given length of road segments does not distinguish the directions, the number of lanes (e.g. the real number of kilometers is much higher when including these options). For instance, the distance between Prague and Brno city is approximately 200 km, but because there are two lanes in each direction, the total length of road segments is four times greater). In contrast to specialized and expensive accelerators, such as Field Programmable Gate Array (FPGA) based machines [28], we propose to utilize a very reasonable solution based on commodity hardware.

The rest of the paper is organized as follows. CA based models for microscopic traffic simulation are introduced in Section II. In order to compare existing and proposed models, we describe a method that utilizes fundamental diagrams in Section III. In Section IV the proposed model is introduced. Results of experimental evaluation are summarized in Section V. Finally, conclusions and sugges-tions for future work are given in Section VI.

II. CELLULAR AUTOMATA BASED MODELS

Cellular automata based models have become popular in the area of microscopic traffic flow simulations because of their simplicity and suitability for acceleration. The first highly suitable model for acceleration of single-lane freeway traffic was introduced by Nagel and Schreckenberg in 1992 [9]. Because of the model's simplicity, it is possible to perform millions of updates in a second [13], [14]. Thus, it may be used for simulating a high volume of traffic over very large networks. Due to some critics of unrealistic behavior (such as [26]), simple CA based models are often extended. On the other hand, CA models were shown to be able to capture all basic phenomena that occur in traffic flows [10], not only in the field of vehicular traffic flow modeling, but also in other fields such as pedestrian behavior, escape and panic dynamics, etc.

This work was partially supported by the Czech Science Foundation under projects P103/10/1517 and GD102/09/H042, by the research program MSM 0021630528 and the BUT grant FIT-S-11-1.

Pavol Korček, Lukáš Sekanina and Otto Fučík are with the Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2, 612 66 Brno, Czech Republic (email: ikorcek@fit.vutbr.cz, sekanina@fit.vutbr.cz, fucik@fit.vutbr.cz).

978-1-4577-0891-6/11/$26.00 ©2011 IEEE

A. *Basic Model*

Nagel's and Schreckenberg's traffic model [9] was initially defined on a one-dimensional array (Fig. 1) with open or periodic boundary conditions and with every single cell representing a road segment. A local transition function (a rule) defines the new state of a cell on the basis of its current state and the state of neighboring cells. Globally viewed, it describes the movements of vehicles from one cell to another cell in a discrete way.

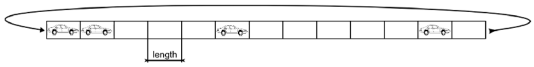

Fig. 1. CA-based traffic network with periodic boundary.

In space domain, each cell represents only a defined length of road segment, so space is coarse-grained. This coarse graininess is fundamentally different from the usual microscopic models, which adopt a semi-continuous space. Some potential lengths are shown in Table 1. The length of 7.5 m is mostly chosen because each car occupies about this amount of space in a complete jam [9]. So this is the average length of a car on the road, including a constant gap in front of and behind each car. In order to model other kinds of vehicles in CA based models, it is better to use another cell length, or even more precisely, use more smaller cells for representing a single vehicle -- especially for bigger types of vehicles [6], [12], [18].

Each CA simulation step represents an amount of time in reality. Based on some previous arguments about driver reaction time, suggestions given in [3], [4] and field data measurements [6], it is possible to represent one simulation step between 0.6 and 1.5 s. In the CA model this value is crucial because together with cell length it gives us the granularity of minimal model speed jumps. Various settings of these variables are shown in Table 1.

Normally, urban traffic road networks are very complex. Paper [15] has shown that arbitrary kinds of road and intersections can be reduced to only a few basic elements. For constructing more complicated traffic networks, we shall simply connect more various kinds of cells to the desired topology. So it is possible to build more traffic lanes [16], [17], roundabounds [18] or very complex topologies for whole cities [19], [20].

B. *Local Transition Function*

Properties of single lane traffic are modeled on the basis of integer valued probabilistic cellular automaton rules [14]. The local transition function can be formalized as follows: each vehicle in a cell has an integer velocity v_v with values between zero and v_{max}. For real vehicle speed v_{act} we use the expression:

$$v_{act} = v_v \cdot speed\_jump \qquad (1)$$

where *speed jumps* are defined in Table 1. We let $\gamma(i)$ denote the cell gap in front of cell *i*. For an arbitrary configuration, one update of the simulation system consists of the following four consecutive steps, which are performed in parallel for all vehicles [9]:

TABLE I
SPEED JUMPS [KM/H] FOR CA MODEL

Cell length [m]	Time step [s]						
	0.5	*0.7*	*0.9*	*1.0*	*1.2*	*1.4*	*1.6*
0.5	3,60	2,57	2,00	1,80	1,50	1,29	1,13
1	7,20	5,14	4,00	3,60	3,00	2,57	2,25
1.5	10,80	7,71	6,00	5,40	4,50	3,86	3,38
2	14,40	10,29	8,00	7,20	6,00	5,14	4,50
2.5	18,00	12,86	10,00	9,00	7,50	6,43	5,63
3	21,60	15,43	12,00	10,80	9,00	7,71	6,75
3.5	25,20	18,00	14,00	12,60	10,50	9,00	7,88
4	28,80	20,57	16,00	14,40	12,00	10,29	9,00
4.5	32,40	23,14	18,00	16,20	13,50	11,57	10,13
5	36,00	25,71	20,00	18,00	15,00	12,86	11,25
5.5	39,60	28,29	22,00	19,80	**16,50**	14,14	12,38
6	43,20	30,86	24,00	21,60	18,00	15,43	13,50
6.5	46,80	33,43	26,00	23,40	19,50	16,71	14,63
7	50,40	36,00	28,00	25,20	21,00	18,00	15,75
7.5	54,00	38,57	30,00	*27,00*	22,50	19,29	16,88
8	57,60	41,14	32,00	28,80	24,00	20,57	18,00

Tab. 1. Single speed jumps used in cellular automaton model in kilometers per hour (km/h) for different cell lengths and various simulation time steps. It is shown that for smaller lengths and greater time steps the model will be more accurate.

1. **Acceleration:** *if* ($v_v < v_{max}$) *and if* ($\gamma(i) > v_v$) *then* $v_v = v_v + 1$.
2. **Slowing down:** *if a vehicle at site i sees the next vehicle at site* $i + j$ ($j \leq v_v$), *it reduces its speed, so* $v_v = j - 1$.
3. **Randomization:** *with probability p, the velocity of each vehicle* (*if* $v_v > 0$) *then* $v_v = v_v - 1$.
4. **Car motion:** *each vehicle is advanced* v_v *sites.*

This simple CA based traffic model shows nontrivial and realistic behavior [9]. Step 3 is essential in simulating traffic flows since the dynamics is completely deterministic, and without this randomness, every initial configuration of vehicles and corresponding velocities reaches a stationary pattern which is shifted backwards. Fig. 5 in Section V also shows that this simple model is capable of reproducing characteristic properties of real traffic, such as certain aspects of flow-density relation, spatio-temporal evolution of jams, stop-and-go waves, etc. [9], [11], [20].

C. *Performance*

In paper [20], an online simulation model of city of Duisburg was introduced. The authors showed that for the most frequently occurring densities the road network of Duisburg can be simulated around 100 times faster than real time. It is necessary to mention that their road network is modeled using only 22 000 cells. Their simulator is not suitable for more cells. An example of explicitly implemented parallelism (using the MPI [29] message passing programming model on IBM SP2 parallel machine) in CA based model was presented in [19]. They have used the standard Nagel's and Schreckenberg's CA model to implement a traffic simulator for city of Geneva and observed a superlinear speedup for 15 000 vehicles. But, as they mentioned, their simulations are not expected to match the exact traffic situation of Geneva (!). CELLSIM is another good example of a recent CA based microscopic simulator [6]. It is tuned to be a high fidelity traffic simulation model, but on the other hand, the field data used for model validation are pretty old. Computational performance of the model, as for other CA based modes, is dependent on the number of vehicles in the system, and it was only showed, that CELSIM is able to simulate around 2 300 vehicles in shorter than real time. As computational power is increasing, we can assume that this number of vehicles may increase too. Recently, some examples of accelerating the standard CA based model were proposed using FPGAs. It was shown that road traffic simulation of entire metropolitan areas can be accelerated with reconfigurable supercomputing that combines 64-bit microprocessors and FPGAs in a high bandwidth and low latency interconnect [28]. For a realistic road description of Portland, which consists of about 6.25 million cells in a CA model, they were able to achieve a speedup of 34.4 independently of the number of vehicles. It is also necessary to mention that their streaming simulation model has used extra and expensive hardware.

III. MODEL VALIDATION

There exists some sophisticated methods for traffic simulator comparison and validation based on mathematical models (created using field measurements of some traffic parameters), or purely in field measurements and model comparison with this measured data [19], [20] and [24]. This section provides a short description of one of the methods – fundamental diagrams.

Road traffic is always in a specific state that is characterized by the flow rate – *f*, the traffic density – *d* and the vehicles mean speed – *v*. One can combine all possible homogeneous and stationary traffic states in an equilibrium function that can be described graphically by three diagrams, so-called fundamental diagrams. This diagram shows the relation between two of the three main macroscopic variables. The third variable can always be recovered by means of the relationship:

$$f = d \cdot v \qquad (2)$$

A fundamental diagram applies to a specific road and is drawn up with basic observations. The stationary and homogenous traffic is always in a state that is located on the line of Fig. 4. It should be noted that Fig. 4 is used in Section V to compare theoretical results with results obtained from our simulation model. But some special state points require extra attention:

a) Completely free flowing traffic

When vehicles are not impeded by other traffic they travel at a maximum speed of v_{max} (also called free speed). This speed is dependent, among other things, on the design speed of a road and the speed restrictions in operation at any particular time and weather. At free speed, flow rate and density will be close to zero.

b) Saturated traffic

On saturated road segments flow and speed go down to zero. The vehicles are queuing and there is a maximum density of d_{max}, which is also called jam density.

c) Capacity traffic

The capacity of a road is equal to the maximum flow rate f_{cap}. The maximum flow also has an associated capacity speed v_{cap} and capacity density d_{cap}. The diagram shows that the capacity speed lies strictly below maximum speed v_{max}.

Mathematical models for fundamental diagrams are represented with parametric mathematical expressions for the equilibrium relations given by entire fundamental diagrams. In 1934 Greenshield (for example, [21]) drew up the first formulation that was based on a small number of measurements. In his diagrams (shown in Fig. 4 with dotted lines), the capacity speed v_{cap} is half the maximum speed v_{max} and capacity density d_{cap} is half the maximum density d_{max}. This formulation is a rough simplification of observed traffic behavior. It was also shown that Greenshield's parabolic speed-flow relationship and its corresponding linear speed-density relationship are inconsistent with the field data [21]. Specifically, the data indicate that vehicles travel at a speed that is higher than half the free-speed when traveling at capacity. Even so, this model is still frequently used because of its simplicity.

Another well-known and also much used formulation of fundamental diagram is the so-called triangular diagram, or according to its author, Pipes' model [5]. His model is multi-regime in the sense that a different model is utilized for the congested versus uncongested regimes. Diagrams (in Fig. 4 sketched with a dashed line) have many advantages in dynamic traffic modeling. In his equilibrium relation the mean speed equals the maximum speed for all traffic states that have densities smaller than the capacity density. The branch of a triangle that links capacity state with saturated state has a negative constant slope.

In order to address the main laws of Greenshield's and Pipes' models into a single-regime model, Van Aerde's model can then be obtained [22], [23]. This model overcomes the shortcomings of Pipes' model in which it is assumed that vehicle speeds are insensitive to traffic density in the un-congested regime, which has been demonstrated to be inconsistent with a variety of field data from different facility types. Alternatively, the model overcomes the main shortcoming of the Greenshields model which assumes that the velocity and flow relationship is parabolic, which again is inconsistent with the field data from a variety of facility types. Corresponding diagrams are sketched with a full line in Fig. 4.

Because all the previously mentioned parameters – flow, density and velocity – are mostly identified for macroscopic models, we need to obtain these properties from the microscopic model. This is simply done for a concrete cell in topology (for flow), or for a concrete number of cells (for obtaining density and mean speed). Details can be found in [25].

IV. Proposed Model

The proposed model was designed with the aim to (i) obtain more accurate traffic behavior from traffic simulations and (ii) accelerate the simulations as much as possible using commonly available devices such as multi-core chips or graphic processing units. The proposed model extends the CA based model presented in Section II.

A. *Parameters of the Cell*

Fig. 2 shows us how the state and parameters of a cell are encoded. Instead of the traditional CA based models implementation, which utilizes several integers (to encode actual speed, speed limit, vehicle type …), we have used only one integer and encoded the parameters at the bit level. For example, for the actual speed (from 0 to 11), we need only 4 bits. This encoding allowed us to speed up the simulation significantly because the memory traffic is highly reduced.

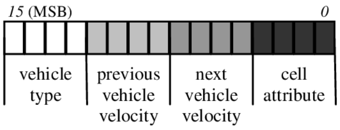

Fig. 2. Single CA cell state encoded at the bit level on 16 bits.

B. *Acceleration and Deceleration Rules*

As in the traditional CA based model, single-regime (i.e. constant acceleration) is used in our model. This gives us a reasonable approximation when the vehicle is accelerating [4].

On the other hand, deceleration has two phases. The first phase is identical with the previously described randomization step (no. 3) described in Section II. The second phase, marked in the CA based model as a slowing down step and used for collision avoidance, is modified to prevent a vehicle stopping from maximum velocity to zero in one simulation step. Elimination of this property is done in our model by the "looking ahead" technique and observing three parameters: *actual vehicle speed, speed of the leader vehicle* and the *size of cell gap* (after calculating the advance of following vehicle). If these values give us enough room for vehicle advancing in the next simulation step, nothing is done. If not, actual vehicle speed is proportionally reduced and its new value is used in the next simulation step. For example, if investigated car has a speed of 5, the cell gap gives us enough room, however, if the following vehicle has only a speed of 1, actual speed will be limited to 3. This causes us to avoid approaching a leader with extremely fast speed. This action is performed only when the follower is not decided to bypass a leading vehicle, so this updated rule (no. 2) only appears if no lane change is assumed on the road. Solely the lane changing logic of our model has been developed using the same car-following features of the model presented in this paper [27]. It was also shown that this is sufficient and fully accurate [6].

C. *Double Buffering*

Network topology is stored in two memory buffers. High-speed simulation is allowed (even for the proposed deceleration technique) using the so-called double buffering, when one simulation step is performed on one copy of buffer and consequently on another buffer copy.

D. *Other Parameters*

The cell length used in our model is 5.5 m, which is a little less than in the original CA model (7.5 m). This constant is based on our own field experiments performed in the Czech Republic [6], where a shorter gap between vehicles is observed and also vehicle sizes are becoming smaller. In order to model other kinds of vehicles, especially bigger ones, we simply use two successive cells, which means doubling the vehicle size to the length of 11 m. By defining the cell length and simulation time, we explicitly specified the minimum speed jumps to 16.5 km/h. For traditional CA based model the speed jumps are 27 km/h, as can be seen in Table 1.

We assume a neighborhood of 11. With a cell length of 5.5 m it gives us a distance of 60.5 m between the leader and the follower. As observed in [3] and [6], for separations greater than 61 m, car-following is negligible on a driver's behavior. Based on the size of the maximum neighborhood, it is also possible to calculate maximum vehicle velocity, which is exactly 181.5 km/h.

Reaction time is equivalent to one simulation time step and in our model it is 1.2 s. This value is very close to 1.21 s – the mean reaction time of unaltered drivers in car-following as suggested in [4].

V. Experimental Results

The proposed model was implemented in the programming language C/C++ and then simulated on a dual processor *Quad Core Xeon(R) CPU E5420 @ 2.50 GHz* machine with a system memory of 16 GB running with a 64bit Linux-based operating system. In order to evaluate the model, we have used urban areas modeled using from 10 thousand to 100 million CA cells. Defining the global behavior of vehicles is necessary to determine the turns at road intersections. Since a traditional origin-destination matrix with a sufficient resolution in time and space is not available, all vehicles are guided randomly through the

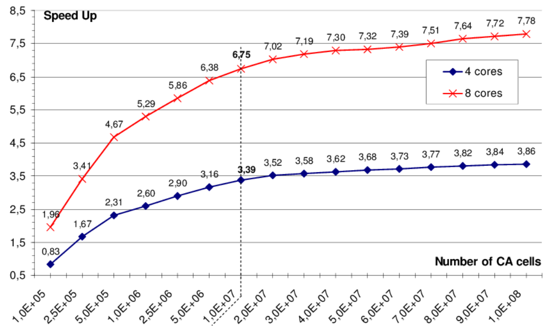

Fig. 3. Speedup obtained for 4 and 8 processor cores. The model is initialized for 50% density of vehicles.

network with respect to the given turning probabilities at road junctions over time. These probabilities were counted based on real field data, mostly from loop and camera based detectors (obtained from our industrial partner). The proposed implementation is able to simulate up $1.5 \cdot 10^6$ cells with 50% vehicle occupancy in a shorter than real time.

A. *Analysis of Scalability*

For more vehicles or simply more CA cells, the execution time has increased more quickly than the simulation time. In order to increase the level of parallelism, the model was extended using *OpenMP API* specification for parallel programming [25]. Fig. 3 shows the speedup obtained for the original model with regard to the extended model. The first run (in Fig. 3 denoted as "4 cores") utilizes only one physical processor (i.e. 4 processor cores) where a single thread was executed per one physical processor core.

The next run (in Fig. 3 denoted as "8 cores") is measured with both processors (i.e. with 8 processor cores). For the simplified model, which requires employing 10 million cells ($5.5 \cdot 10^4$ km of roads [24]), the obtained speedup is 3.39 using 4 cores and 6.75 using 8 cores wrt a single core processor. This is also enough to simulate the model in

TABLE II
SPEED UP FOR VARIOUS NUMBER OF PROCESSOR CORES

Number of cores	*16*	*32*	*64*	*128*	*256*	*512*
Speed up	15.50	31.00	62.00	124.00	248.01	496.03

Tab. 2. Prediction of speed up for various number of processor cores based on the assumption of constant effectiveness of 96.88 % at a maximum number of CA cells (1.0e8) and with 50 % vehicle occupancy.

multiple real time, so an additional "free time" may be used in the future. The speedup is 7.78 (or 3.86) if the number of cells is 10 times higher. However, this speedup is not sufficient for real time simulation of a system consisting of 100 million cells.

If we consider that our implementation utilizes a double buffering technique, we can reach the average effectiveness of 96.88 % for both number of cores (96.5 % for 4 cores and 97.25 % for 8 cores). Hence we may predict the speedup for

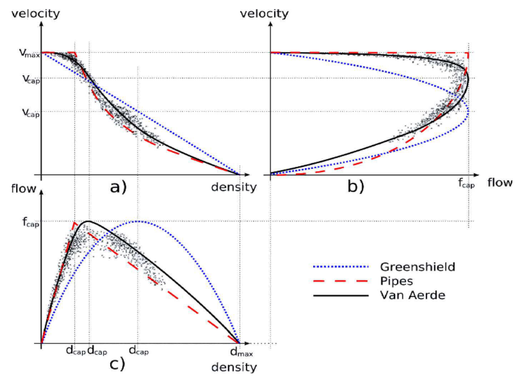

Fig. 4. Fundamental diagrams relations. They show basic relation between the vehicle velocity at one kilometer per hour, the density in the number of vehicles per one kilometer and the flow in the number of vehicles per hour. Dotted plots are measured over 2 100 time steps with the proposed CA based microsimulation model.

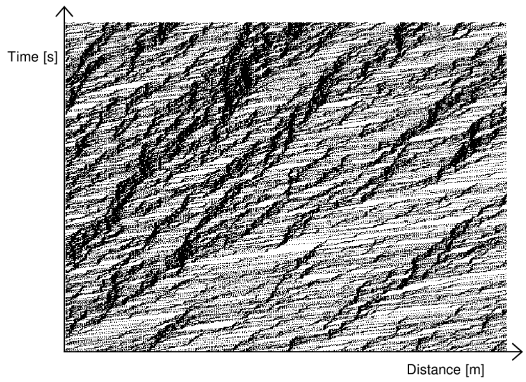

Fig. 5. Time-distance plot of density waves in high density traffic. Each trajectory represents a single vehicle. A 5.5 km long road segment is simulated before road junction for approximately 35 minutes.

more cores. Table 2 shows that if we had 16 processor cores, the speedup would be 15 which is very close to the linear acceleration. Therefore, we can claim that our model is perfectly scalable.

B. *Analysis of Accuracy*

Fig. 4 compares all mentioned theoretical fundamental diagrams with the fundamental diagrams calculated for our model (with dots). We can observe that the data from our model best fit the Van Aerde fundamental diagram. It was shown that this model is very good in reflecting traffic stream behavior for a number of varying facility types, including a freeway or a tunnel roadway [21]. It is also important to note that our model requires a lower level of randomization probability (used for slowing down) due to our deceleration improvements, but this model is still able to reproduce basic traffic phenomena, as is shown in Fig. 5.

VI. CONCLUSION

In this paper, a model for simulations of very large scale traffic networks has been presented. The heart of the simulator is an extended microscopic CA based model, which eliminates unwanted properties of common CA based models, such as stopping from maximum speed to zero in one time step. Our model is tuned to be highly accurate, which was illustrated by comparisons with fundamental diagrams. Accuracy will also be validated on real field data in our future work. For traffic data measurement we are going to use a variety of traffic sensors, including traditional inductive loops, traffic radars and video-detection systems for single vehicle tracking in traffic network (i.e. the technology available from our industrial partners and agencies, which is already in use). Fusing data from these traffic sensors into a coherent, consistent, and reliable picture of the prevailing traffic conditions (e.g. densities, speeds, flows) will be a critical and challenging task in the online traffic management or information system, which we are going to develop.

Another very important finding of our work is that the proposed model is really suitable for acceleration. We can achieve an almost linear speedup on widely accessible multiple-core systems, which allows us to do multiple in real time simulations. In means of number, we are able to simulate 1 minute in huge traffic networks shorter than in 5 seconds! For future experiments, it is planned to couple the current data provided by the online simulation to historical data collected in a database from sensors in order to provide a very precise traffic state forecast. Hence traffic simulation of large networks must be executed multiple in real time.

REFERENCES

[1] S. Maerivoet and B. De Moore, "Transportation planning and traffic flow models," Technical Report No. 05-155, Katholieke Universiteit Leuven, 2005.

[2] M. Brackstone and M. McDonald, "Car-following: a historical review," *in Transportation Research Part F: Traffic Psychology and Behaviour,* vol. 2, no. 4, Elsevier, 1999, pp. 181 – 196.

[3] M. F. Aycin and R. F. Benekohal, "Comparison of car-following models for simulation", Transportation Research Record, vol. 1678, Washington, 1999, pp. 116-127.

[4] M. F. Aycin and R. F. Benekohal, "Linear acceleration car-following model development and validation," *in Transportation Research Record: Journal of the Transportation Research Board,* vol. 1644, Washington, 1998, pp. 10 – 19.

[5] L.A. Pipes, "An operational analysis of traffic dynamics," *in Journal of Applied Physics,* 24(3), 1953, pp. 274 – 281.

[6] G. H. Bham and R. F. Benekohal, "A high fidelity traffic simulation model based on cellular automata and car-following concepts," *in Transportation Research Part C: Emerging Technologies,* vol. 12, no. 1, Elsevier, 2004, pp. 1 – 32.

[7] S. Algers, E. Bernauer, M. Boero, L. Breheret, C. di Taranto, M. Dougherty, K. Fox, and J. Gabard, "*Simulation modeling applied to road transport european scheme tests (SMARTEST)* – Review of Micro-Simulation Models." SMARTEST Project Deliverable D3, Institute for Transport Studies, University of Leeds, March 1998.

[8] Road and Motorway Directorate of Czech republic, "Roads and motorways in Czech Republic," *in Annual report,* Chapter 1, 2009 [online],http://www.rsd.cz/rsd/rsd.nsf/0/AE55C3DAD269424BC1257 5CB0050A3A7/$file/RSD2009en.pdf

[9] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *in Journal de Physique I*, vol. 2, Issue 12, 1992, pp.2221 – 2229.

[10] D. Chowdhury, L. Santen, A. Schadschneider, "Statistical physics of vehicular traffic and some related systems," *in Physics Reports 329,* 2000, pp. 199 – 329.

[11] D. E. Wolf, "Cellular automata for traffic simulations," *in Physica A,* vol. 263, 1999, pp. 438 – 451.

[12] S. F. Hafstein, R. Chrobok, A. Pottmeier, M. Schreckenberg, and F. C. Mazur, "A high-resolution cellular automata traffic simulation model with application in a freeway traffic information system," *in Computer–Aided Civil and Infrastructure Engineering,* vol. 19, no. 5, Blackwell Publishing, 2004 , pp. 338 – 350.

[13] K. Nagel, "High-speed microsimulations of traffic flow," Ph.D. Thesis, University of Cologne, 1995.

[14] S. Krauss, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics, Ph.D. Thesis, University of Cologne, 1998.

[15] J. Esser and M. Schreckenberg, "Microscopic simulation of urban traffic based on cellular automata," *in International Journal of Modern Physics C,* vol. 8, Issue 5, 1997, pp. 1025 – 1036.

[16] M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour, "Two lane traffic simulations using cellular automata," *in Physica A,* vol. 231, Issue 4, 1996, pp. 534 – 550.

[17] K. Nagel, D. E. Wolf, P. Wagner, and P. Simon, "Two-lane traffic rules for cellular automata: A systematic approach" *in Physica E,* vol. 58, Issue 2, 1998, pp. 1425 – 1437.

[18] E. G. Campari, G. Levi, and V. Maniezzo, "Cellular automata and roundabout traffic simulation," *in Cellular Automata,* ser. LNCS, vol. 3305, Springer Berlin, 2004, pp. 202 – 210.

[19] A. Dupuis and B. Chopard, "Cellular automata simulations of traffic: A Model for the city of Geneva," *in Networks and Spatial Economics,* vol. 3, Number 1, Springer Netherlands , 2003, pp. 9 – 21.

[20] R. Barlovic, J. Esser, K. Froese, W. Knospe, L. Neubert, M. Schreckenberg, and J. Wahle, "Online traffic simulation with cellular automata," *in Traffic and Mobility: Simulation–Economics Environment,* 1999, pp. 117 – 134.

[21] H. Rakha and B. Crowther, "Comparison of Greenshields, Pipes, and Van Aerde car-following and traffic stream models," *in Transportation Research Record: Journal of the Transportation Research Board,* vol. 1802, Washington, 2002, pp. 248 – 262.

[22] M. Van Aerde, "Single regime speed-flow-density relationship for congested and uncongested highways," *Presented at the 74th TRB Annual Conference,* Washington, 1995.

[23] M. Van Aerde and H. Rakha, "Multivariate calibration of single regime speed-flow-density relationships," *in Proc. of the Vehicle Navigation and Information Systems (VNIS) conference,* Seattle, Washington, 1995.

[24] E. Brockfeld, R. D. Kühne, and P. Wagner, "Calibration and validation of microscopic traffic flow models," *in Transportation Research Record: Journal of the Transportation Research Board,* vol. 1876, Washington, 2004, pp. 62 – 70.

[25] OpenMP Architecture Review Board, "OpenMP Application Program Interface," Version 3.0, 2008 [online], http://www.openmp.org/mp-documents/spec30.pdf

[26] P. Chakroborty and A. K. Maurya, "Microscopic analysis of cellular automata based traffic flow models and an improved model," *in Transport Reviews,* vol. 28, issue 6, 2008, pp. 717 – 734.

[27] G. H. Bham and R. F. Benekohal, "Development and validation of a high fidelity multi-lane traffic simulation model," *in Proc. of the 7th International Conference on Applications of Advanced Technologies in Transportation (AATT7),* Boston, 2007.

[28] J. L. Tripp, H. S. Morveit, A. A. Hansson, and M. Gokhale, "Metropolitan road traffic simulation on FPGAs," *in 13th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'05),* 2005, pp.117 – 126.

[29] Message Passing Interface (MPI) forum, "MPI: A Message-Passing Interface Standard," Version 2.2, 2009 [online], http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf

Paper II

Cellular Automata Based Traffic Simulation Accelerated on GPU.

CELLULAR AUTOMATA BASED TRAFFIC SIMULATION ACCELERATED ON GPU

Pavol Korcek, Lukas Sekanina and Otto Fucik

Brno University of Technology – Faculty of Information Technology
Department of Computer Systems
Bozetechova 1/2, 612 66 Brno
Czech Republic
{ikorcek, sekanina, fucik}@fit.vutbr.cz

Abstract: *Intelligent transportation systems become more and more important with the increasing traffic densities and safety requirements. A reasonably good traffic prediction can be obtained using microscopic traffic simulation models witch distinguish and trace every traffic entity. However, microscopic simulation requires considerable computing resources. In this paper, we propose to accelerate a cellular automata based microscopic traffic simulator using graphic processing units (GPU). The proposed accelerator provides speed-up of 204.65 with respect to a single core solution for problems instances containing 170 mil. cells, equivalent to 935 000 km of traffic network. This solution is sufficient to predict traffic simulations multiple in real-time.*

Keywords: *Traffic, Microsimulation, Cellular automata, Acceleration, GPU, CUDA.*

1 Introduction

Microscopic traffic simulation models, in contrast to macroscopic models, distinguish and trace every single vehicle and driver. The models are very precise and typically employ characteristics such as vehicle lengths, speeds, accelerations, time and space headways, vehicle and engine capabilities, as well as some rudimentary human characteristics that describe the driving behaviour [1], [2].

In the past, the use of microscopic models has been limited in scope and scale of the problem. The applications were restricted to small networks with a limited number of vehicles. With the advent of fast computers a large number of microscopic simulation models have been developed. For an illustration, the SMARTEST report (1999) identified 58 traffic simulation models of which 32 were analyzed [3]. Some of them are true microscopic simulators in the sense that they model the behaviour of each individual vehicle in the traffic flow. Another latest short overview of the microscopic traffic models (2004) can be found, for example, in [4].

Simulation results could be utilized in various so-called intelligent transportation systems (ITS) [7]. Such an example can be driver assistance or collision avoidance systems where simulations are utilized for the traffic state prediction. The most important requirement for these simulators is to simulate the traffic faster than real time, or even better, multiple in real-time. This isn't easily met when traffic is simulated on microscopic level of detail because there exist millions of simulated entities with complex relations among themselves [1], [2], [3], [4], [6]. Moreover, our approach attempts to extend the simulation beyond a few roads and intersections to large areas (e.g. we are going to model and simulate very huge networks such as the whole Czech Republic with approximately 55 000 km of road segments [9]. This length of road segments does not distinguish the directions, the number of lanes, e.g. the real number of kilometres is much higher when including these options. For instance, distance between Prague and Brno city is approximately 200 km, but because there are two lanes in each direction, total length of road segments is four times greater). Hence the acceleration of microscopic models continues to grow in importance.

The goal of this paper is to present an acceleration of microscopic traffic simulation using our advanced cellular automata (CA) based model [8]. For whole model acceleration, a big potential of today's common GPUs is used. In contrast to specialized and expensive accelerators, such as field programmable gate array (FPGA) based machines [10], we propose to utilize a very reasonable solution based on commodity hardware. We retrieved only 3 papers dealing with traffic simulation acceleration on common GPUs. The first one [15] is concentrated on traffic simulation for emergent behaviour. The acceleration of crowd behaviour was shown, which could be possibly mapped to a traffic model. Paper [15] differs from our model in one important aspect -- traffic topology (e.g. intersections) is not modeled. Another paper [16] also deals with traffic simulation but on a different level of detail. Authors showed an efficient execution of field-based vehicular mobility models with their hybrid method on GPU. Generally, the traffic simulation models are faster when the higher abstract level is used [17]. On the other hand, these models are not as precise as the true microsimulators [1], [3] and their approach is therefore not useful in our prediction-based traffic simulations [7]. The last and most recent paper [14] deals with a multi-agent traffic simulation. Different queue-based implementations were analyzed and speed-up of 67 was achieved when compared to a highly optimized Java version. Recall that we do not simulate queues and agents in these queues in our work; we are performing real traffic microsimulations. It means that

we have traffic lanes instead of queues with exact "agent" position on it. Another big difference is that we do not have „traffic on demand", which means to have pre-counted all traffic routes of each vehicle or driver.
The outline of this paper is as follows: After presenting the idea of general purpose computation on GPU in Section 2, the original and advanced model of traffic microsimulation using CA is described in Section 3. In Section 4 a mapping of model to the GPU is presented. Results of experimental evaluation are summarized in Section 5. Finally, conclusions are given in the last section of the paper.

2 General Purpose Computation on GPU

Over the last decade, the graphic cards found in common home PCs have evolved from mere display devices over 3D rendering devices to today's generally programmable devices. Relevant graphics device companies (NVIDIA and ATI/AMD) have come up with frameworks (SDKs) to program GPUs for general problems. These SDKs are named *FireStream* [11] and *CUDA* (Compute Unified Device Architecture) [12] and can be used on graphic cards of their SDK vendors. The presence of SDKs has changed premises rather dramatically. It has become feasible to take a given primarily CPU based algorithm and convert it for GPU execution in a straight way. The problem of porting algorithm in the same way to the various vendors' GPUs has been solved. A framework for writing programs that can be executed across different GPUs and also across heterogeneous platforms (consisting of CPUs, GPUs, and other elements) has been introduced (*OpenCL* [13]). Hence every programmer can easily utilize a modern GPU for general computation. In common, whole processing flow on GPUs consists of four successive steps also shown in Fig. 1:

1. *Copy data from the host memory to the GPU memory.*
2. *CPU instructs the process to GPU.*
3. *GPU executes each processing element in parallel.*
4. *Copy the result from GPU memory to the host memory.*

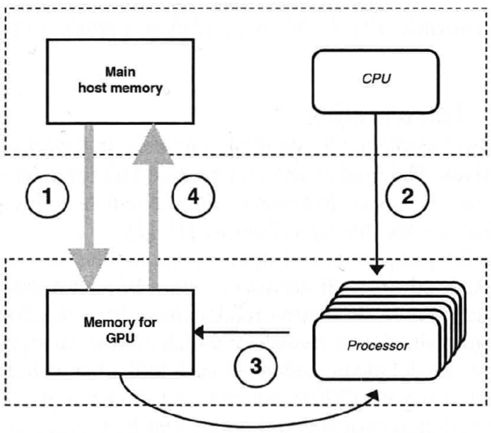

Fig. 1: Processing flow using GPU.

2.1 CUDA

We have chosen CUDA for better compatibility with chosen NVidia graphic card (see Tab.1). This SDK with the new Fermi architecture [31] promises the best achievable speed-ups on GPU and an active community of developers [12]. CUDA can be performed on any NVidia graphic card from the *GeForce-8* generation on both Linux and Windows operating system.

2.1.1 SW model

From software point of view, the CUDA framework is an extension to the standard C language that enables us to write code for CPU and GPU in the same file. It therefore is rather easy to program in for any experienced C/C++ programmer. It basically adds the keywords *__device__*, *__global__*, *__host__* as method decorators to indicate whether the methods are run on the host CPU or the GPU and from where they could be called. Additionally it adds own syntax for describing with how many parallel threads a method should be started. The methods declared as a global are so called "kernels". These kernels could be called from the "host" (the PC/CPU the graphics device is running in) and run on the GPU. These kernels run simultaneously on different data sets in multiple threads of execution because GPU is a massively parallel processor, which supports thousands of active threads [12]. As we only have a limited number of "real" hardware processors, threads are joined to thread blocks, which again are bundled to a grid of thread blocks. Hardware thread scheduler manages recently up to 1536 simultaneously active threads for each streaming multiprocessor across 16 kernels [32]. Threads are sharing a set of registers as well as a rather small on-chip memory area. Threads running in the same block can be synchronized with each other, whilst threads amongst block of threads cannot be easily synchronized. Threads within a block can also access a small amount of additional local shared memory. Though only small amount of threads can make up one block of threads the grid of blocks can run thousands or even trillions of threads in parallel as can be seen in Fig. 2. More details about CUDA application programming can be found in [12].

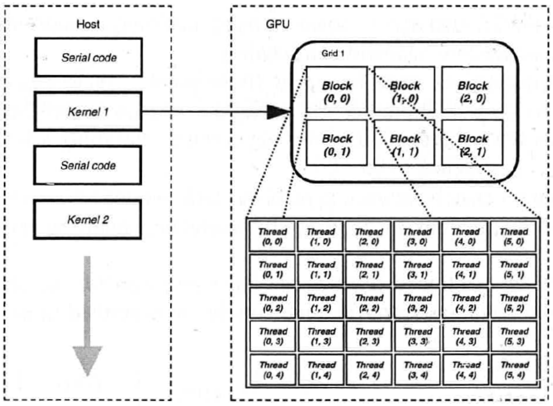

Fig. 2: CUDA programming model.

2.1.2 HW model

Today's graphic cards are connected via PCI Express system bus to the host computer. However, this bus is a potential bottleneck or possible source of big latency in such applications where a huge number of transfers are needed between the GPU and the host CPU [18]. The architecture of the NVidia series GPU is shown in Fig. 3. There is an on-board main memory (with GB capacity). It also acts as an entry or output point during the communication with CPU. Unfortunately, big capacity is outweighed with high latency and this memory is not cached anywhere. The second part is a graphic chip, which consists of *N* symmetric multiprocessors (SM) each based on SIMD (Single Instruction Multiple Data) architecture. These SMs have their own on-chip shared memory which can be accessed by all the threads of a thread block, texture and constant memory which are "readily" cached on-chip, built-in hardware based scheduler and also shared instruction unit with a hardware thread scheduler. Each SM consists of *M* functional units (the newest NVidia Fermi GPUs have 32 functional units per single SM [32]), which have own set of registers. These registers are the fastest memory components in the graphic card memory hierarchy model, but on the other hand, they have the smallest capacity.

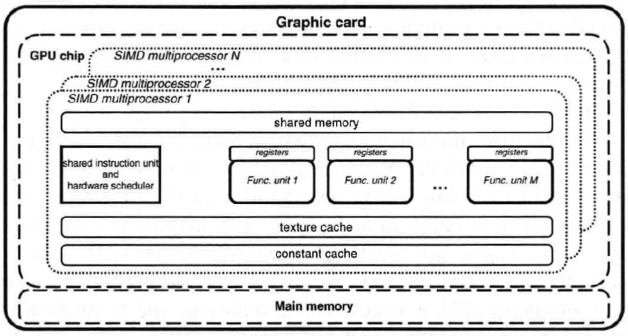

Fig. 3: The architecture of NVidia GPUs.

2.1.3 Improvements in a new generation NVidia Fermi GPUs

The onfiguration of our test system is shown in Tab. 1. As it can be seen, the new Fermi architecture [31] is in place with our *NVidia GTX 480* graphic card. Fermi, compared to previous GPU architectures, offers some improvements:

- *Indirect branching,* which is common for CPUs, but was previously prohibited for NVidia GPUs.
- *Single unified address space* for local (thread private), shared (for each thread block) and global memory (device and system-wide) is used in this generation NVidia hardware.
- *Increased register file* to 32 000 entries or 128 kB in total. The number of registers per execution unit is dropped by half, to 1 000 entries, versus the previous NVidia GPU generation. The register file is designed to sustain full 32-bit throughput every clock cycle.
- *L1 cache per each core* that can be either configured as 16 kB or 48 kB in size. The L1 caches are tied together with a shared L2 cache. Shared memory and L1 data cache (L1D) are respectively used for explicit and implicit communication between threads. The array can be configured with a 16 kB or 48 kB shared memory, with the remainder used for L1D cache.
- *L2 cache* (768 kB) and extra atomic execution units are used to accelerate atomic operations. Fermi's unified L2 cache serializes atomic operations in a different manner, to reduce the number of memory write backs. For

non-atomic operations, the L2 also acts to coalesce many different memory requests (e.g. from a whole warp) into fewer transactions, improving bandwidth utilization.

- *Newer scheduler* can maintain the state for up to 16 different kernels, one per SM. Each SM runs a single kernel, but the ability to keep multiple kernels in flight increases its utilization, especially when one kernel begins to finish and has fewer blocks left. More importantly, assigning a kernel per core means that smaller kernels can be efficiently dispatched to the GPU.
- *Reduced latency* for context switch between kernels has been reduced 10 times (to around 25 microseconds).
- *Full predication* for all instructions to improve the instruction fetch by removing bubbles caused by taken branches.

As it can be seen, all of previously mentioned features are more common for the standard CPU architecture. Traffic simulation, especially cellular automata based traffic microsimulation described in next section, can potentially highly exploit from all of them.

	CPU	PC memory	GPU	GPU memory	OS + SW
System configuration	*Intel Core i7 920@3.3GHz*	*12 GB*	*GTX 480 with 15 SM*	*1.5 GB*	*Ubuntu 10.4 LTS x64 Cuda 3.0, gcc 4.1 (with CUDA support)*

Tab. 1: Test system configuration.

3 Cellular automata based traffic simulation

Cellular automata based models have become popular in the area of microscopic traffic flow simulations because of their simplicity and suitability for acceleration. The first highly suitable model for acceleration of single-lane freeway traffic was introduced by Nagel and Schreckenberg in 1992 [19]. Because of the model's simplicity, it is possible to perform millions of updates in a second [20]. Thus, it may be used for simulating a high volume of traffic over very large networks. Due to some critics of unrealistic behavior (such as [21]), simple CA based models are often extended. On the other hand, CA models were shown to be able to capture all basic phenomena that occur in traffic flows [22], not only in the field of vehicular traffic flow modeling, but also in other fields such as pedestrian behavior, escape and panic dynamics, etc.

Nagel's and Schreckenberg's traffic model [9] was initially defined on a one-dimensional array of cells with open or periodic boundary conditions and with every single cell representing a road segment. A local transition function (a rule) defines the new state of a cell on the basis of its current state and the state of neighboring cells. Globally viewed, it describes the movements of vehicles from one cell to another cell in a discrete way.

In space domain, each cell represents only a defined length of road segment, so space is coarse-grained. This coarse graininess is fundamentally different from the usual microscopic models, which adopt a semi-continuous space. The length of 7.5 m is mostly chosen because each car occupies about this amount of space in a complete jam [9]. So this is the average length of a car on the road, including a constant gap in front of and behind each car. In order to model other kinds of vehicles in CA based models, it is better to use another cell length, or even more precisely, use more smaller cells for representing a single vehicle -- especially for bigger types of vehicles [23], [24].

Each CA simulation step represents an amount of time in reality. Based on suggestions about driver reaction time given in [25], [26] and field data measurements [23], it is possible to represent one simulation step between 0.6 and 1.5 s. In the CA model this value is crucial because together with cell length it gives us the granularity of minimal model speed jumps[1].

Normally, urban traffic road networks are very complex. Paper [27] has shown that arbitrary kinds of road and intersections can be reduced to only a few basic elements. For constructing more complicated traffic networks, we shall simply connect more various kinds of cells to the desired topology. So it is possible to build more traffic lanes [28], roundabounds [29] or very complex topologies for whole cities [30]. Fig.4. shows an example of complicated road intersection modeled using CA.

Properties of single lane traffic are originally modeled on the basis of integer valued probabilistic cellular automaton rules [20]. The local transition function can be formalized as follows: each vehicle in a cell has an integer velocity v_v with values between zero and v_{max}. Real vehicle speed v_{act} is multiplication of minimal speed jump and actual vehicle velocity v_v.

We let $\gamma(i)$ denote the cell gap in front of cell i. For an arbitrary configuration, one update of the simulation system consists of the following four consecutive steps, which are performed in parallel for all vehicles [19]:

1. **Acceleration:** *if* $(v_v < v_{max})$ *and if* $(\gamma(i) > v_v)$ *then* $v_v = v_v + 1$.
2. **Slowing down:** *if a vehicle at site i sees the next vehicle at site* $i + j$ $(j \leq v_v)$, *it reduces its speed, so* $v_v = j - 1$.
3. **Randomization:** *with probability p, the velocity of each vehicle (if* $v_v > 0$*) then* $v_v = v_v - 1$.
4. **Car motion:** *each vehicle is advanced* v_v *sites.*

[1] For instance, if simulation step is *1 s* and cell length is *5 m* [19], minimal speed jumps are 5 m/s (27 km/h).

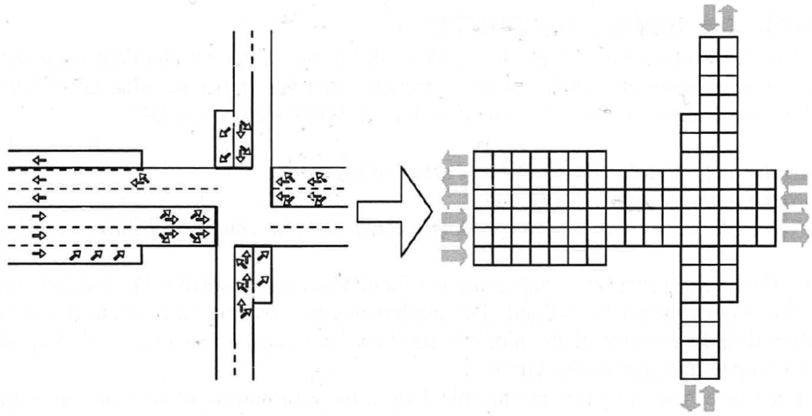

Fig. 4: A road intersection (left side) modeled with the cellular automata (right side).

This simple CA based traffic model shows nontrivial and realistic behavior of car-following principle [19]. Step 3 is essential in simulating traffic flows since the dynamics is completely deterministic, and without this randomness, every initial configuration of vehicles and corresponding velocities reaches a stationary pattern which is shifted backwards. This simple model is capable of reproducing characteristic properties of real traffic, such as certain aspects of flow-density relation, spatio-temporal evolution of jams, stop-and-go waves. More details can be found in [19], [6].

For multilane traffic simulation not only car-following rules have to be applicable. The behavior of car overtaking should be considered as an additional set of rules to the basic model. Such an example can be found in paper [28] where each vehicle is able to overtake other vehicle only when its speed is sufficient and there is enough space for safe overtaking. These basic rules follow the field behavior of drivers.

3.1 Advanced cellular automata model

Basic cellular automata traffic microsimulation model has been updated to the advanced cellular automata in previous work [8], in order to *a) adapt simulation model to local conditions, b) achieve better precision of model* and finally *c) accomplish faster microsimulations.* Based on local measurements, cell length has been reduced from 7.5 m to 5.5 m. The simulation time has been adjusted from 1 sec. to 1.2 sec according to average driver reaction time [23]. Both these values give minimal speed jump of 16.5 km/h. Cell neighbor is 11, and together with cell length it gives a maximal distance of 60.5 m between the leader and the follower. As observed, for example in [25], for separations greater than 61 m, car-following is negligible on a driver's behavior. For cell state encoding a bit level encoding utilizing only one 16-bit integer has been used. Finally, double-buffering technique for previous and next cell state storage has been utilized.

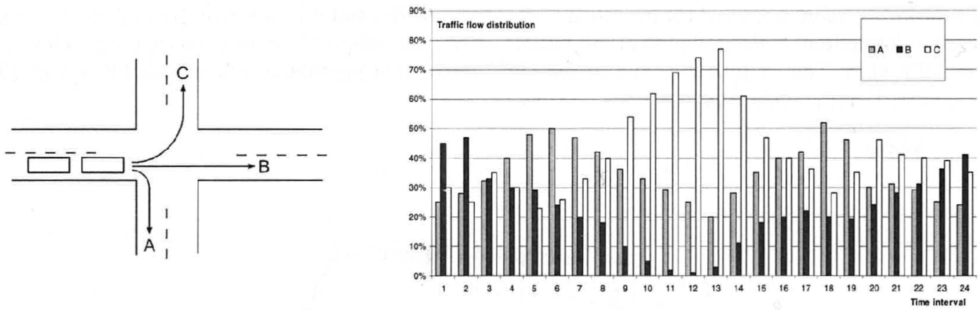

Fig. 5: Simple road intersection (right side) with traffic flow distribution over the time (left side).

From the global point of view the behavior on road intersections should be defined. In the original model, this was done using traffic routes for each vehicle [19]. There isn't available such information in our case, hence another solution must be considered. Based on data provided from our industrial partner we are able to measure an average traffic flows on selected road intersections. An example of flow distribution on a simple road intersection over time is shown in Fig. 5. Each incoming vehicle (Fig. 5 on the left) has one of three (A, B or C) chance to turn and the probability (on the y-axis) of turning vehicle in any direction is various in different time interval (on the x-axis).

All other details about advanced cellular automata can be found in paper [8]. This updated model has been already proposed to model road traffic more realistically and also quite efficiently employing standard CPU based multicore devices, where, depending on the number of processing elements, a nearly linear speed-up has been achieved [8].

4 CA based traffic simulation using GPGPU

Traffic microsimulation with advanced cellular automata model needs to run an identical local transition function on different cells (that can be considered as different data), meaning that this is exactly what GPUs have been designed to speed-up. Proposed microsimulation was implemented in three different ways using GPU:

a) road intersection cells simulated on CPU and other cells in GPU,
b) traffic network simulated only in GPU, and
c) scenario of b) with standard operations (adding, subtraction and comparison) instead of bit level operations.

The implementation (a) exploits a relatively huge amount of data transfers between GPU memory and the host memory, but the synchronization can explicitly be defined. The implementation (b) gets all the data to the GPU main memory. Simulation ends when desired number of iterations is reached. The implementation (c) is proposed to compare the effectiveness of various operator types on the GPU.

Each cell state is stored in 8 bytes (16 bits is multiplied by 2 for information redundancy used in advanced cellular automata model for better accuracy [8] then multiplied again by 2 to employing the double-buffering technique). In both cases, a simple traffic network model has been implemented, where a road intersection occurs every 500 m. This value reflects the mean length of roads after which road intersection occurs in reality [9]. Every CA cell knows if it is the input cell to the road intersection based on its position. Supposing 24 time intervals on every road intersection (see Fig.5 above) and supposing 16 bits used for encoding one probability value in one direction (of 3: A, B or C), we need 768 bits (96 B) for encoding one road intersection which are stored separately for every intersection on host memory and cached (a) or in GPU main memory and also cached (b, c).

5 Results

To measure the speed-up of the proposed solution each of implemented versions was tested on our system. We recorded the start time and after reaching demanded number of simulation steps (equal to 42 min), the execution of program was terminated and results were sent to CPU. After this, the running time was recorded (including time of bus transfers). Fig. 6 shows the achieved speed-up compared to a single core version running on *Intel Xeon CPU5420 @ 2.5GHz* [8]. The speed-up of 204.65 was achieved in the fastest implementation. The fastest solution is the microsimulation implemented entirely on the GPU utilizing standard operations. The slowest implementation is running on both CPU and GPU. One very important finding is that the GPU solution is faster (relatively to the CPU version) when simulating higher traffic density. Another finding is that the GPU solution is ineffective in case when the size of problem (number of CA cells) is small. It is probably because the initialization phase is very time consuming. Consequently, the simulated time should be at least 42 min (for the smallest problem size -- 250 000 cells). This is because of the initialization phase overhead.

Other experimental results can be seen in Fig. 7. We used the fastest implementation with the best speed-up (simulating 170 mil. of CA cells) to evaluate dependency of the GPU acceleration on the *p-value* change. As it was explained in Section 3, this probability value is crucial for modeling traffic realistically, and it is normally set to 30%. We found out, that the acceleration is minimal if this value is in the middle (50%) and relatively symmetric on other sides (e.g. 30% almost same as 70% etc.). This is probably due to the GPU instruction prediction, when both (30% and 70%) are predicted (and executed) in the same way.

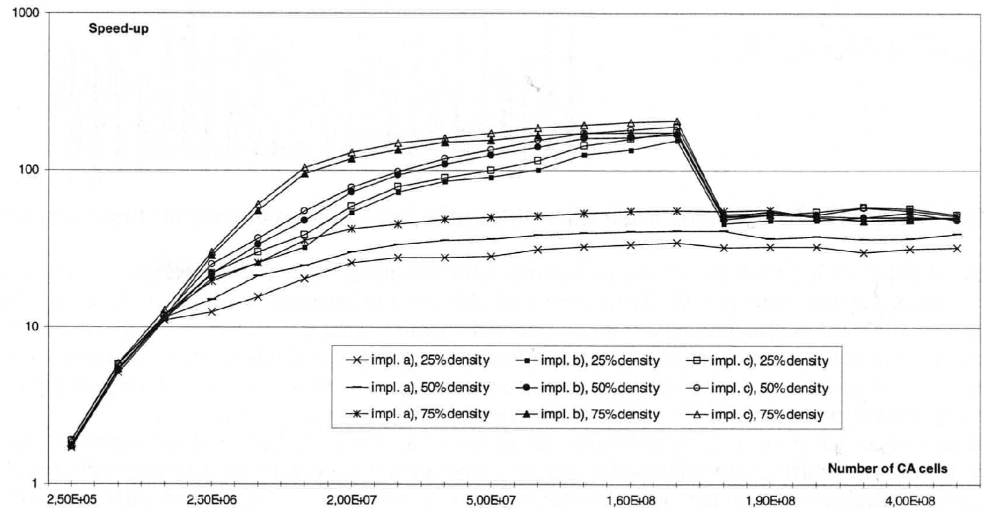

Fig. 6: GPU speed-up of three different implementations with respect to a single core version running on the Intel Xeon E5420 @ 2.50 GHz. Each run is 42 minutes of the real traffic.

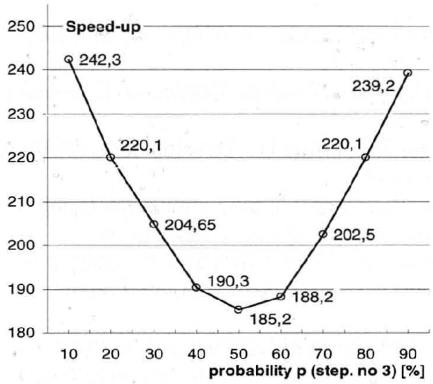

Fig. 7: GPU speed-up with different algorithm settings. P-value on the y-axis is the probability of randomization step in the CA microscopic traffic simulation model.

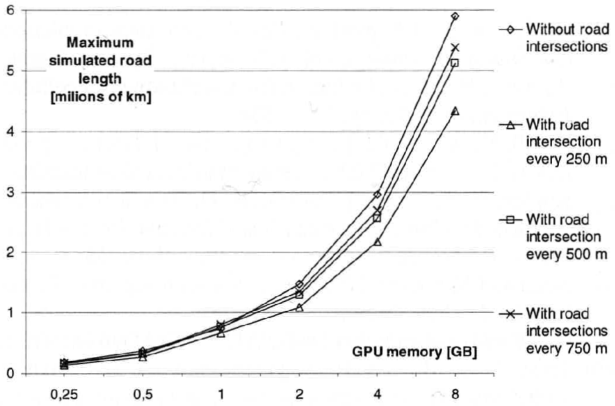

Fig. 8: The maximum theoretical length of simulated road network depending on the GPU memory size.

Fig. 8 shows the maximum theoretical length of simulated road network (with or without intersections) depending on capacity of the GPU memory. This maximum was nearly achieved ("nearly" because the GPU memory is also used for storing another kind of data -- programs, etc. [33]).

The estimated efficiency of simple multicore version of cellular automata based traffic microsimulation introduced in [8] is shown in Fig. 9. For comparison, efficiency of solution presented in this paper is also shown. The number of processing elements in our GPU is used (480) as the number of cores. A simple CPU multicore version is more effective than our GPU implementation, but on the other hand, real results were obtained unig 4 and 8 cores; the remaining values were estimated based on proposed efficiency.

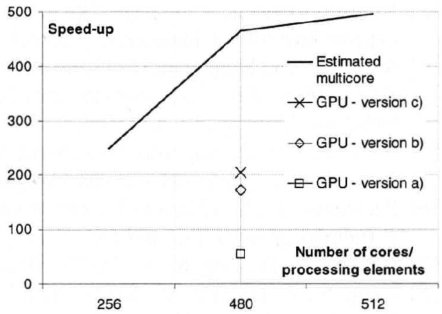

Fig. 9: Efficiency of the simple multicore version [8] and the GPU versions. All of them are with the same configuration of the model.

6 Conclusion

In this paper, it was shown that modern GPU can be effectively used for acceleration of microscopic and cellular automata based traffic simulation. The acceleration profits from the new Fermi architecture. On the other hand, there exist also some limitations. It was shown that the bus traffic is a bottleneck and therefore all data should be stored at least in the relatively big GPU memory (1.5 GB). Moreover, this also means that the GPU memory makes limitations to size of simulated road length (see speed-up drop in Fig. 6). Even this limitation, we are able to simulate 17 times greater traffic area than the traffic network of the Czech Republic [9] multiple in real-time. Finally, we can claim that our GPU version is still reasonable for traffic microsimulation even if multicore version has a higher efficiency. It is because there isn't any multicore device with a suitable number of cores.

Acknowledgement: This work was partially supported by the Czech Science Foundation under projects P103/10/1517 and GD102/09/H042, by the research program MSM 0021630528 and the BUT grant FIT-S-11-1.

References:

[1] Maerivoet, S. and De Moore, B., Transportation planning and traffic flow models, *Technical Report No. 05-155,* Katholieke Universiteit Leuven, 2005.

[2] Brackstone, M. and McDonald, M., Car-following: a historical review, *Transportation Research Part F: Traffic Psychology and Behaviour,* vol. 2, no. 4, Elsevier, 1999, pp. 181 – 196.

[3] Algers, S., et. al., Simulation modelling applied to road transport European scheme tests (SMARTEST) – Review of Micro-Simulation Models, *SMARTEST Project Deliverable D3*, Institute for Transport Studies, University of Leeds, March 1998.

[4] Aycin, M., F. and Benekohal, R., F., Linear acceleration car-following model development and validation, *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1644, Washington, 1998, pp. 10 – 19.

[5] Dupuis, A. and Chopard, B., Cellular automata simulations of traffic: A Model for the city of Geneva, *Networks and Spatial Economics,* vol. 3, Number 1, Springer Netherlands , 2003, pp. 9 – 21.

[6] Barlovic, R., et. al., Online traffic simulation with cellular automata, *Traffic and Mobility: Simulation–Economics Environment,* 1999, pp. 117 - 134.

[7] Boxill, S., A. and Yu, L., An Evaluation of Traffic Simulation Models for Supporting ITS Development, *Technical report,* Number 167602-1, Texas Transportation Institute, Texas, 2000. 116 p.

[8] Korček, P., Sekanina, L. and Fučík, O., Towards Scalable and Accurate Microscopic Traffic Simulation Using Advanced Cellular Automata Based Models, *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems Workshops,* Madeira, IEEE ITSC, 2010, pp. 27-35, ISBN 978-972-8822-20-0.

[9] Road and Motorway Directorate of Czech republic, Roads and motorways in Czech Republic, *Annual report,* Chapter 1, 2009, Online: http://www.rsd.cz/rsd/rsd.nsf/0/AE55C3DAD269424BC12575CB0050A3A7/$file/RSD2009en.pdf [9/2010]

[10] Tripp, J., L., Morveit, H., S., Hansson, A., A. and Gokhale, M., Metropolitan road traffic simulation on FPGAs, *13th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'05),* 2005, pp. 117 - 126.

[11] AMD FireStream www page. AMD Accelerated Parallel Processing (APP) SDK (formerly ATI Stream), Online: http://developer.amd.com/gpu/AMDAPPSDK/Pages/default.aspx [3/2011].

[12] Nvidia CUDA www page. CUDA Zone, Online: http://www.nvidia.com/object/cuda_home_new.html, [3/2011].

[13] Khronos OpenCL www page. OpenCL - The open standard for parallel programming of heterogeneous systems, Online: http://www.khronos.org/opencl/ [2/2011].

[14] Nagel, K. and Strippgen, D., Using common graphics hardware for multi-agent traffic simulation with CUDA, *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools '09),* ICST 2009, Belgium, 2009, pp. 1 – 8, ISBN: 978-963-9799-45-5.

[15] Passos, E., et. al., Supermassive crowd simulation on GPU based on emergent behavior. *Proceedings of the Seventh Brazilian Symposium on Computer Games and Digital Entertainment,* 2008, pp. 81 – 86.

[16] Perumalla, K., S., Efficient Execution on GPUs of Field-Based Vehicular Mobility Models, *22nd Workshop on Principles of Advanced and Distributed Simulation (PADS '08),* Roma, 2008, pp. 154 – 155.

[17] Immers, L., H., Logghe, S., Traffic Flow Theory, Katholieke Universiteit Leuven, English version, 2005. p. 39.

[18] Ueng, S., Z., Lathara, M., Baghsorkhi, S., S. and Hwu, W., W., *CUDA-Lite: Reducing GPU Programming Complexity, Lecture Notes in Computer Science: Languages and Compilers for Parallel Computing,* Vol. 5335, 2008, pp. 277 – 288.

[19] Nagel, K. and Schreckenberg, M., A cellular automaton model for freeway traffic, *Journal de Physique I,* vol. 2, Issue 12, 1992, pp. 2221 – 2229.

[20] Nagel, K., High-speed microsimulations of traffic flow, Ph.D. thesis, University of Cologne, 1995.

[21] Chakroborty, P., and Maurya, A., K., Microscopic analysis of cellular automata based traffic flow models and an improved model, *Transport Reviews,* vol. 28, issue 6, 2008, pp. 717 – 734.

[22] Chowdhury, D., Santen, L., Schadschneider, A., Statistical physics of vehicular traffic and some related systems, *Physics Reports 329,* 2000, pp. 199 – 329.

[23] Bham, G., H. and Benekohal, R., F., A high fidelity traffic simulation model based on cellular automata and car-following concepts, *Transportation Research Part C: Emerging Technologies,* vol. 12, Elsevier, 2004, pp. 1 – 32.

[24] Hafstein, S. F., et. al., A high-resolution cellular automata traffic simulation model with application in a freeway traffic information system, *Computer–Aided Civil and Infrastructure Engineering,* vol. 19, no. 5, Blackwell Publishing, 2004 , pp. 338 – 350.

[25] Aycin, M., F. and Benekohal, R., F., Comparison of car-following models for simulation, *Transportation Research Record,* vol. 1678, Washington, 1999, pp. 116-127.

[26] Aycin, M. F. and Benekohal, R., F., Linear acceleration car-following model development and validation, *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1644, Washington, 1998, pp. 10 – 19.

[27] Esser, J. and Schreckenberg, M., Microscopic simulation of urban traffic based on cellular automata, *International Journal of Modern Physics C,* vol. 8, Issue 5, 1997, pp. 1025 – 1036.

[28] Nagel, K., Wolf, D., E., Wagner, P. and Simon, P., Two-lane traffic rules for cellular automata: A systematic approach, *Physica E,* vol. 58, Issue 2, 1998, pp. 1425 – 1437.

[29] Campari, E., G., Levi, G. and Maniezzo, V., Cellular automata and roundabout traffic simulation, *Cellular Automata,* ser. LNCS, vol. 3305, Springer Berlin, 2004, pp. 202 – 210.

[30] Dupuis, A. and Chopard, B., Cellular automata simulations of traffic: A Model for the city of Geneva, *Networks and Spatial Economics,* vol. 3, Number 1, Springer Netherlands, 2003, pp. 9 – 21.

[31] NVidia Fermi www page. Next generation CUDA Architecture, Online: http://www.nvidia.com/object/fermi_architecture.html, [4/2011].

[32] Glaskowsky, P., N., NVidia's Fermi: The First Complete GPU Computing Architecture, White Paper, 2009.

[33] Nguyen, H., GPU Gems 3. Addison-Wesley Professional, 2007, ISBN 978-0321515261.

Paper III

Evolutionary Approach to Calibration of Cellular Automaton Based Traffic Simulation Model.

2012 15th International IEEE Conference on Intelligent Transportation Systems
Anchorage, Alaska, USA, September 16-19, 2012

Evolutionary Approach to Calibration of Cellular Automaton Based Traffic Simulation Models

Pavol Korcek, Lukas Sekanina and Otto Fucik

***Abstract*— Microscopic traffic simulation models have become very popular in the evaluation of transportation engineering and planning practices in the past few decades. To achieve high fidelity and credibility of simulations, a model calibration and validation must be performed prior to deployment of the simulator. In this paper, we proposed an effective calibration method of the microscopic traffic simulation model. The model is based on the cellular automaton, which allows fast large-scale real-time simulation. For its calibration, we utilized a genetic algorithm which is able to optimize different parameters much better that a human expert. Furthermore, it is possible to readjust the model to given field data coming from standard surveillance technologies such as loop detectors in our case. We have shown that the precision of simulations can be increased by 20 % with respect to a manually tuned model.**

I. INTRODUCTION

Microscopic traffic simulation models distinguish and trace every single vehicle and driver on the road. These models typically employ the characteristics such as individual vehicle speeds, accelerations, time and distance headways to the preceding vehicle or some rudimentary human characteristics that describe driving behavior [1], [2]. This allows us to simulate the traffic very precisely. Today, the microscopic traffic simulation software is widely accepted and applied in all branches of transportation engineering as an efficient and cost effective analysis tool. On the other side, a lot of modeled entities extremely increase the simulation time.

A. Model performance

In case when the simulation is used as a part of a complex ITS (e.g. for online traffic states prediction), the simulation time becomes extremely critical. To be able to simulate large traffic at the microscopic level of detail, different acceleration techniques have already been developed. Some of them utilized special hardware [3], other ones employed multicore architectures or even high performance graphic cards [4] that are becoming parts of nowadays commodity hardware too. The simulation time significantly depends on the traffic model representation (e.g. form of the model). It seems that a reasonable representation is used when the traffic is simulated with cellular automaton (CA) based models. Recently, CA have become popular in the area of microscopic traffic simulations because of their simplicity and suitability for acceleration. For example, in paper [5], an online CA based simulation model of city of Duisburg was introduced. The authors showed that for the most frequently occurring densities the road network of Duisburg can be simulated around 100 times faster than real time. It is necessary to mention that their road network is modeled using only 22 000 cells and their simulator is not suitable for more cells. An example of explicitly implemented parallelism in a simulation model was presented in [6]. The authors have used a CA model to implement a traffic simulator for city of Geneva and they reported a super linear speed-up for 15 000 vehicles. However, their simulations are not expected to match the exact traffic situation of Geneva, as they mentioned. CELLSIM is another good example of a recent CA based microscopic simulator [7]. Computational performance of this model, as for other microscopic models, is dependent on the number of vehicles in the system. It was shown, that CELSIM is able to simulate around 2300 vehicles in shorter than real time. As computational power is constantly increasing, we can assume that this number of vehicles will increase too.

In our previous work, we also utilized such techniques to speed-up our large-scale CA based model with special libraries for explicitly programmed parallel model for multicore processors [8] or for modern graphic cards [9], which allowed us to simulate huge networks (e.g. 935 000 km with 50% crowdedness) multiple in real-time.

B. Model quality

However, not only the performance is crucial in microscopic traffic simulation models. A very important stage of development of any traffic model is its comparison with reality, namely calibration and validation. In [10], authors proposed an effective three-step process for the microscopic traffic model calibration. Another paper [11] gives some basic guidelines for calibration of microscopic simulation models in form of framework and applications. The developers usually calibrate and validate the model on their own using some data sets that they have access to and publish the results obtained. For example, in paper [12] authors tried to perform a simple calibration of ten microscopic traffic simulation models in a way that the models were calibrated and compared to each other with the GPS based field data from year 2004 in Japan. But it should be noted, that in almost all previous calibration approaches, some real data are desired in a form, which is not generally available. It was shown that it is important to find a few basic parameters for the model calibration [13]. Namely a driver sensitivity

This work was partially suported by the Czech Science Foundation under projects P103/10/1517 and GD102/09/H042, by the IT4Innovations Centre of Excellence project CZ.1.05/1.1.00/02.0070, by the research program MSM 0021630528, FIT BUT grant FIT-S-12-1 and CAMEA spol. s r. o.

Pavol Korcek, Lukas Sekanina and Otto Fucik are with Brno University of Technology, Faculty of Information Technology, IT4Innovations Centre of Excellence, 612 66 Brno, Czech Republic {ikorcek, sekanina, fucik}@fit.vutbr.cz.

(e.g. reaction time), a jam density headway and free-speed (maximum speed when vehicle is not constrained) have to be determined. It was also stated that this process is neither a straight-forward nor an easy task. For example, while the free-speed is relatively easy to estimate in the field and generally lies between the speed limit and the design speed of the roadway, the jam density headway is more difficult to calibrate but typically ranges between 110 to 150 vehicles/km/lane. The driver sensitivity factor is extremely difficult to calibrate because it can not be measured using standard surveillance technologies (e.g. detection loops that work on magnetic-induction principle).

C. Goals of the paper

In this work we propose to utilize our CA based microscopic traffic simulation model, which was shown not only to be extremely fast to achieve multiple in real-time simulations, but also updated to eliminate unwanted properties of ordinary CA based models. The quality of this updated model has been previously evaluated by comparison with fundamental diagrams and discussed [8]. Secondly, we will try to calibrate parameters of this model to various field data that can be obtained from standard surveillance technologies. We will show, that it is possible to achieve a better precision of simulation for a given road segment. Moreover, except ordinary traffic parameters, we will also optimize some CA model parameters as well as calibrate some parameters which, as stated for example in [13], are extremely difficult to calibrate with other common techniques. The optimization/calibration will be performed by genetic algorithm (GA).

The rest of the paper is organized as follows. Section II introduces a stream model calibration technique that allows us to fairly compare real data with the calibrated simulation model. Then, in Section III, cellular automaton based microscopic traffic simulation model is presented with all its updates and parameters utilized in the process optimization. Section IV describes the process of optimization of the model with selected GA. Results of experimental evaluations for each field data set are then presented in Section V. Finally, conclusions and suggestions for future work are given in the last Section VI.

II. STREAM MODEL CALIBRATION

Gazis et al. [14] were the first ones who formulated a relation between microscopic and macroscopic traffic models. In their theory, the flow rate can be expressed as the inverse of the average vehicle time headway. Similarly, the traffic stream density can be approximated for the inverse of the average vehicle spacing for all vehicles within a section of the roadway.

Road traffic is always in a specific state that can be characterized by three macroscopic parameters: flow (q), vehicle space-mean speeds (u) and density (k) shown in Eq. 1:

$$q = k \times u. \tag{1}$$

By combining all the possible traffic states (steady or non-steady) in an equilibrium function, one can derive three fundamental traffic diagrams [15]. A diagram shows the relation between two of the three variables (speed-density, speed-flow, flow-density). The third variable can always be recovered using Eq. 1.

It was shown [16], that estimation of the traffic stream parameters requires the calibration of a traffic stream model to field data. This effort entails making some decisions, such as defining the functional form to be calibrated, identifying dependent and independent variables, defining the optimum set of parameters, and finally developing an optimization technique to compute the set of parameter values. Unfortunately, in the traffic flow theory it is not always clear which variable should be set to be independent or dependent. In paper [13] and later in [17], authors developed a calibration approach that minimizes the orthogonal error about the fundamental diagram to estimate the expected value of the traffic stream parameters. The model, which is also used to preprocess our data, is briefly described here, however a more detailed description can be found in [17]. Also, this approach is unique because it does not require the identification of dependent or independent variables since it applies a neutral regression (i.e. minimizes the orthogonal error as shown in Eq. 2).

$$E = \sum_i \left\{ \left(\frac{u_i - \widehat{u_i}}{\widetilde{u}} \right)^2 + \left(\frac{q_i - \widehat{q_i}}{\widetilde{q}} \right)^2 + \left(\frac{k_i - \widehat{k_i}}{\widetilde{k}} \right)^2 \right\} \tag{2}$$

where $\forall i$:

$$\widehat{k_i} = \frac{1}{c_1 + \frac{c_2}{u_f - \widehat{u_i}} + c_3 \widehat{u_i}}$$

$$\widehat{q_i} = \widehat{k_i} \times \widehat{u_i}$$

$$\widehat{q_i}, \widehat{k_i}, \widehat{u_i} \geq 0$$

$$\widehat{u_i} < u_f$$

and

$$0.5 u_f \leq u_c \leq u_f$$

$$q_c \leq \frac{k_j u_f u_c}{2u_f - u_c}$$

$$c_1 = \frac{u_f}{k_j u_c^2} (2u_c - u_f)$$

$$c_2 = \frac{u_f}{k_j u_c^2} (u_f - u_c)^2$$

$$c_3 = \frac{1}{q_c} - \frac{u_f}{k_j u_c^2}$$

$$u_f^{min} \leq u_f \leq u_f^{max}; u_c^{min} \leq u_c \leq u_c^{max}$$

$$q_c^{min} \leq q_c \leq q_c^{max}; k_j^{min} \leq k_j \leq k_j^{max}$$

where u_i, k_i, and q_i are the field observed space-mean speed, density, and flow measurements, respectively. The speed, density, and flow variables with hats are estimated speeds, densities, and flows while the tilde variables are the maximum field observed speed, density, and flow measurements.

c_1, c_2, and c_3 are then selected model constants. The objective function ensures that the formulation minimizes the normalized orthogonal error between three-dimensional field observations and the functional relationship – in this case the *Van Aerde* functional form is utilized. The error terms are normalized in order to ensure that the function is not biased towards reducing the error in one of three variables at the expense of the other two variables. This data normalization ensures that the parameters in each of the three axes range from 0.0 to 1.0 and thus a minimization of the orthogonal error provides fitting equivalent across all three axes. The initial constraint, which is non-linear, ensures that the Van Aerde functional form is maintained, while the second constraint is added to constrain the third dimension, namely the flow rate. The third and fourth set of constraints guarantee that the results of the minimization formulation are feasible. The fifth and sixth set of constraints ensures that the four parameters that are selected do not result in any inflection points in the speed-density relationship (e.g. it ensures that the density at any point is less than or equal to the jam density). Next set of equations provides estimates for three model c-constants [18] based on the roadways mean free-flow speed (u_f), speed at capacity (u_c), capacity flow (q_c), and jam density (k_j). The final set of constraints provides a valid search window for four traffic stream parameters that are being optimized (u_f, u_c, q_c, and k_j). All details can be found in paper [16].

The primary goal for using this stream model calibration to the field data sets is to fairly compare the real data with results obtained from the calibrated simulation model. So it is possible to get all three fundamental traffic diagrams, which is very important, because a good fit in one domain (e.g. speed-flow) does not necessarily imply a good fit in another domain (e.g. speed-density) [16].

III. CA BASED TRAFFIC SIMULATION MODEL

The first highly suitable CA model of single-lane freeway traffic was introduced by Nagel and Schreckenberg in 1992 [19]. Because of simplicity of this model, it was possible to perform millions of updates in a second [20]. Hence, the model can be used for simulating a high volume of traffic over very large networks. Due to some critics of unrealistic behavior (such as [21]), simple CA based models are often extended. On the other hand, these models were shown to be able to capture all basic phenomena that occur in traffic flows [22], not only in the field of vehicular traffic flow modeling, but also in other fields such as pedestrian behavior, escape and panic dynamics, etc.

Nagels and Schreckenbergs traffic model was initially defined on a one-dimensional array with open or periodic boundary conditions and with every single cell representing a road segment [19]. A local transition function (a rule) defines the new state of a cell on the basis of its current state and the state of neighboring cells. Globally viewed, it describes the movements of vehicles from one cell to another cell in a discrete way. In space domain, each cell represents only a defined length of road segment, so space is coarse-grained. This coarse graininess is fundamentally different from the usual microscopic models, which adopt a semi-continuous space. The length of 7.5 m was chosen in the original model because each car occupies about this amount of space in a complete jam [19]. In other words, this is the average vehicle length on the road, including a constant gap in front of and/or behind each car. In order to model other kinds of vehicles in CA based models, it is better to use different cell lengths, or even more precisely, use more smaller cells for representing a single vehicle – especially for bigger types of vehicles (e.g. [23]).

Each CA simulation step represents an amount of time in reality. In the CA model this value is crucial because together with the cell length it gives us the granularity of minimal vehicle speed jumps. This value also represents a minimal time for any responds, e.g. it is also the driver reaction time. Based on suggestions given in [24] and field data measurements [23], it is reasonable to consider one such simulation step as 0.6 – 1.5 s.

Normally, urban traffic road networks are very complex. It was shown that arbitrary kinds of road and intersections can be reduced to only a few basic elements. For constructing more complicated networks, we shall simply connect more various kinds of cells to the desired topology. In this way it is possible to build more traffic lanes, roundabounds or very complex topologies for whole cities [6], [5].

A. *Local transition function*

Properties of a single lane traffic are modeled on the basis of integer-valued probabilistic cellular automaton rules. The local transition function can be formalized as follows: each vehicle (i) in a cell has an integer velocity $v_v(i)$ with values between zero and $v_{max}(i)$. We let $gap(i)$ denote the cell gap in front of the vehicle i – i.e. a leader to the follower distance. For an arbitrary configuration, one update of the simulation system consists of four consecutive steps, which are performed in parallel for all vehicles. The original procedure consists of (1) acceleration (every vehicle tends to accelerate), (2) slowing down (i.e. collision avoidance), (3) randomization with a given probability and finally (4) vehicle motion, where each vehicle is advanced based on a new computed speed $v_v(i)$. It is important to note, that this simple CA based traffic model shows nontrivial and realistic behavior. The randomization step is essential in simulating traffic flows since the model dynamics is completely deterministic, and without this randomness, every initial configuration of vehicles and corresponding velocities reaches a stationary pattern which is shifted backwards. This simple model is also capable of reproducing characteristic properties of real traffic, such as certain aspects of flow-density relation, spatio-temporal evolution of jams, stop-and-go waves, etc. [25].

B. *Updated local transition function*

In our previous work [8], we updated the original function briefly described in previous section to a new form, where some brand new parameters can be found. The traffic simulation model is extended to eliminate unwanted properties of ordinary CA based models, such as stopping from maximum velocity to zero in one time step [8]. This is possible due to storing the previous (or the leading) vehicle velocity $v_{prev}(i+1)$. When there is such vehicle, the following vehicle (i) is able to determine its positive or negative acceleration with $acc(i+1)$. According to Alg. 1, it is

Algorithm 1 Updated local transition function.

if $v_v(i) < p_4$ and $v_v(i) < v_{max}(i)$ **then**
    $v_v(i) := v_v(i) + 1$ with probability p_7
end if
if $(gap(i) + acc(i+1)) > v_v(i)$ **then**
    if $v_v(i) < p_6$ **then**
        $v_v(i) := v_v(i) - 1$ with probability p_5
    else
        $v_v(i) := v_v(i) - 1$ with probability p_8
    end if
else
    if $acc(i+1) > 0$ **then**
        $v_v(i) := 1/p_9 \times (gap(i) + acc(i+1))$
    else
        $v_v(i) := 1/p_{10} \times (gap(i) + acc(i+1))$
    end if
end if

Ensure: Each vehicle at sites i is advanced $v_v(i)$ times and $v_{prev}(i) := v_v(i)$.

firstly determined, if investigated vehicle could accelerate (i.e. vehicle velocity $v_v(i)$ is not greater than maximal vehicles speed p_4 or given vehicle velocity limit $v_{max}(i)$). If so, its speed-up is accomplished with probability p_7, so not all vehicles tend to always accelerate as in the original model [19]. Then, if there is a plenty of room for vehicle to get in (i.e. $gap(i)+acc(i+1) > v_v(i)$) or there is no previous vehicle in the same lane, collision avoidance mechanism is not performed. Similarly to the original CA local transition function, only deceleration based on probabilities could be applied in this situation. In case of small vehicle velocities ($v_v(i) < p_6$), deceleration is performed with probability p_5, otherwise ($v_v(i) > p_6$) with probability p_8.

Collision avoidance occurs when there is no free room for vehicle i in the same lane to get in (i.e. $gap(i) + acc(i+1) <= v_v(i)$). Two situations may occur. If the leading vehicle tends to accelerate ($acc(i+1) > 0$), the actual vehicle velocity $v_v(i)$ is reduced to $1/p_9 \times (gap(i) + acc(i+1)$. Otherwise, ($acc(i+1) <= 0$), actual vehicle speed $v_v(i)$ is reduced to $1/p_{10} \times (gap(i) + acc(i+1)$. It can be seen that these two parameters are more driver-based parameters than model oriented. We will try to find out if these ones could be determined statistically for a given road segment. Finally, each vehicle is advanced $v_v(i)$ sites and vehicle velocity updates must be also performed.

IV. OPTIMIZATION OF THE CA BASED MODEL

In this work, genetic algorithm (GA) is used to find the most useful parameters of the CA model in order to maximize the precision of the traffic simulator. The main idea of GA is to evolve a population (set) of candidate solutions to find better ones [26]. Such candidate solution is encoded as a chromosome which is an abstract representation that can be modified with genetic operators (e.g. mutation and crossover, etc.).

A. Parameters encoding

In order to simplify GA, all simulation model parameters, which will be optimized, are encoded in binary form. In case of real numbers from a given interval (e.g. $[1, 0]$), the interval is divided into the N pieces of the same size. The value N depends on the number of bits used for encoding of the parameter.

Using a 6-bit value and the minimal length of the cell $0.125\ m$ the maximal cell length is $8\ m$ (64×0.125). The cell length is the first model parameter – p_1. One vehicle always occupies as many such cells as it fits into the $5.5\ m$ (or nearer, but not smaller). For example, for the smallest cell length ($0.125\ m$) it is exactly 44 cells. Bigger vehicles, such as trucks, occupy only two times bigger place ($11\ m$). The second model parameter, p_2, is the simulation step or also the reaction time with the minimal value of 0.05 and maximum value of 3.2 seconds encoded again using 6 bits. The cell neighbor, p_3, is encoded using 12 bits (e.g. when the cell length is at minimum then the maximum neighbor is $0.125 \times 4096 = 512\ m$). The next parameter is maximal vehicles speed p_4 (encoded on 11 bits, i.e. 2048 possible values for a chosen reaction time and cell lenght) giving, as in the original model, the number of cells per simulation step. The probability of slowing down is represented by p_5 (encoded on 8 bits) and slow speed boundary is encoded as p_6 ($1 - 512$ cells per simulation step on 9 bits). Then, the speed-up probability is denoted as p_7. The parameter p_8 is probability of vehicles slowing down in case of a vehicle speed greater than the slow speed p_6. Further model constants p_9 and p_{10} are coefficients of vehicle approximation in case of previous vehicle acceleration and previous vehicle slowing-down. Both parameters have minimal value of 1 and maximal value of 32 (encoded on 5 bits). All parameters with their respective minimal values, maximal values and step, are briefly summarized in Tab. I.

B. Chromosome

The proposed GA has an auto-evolution or also self-adaptation capability, which means that parameters of the algorithm (the probability of mutation p_m and crossover p_c) are also part of the chromosome. Hence the user is not forced to set them. The whole set of parameters is represented using one 92-bit number. It is important to note that each parameter of the chromosome is encoded using *Gray encoding* to ensure that the maximal Hamming distance between two successive values is only one. This setup does not allow big jumps between values in case of a single bit change. The

TABLE I: CA model parameters and values.

	Bits used [#]	Min. value	Max. value	Step
p_1	6	0.125	8.000	0.125
p_2	6	0.05	3.20	0.05
p_3	12	p_1	$2^{12} \times p_1$	p_1
p_4	11	p_1/p_2	$2^{11} \times p_1/p_2$	p_1/p_2
p_5	8	0.00392	1.00000	0.00392
p_6	9	p_1/p_2	$2^9 \times p_1/p_2$	p_1/p_2
p_7	8	0.00392	1.00000	0.00392
p_8	8	0.00392	1.00000	0.00392
p_9	5	1	32	1
p_{10}	5	1	32	1
p_m	10	0.00097	1.00000	0.00097
p_c	4	0.06667	1.00000	0.06667

first population ($X(0)$) consists of 60 such chromosomes ($|X(0)| = 60$) generated randomly.

C. Fitness function

All chromosomes from population X_i are separately evaluated using the same fitness function. Firstly, a candidate CA model is constructed using the parameters obtained from a candidate chromosome. Then simulation is performed for the given model. Because field data usually consists of various scenarios over different traffic speeds, densities and flows, it is possible to initialize the new model with *random data* inputs (e.g. random times of vehicle arrivals) to gather different flow/density/speed situations from calibrated model. Note, that this random data generation could be notably simplified, if the average incoming vehicle speed and also the average incoming times with their respective variability are enumerated from given field data set. Whole measurement procedure (from the simulation model) is performed in same way as measurements from the field, which will be described later on. Depending on the facility type, also various vehicle types are generated where possible. Whole simulation is executed until the same number of samples (flow vs. speed and flow vs. density) as the number of field data samples is reached (see Tab. II). After that, the fitness function $F(x)$ is calculated as a sum of error function $E(x)$ (shown in Eq. 2) and the penalty function $P(x)$, where x represents a candidate solution. Due to noticeable slower simulation runtime for solutions where the cell length is very small, the fitness function must be adjusted with the penalization function $P(x)$ as showh in Eq. 3.

$$P(x) = 10/cel\_length(x) \tag{3}$$

This penalization ensures that the solutions with smaller cell lengths will not be preferred. Moreover, this penalization is multiplied three times (the error function is calculated for three different variables separately) and also multiplied by the number of samples (to add a constant error to every sample) as shown in Eq. 4.

$$F(x) = E(x) + (3 \times number\_of\_samples(x) \times P(x)) \tag{4}$$

Finally, GA tries to minimize the fitness function $F(x)$, so better solutions are always with lower fitness value.

D. Creating a new population

1) Selection: After evaluation of all chromosomes from the population $X(i)$ is complete, some of them are selected for next operations using a tournament selection with base 2 giving a new population $X_S(i)$, where $|X_S(i)| = 30$.

2) Crossover: Two-point crossover is applied between two randomly selected individuals giving a new set $X_C(i)$ (where $X_C(i) \subset X_S(i)$ and $|X_C(i)| = 30$). The first point of crossover operation is between the p_3 and p_4 parameter and the second one right after p_{10} parameter, to allow alternation of the model and the GA parameters individually. This operator is applied with the average probability calculated from two chosen chromosomes (p_c).

3) Mutation: On all chromosomes from $X_C(i)$ a mutation operator (i.e. changing bit $0 \to 1$ or $1 \to 0$) is applied with the probability (p_m) taken from evaluated individual, which gives a brand new population $X_M(i)$ of the same size.

4) Population recovery: Finally, a new population of 60 individuals $X(i+1)$ is selected from the previous population $X(i)$ and the $X_M(i)$ population. This ensures that the best solution will always survive (i.e. elitism is present) [26].

Described GA procedure is repeated until 125 000 generations are exhausted as shown in Alg. 2.

Algorithm 2 Genetic algorithm procedure.

$i = 0$
Generate population $X(i)$ randomly, $|X(i)| = 60$
Evaluate all candidates from $X(i)$ with $F(x)$
repeat
1. Create $X_S(i)$ using tournament selection from $X(i)$
2. Create $X_C(i)$ using crossover operator on $X_S(i)$
3. Create $X_M(i)$ using mutation operator on $X_C(i)$
4. Evaluate all candidates from $X_M(i)$ with $F(x)$
5. Create $X(i+1)$ by selecting 60 best individuals from $X_M(i) \cup X(i)$
6. $i := i + 1$
until $(i <= 125000)$

V. EXPERIMENTAL RESULTS

A. Field data

In order to evaluate the proposed method, field data from different facility types have been utilized. These macroscopic data sets consist of data captured by our industrial partners in year 2010.

The first data set (a) is a bit crocked road segment between two bigger villages in the Slovak Republic with a maximum allowed speed of 50 km/h. The particular segment is on the way to the country seat, so the road is utilized mostly by drivers going to work and back on ordinary business days, but traffic is not strictly homogenous here. This road segment is also a part of the route between two biggest cities in the region and statistically given 5% of traffic comes from bigger vehicles (e.g. busses, trucks, etc.).

The second data set (b) comes from two climb-lanes on the freeway road segment between two biggest cities in

TABLE II: Number of field data samples for each data set.

Data set	Usable samples [#]	Empty samples [%]
(a)	31157	11.08
(b)	32268	7.91
(c)	29396	16.11

the Czech Republic. There is a speed limit of 130 km/h and compared to previous data set (a), vehicles are less constrained by the roadway geometry and/or friction, but they are more constrained by regulatory conditions (e.g. speed limit). Field data tend to demonstrate a fairly linear increase in speed as a function of the distance headway. The traffic is not homogenous, but it is important to note, that there is also a vehicle classification system installed here, so it was possible to classify various types of vehicles. The precision of classification is 88.5%.

The last data set (c) comes from a short and (compared to previous ones) small road segment in the city of Prague bordered from both sides with traffic lights. All the days, there is a heavy traffic with high probability of traffic jams. This facility is with no admission for heavy and/or long vehicles. Motorbikes do not often appear here. The maximum speed on this segment is restricted to 80 km/h.

All three data sets (a, b and c) were obtained using standard surveillance technologies and measured using detection loops and detection cameras (used as a supplying secondary or correlation source of data) for every day and night of the year on each facility. Therefore it was possible to measure macroscopic variables (q, u) across a certain time interval. The third macroscopic value (k), can always be calculated from the relation of the traffic flow theory (see Eq. 1). The discrete nature of traffic requires capturing time intervals of a least half a minute if we want to achieve meaningful information. Also, when time intervals exceed the duration of five minutes, certain dynamic characteristics are lost as stated in [15]. Based on this, we decided to divide every day of the year to 96 time intervals of 15 minutes each. These intervals are then divided into the two parts. First part (5 minutes) is a measurement time interval where the field data are used. Second interval (10 minutes) is thrown away. Thus we got 35040 samples of flow versus vehicle speed relation samples during the measurement interval on each facility during a year. If there is an empty one (i.e. no vehicle occurred), a sample from a given data set is deleted. The final number of usable field data samples as well as percentage of empty samples from original 35040 samples is given in the Table II.

B. Calibrated models

All parameters of the CA based microscopic traffic simulation model ($p_1 \dots p_{10}, p_m$ and p_c), which were evolved for all three data sets $(a), (b)$ and (c) separately, are shown decoded as real values in Tab. III. All comes from the best solution of the last generation of GA. Tab. III also shows parameters of our previously manually tuned and updated CA model as introduced in [8] and [9]. Some of those manually

TABLE III: Parameters and per point errors for updated model and models evolved for each data set (a, b and c).

	Upd. model	Model for a)	Model for b)	Model for c)
p_1	5.500 m	2.625 m	7.125 m	2.250 m
p_2	1.20 s	1.35 s	1.50 s	0.65 s
p_3	60.5 m	231 m	441.75 m	69.75 m
p_4	181.5 $\frac{km}{h}$	84.00 $\frac{km}{h}$	153.9 $\frac{km}{h}$	87.23 $\frac{km}{h}$
p_5	0.3000	0.2902	0.1012	0.9098
p_6	181.5 $\frac{km}{h}$	28.00 $\frac{km}{h}$	51.3 $\frac{km}{h}$	74.77 $\frac{km}{h}$
p_7	1.0000	0.8471	0.9294	0.6392
p_8	n/a	0.2863	0.0431	0.7294
p_9	12	2	8	11
p_{10}	12	3	5	9
p_m	n/a	0.00293	0.02248	0.00879
p_c	n/a	0.66667	0.40000	0.20000
$E_p(a)$	31.81%	**3.24%**	29.01%	24.78%
$E_p(b)$	12.69%	13.96%	**2.78%**	14.8%
$E_p(c)$	25.82%	22.98%	17.80%	**4.02%**

updated values, are generally not available (GA parameters) or have a bit different meaning in our previous model. Such an example is the low speed boundary value p_6, which is identical with maximal vehicles speed p_4. This is caused by absence of the first parameter in updated model, because slowing down was performed for all available vehicles (with probability p_5). Also all vehicles in the updated model tend to always accelerate, so $p_7 = 1.0$.

Tab. III also shows the quantified error $E_p(x)$ (see Eq. 5)

$$E_p(x) = E(x)/(3 \times number\_of\_samples(x)) \qquad (5)$$

for a single sample point and for a given data set x. We also measured this error for our manually updated model with additionam maximal speed adjustment for given datasets (in the first column). It is very important to note, that some error is not bad at all, because this error is enumerated on the regressed function (e.g. some sample point could be out of the scope of this function). It can be seen that all three new calibrated models, which were obtained using our GA, are significantly better on a particular data set in comparison to our manually updated model. Moreover, all new models are also better when compared to different data sets. The only exception is the result for data set (b) which is probably due to the original purpose of updated model (made for simulating freeways). It is also remarkable in the cell length parameter (p_1), which has the biggest value meaning the biggest distance to previous vehicle in case of a complete jam. On the other side, the smallest cell length was evolved for (c), so a single vehicle (and the corresponding gap) must be modeled with two cells representing 5.5 meters. The same holds in the model for (a), where this length is 5.25 meters. In order to check whether this value is not only a result of stochastic nature of GA, Fig. 1 shows the evolution of this parameter during 125 000 generations as an average for 100 independent runs of GA. Nevertheless, it can clearly be seen that this parameter tends to converge to one particular value in all three data sets. A similar test was performed for every one evolved parameter, but due lack of space we do not illustrate them here.

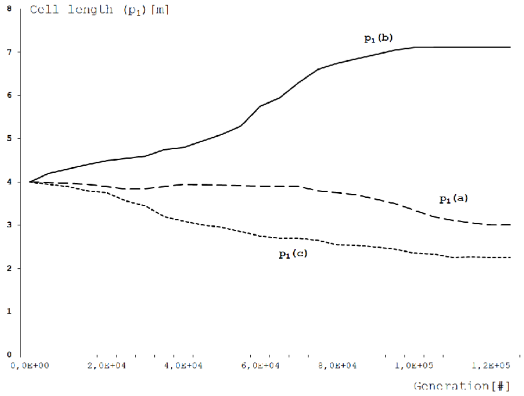

Fig. 1: Evolution of parameter p_1 in 125 000 generations for data set $(a), (b)$ and (c).

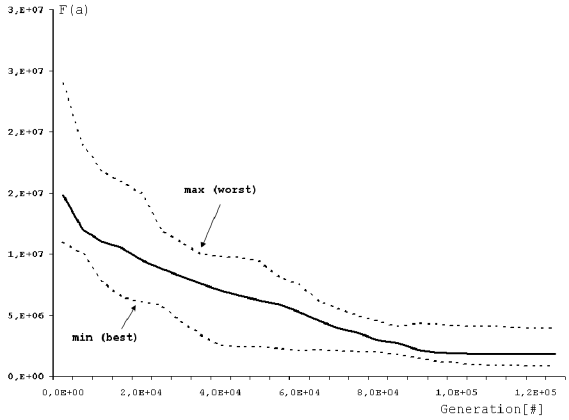

Fig. 2: Fitness $F(a)$ in all generations as an average value out of 100 independent runs with its the best and worst values when calibrating to data set (a).

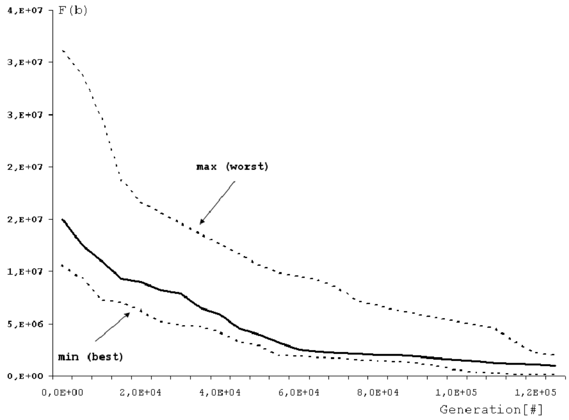

Fig. 3: Fitness $F(b)$ in all generations as an average value out of 100 independent runs with its the best and worst values when calibrating to data set (b).

The reaction time (p_2) also corresponds to what it could be expected when looking closer at the field data. The calibrated model for the second data set has again the biggest value of 1.5 seconds, that corresponds to the minimal increment of speed which is 17.1 km/h (p_1/p_2 as seen in Tab. I) and this is the biggest increment for all data sets. Surprisingly, a very big increment value could be seen also for the model calibrated to the third data set (12.46 km/h) where the data comes from often jammed short road segment, where it was thought that too big speed increments should not be supposed.

The smallest cell neighbor (p_3) was evolved for the model calibrated to (c) (i.e. 31 cells). On the other side, the biggest cell neighbor (62 cells) can be seen in model calibrated to (b). This also corresponds to nature of field data. Next parameter, the maximum allowed speed (p_4), is for all models higher than local speed restrictions as we have supposed.

For the model calibrated to (a), the probability of slowing down (p_5) is 0.2902 for vehicle speeds lower than the evolved boundary (p_6) of 28.00 km/h. The same slowing down (in case of speed lower than 51.3 km/h) occurs for 10.12% in the model calibrated to (b) and finally, up to 90.98% of vehicles are slowing down in case of their speed is lower than 74.77 km/h in the last model calibrated to (c). On the other hand, the probability of acceleration (p_7) is the highest for the second model (0.9294), then for the model calibrated to (a) (0.8471) and the lowest probability (0.6392) is in the last model. The parameter of slowing down (p_8) in case of speeds greater than the evolved boundary speed has an opposite sequence as the previous one. This could indicate that it would be possible to interoperate both of these parameters.

Parameters p_9 and p_{10} are surprisingly quite small. However, based on their convergence tests, we claim that these parameters (i.e. driver sensitivity) can be also statistically obtained for a desired road segment.

The CA traffic simulation model for the last data set was relatively easy to find which can be seen in the fastest fitness convergence. However, this model exhibits the biggest error. Models for other two sets were evolved after a longer time. It is also important to note, that completing all runs for one data set (100 runs of 125 000 generations) takes more than four days running at *Intel Xeon CPU5420 @ 2.5 GHz*. Fig. 2, Fig. 3 and Fig. 4 show the average fitness value for 100 successive runs for each data set. It can be seen, that it tends to decrease during evolution which is ensured by elitism. In these figures one can also see the best solution and the worst solution during the evolution as an average of 100 independent runs. After 125 000 generations, the quality of population is not changing dramatically. Our genetic algorithm was tuned to always find a reasonable solution after this number of generations. The whole tuning process will be described in the forthcoming paper.

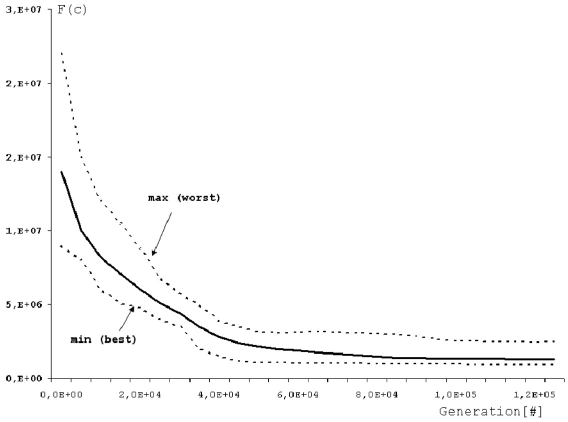

Fig. 4: Fitness $F(c)$ in all generations as an average value out of 100 independent runs with its the best and worst values when calibrating to data set (c).

VI. CONCLUSIONS

In this paper, we proposed an effective calibration method for a simple microscopic traffic simulation model. The proposed model is based on the cellular automaton, which can easily be accelerated. We utilized evolutionary approach, in particular genetic algorithm, that was able to find suitable parameters of the CA model for a given field data. For those test road segments, we increased the precision of simulator by 20.09% in average in comparison with a manually updated and tuned model. The proposed methods seem to be promising in calibration of pre-selected road segments of interest. Moreover, with this process it is possible to readjust the model to given field data which could come from standard surveillance technologies such as loop detectors in our case.

In our future work, we would like to analyze different calibrated models in greater details. For example, to compare the distribution of lead to follower distances during the time against the field data. This could be the first step to precisely predict the future traffic states or travel time. Also, it would be very interesting to derive how much data has to be used for a proper model calibration in the case when a sufficient amount of data is not available.

REFERENCES

[1] S. Maerivoet and B. De Moore, Transportation planning and traffic flow models, Technical Report No. 05-155, Katholieke Universiteit Leuven, 2005.

[2] M. Brackstone and M. McDonald, Car-following: a historical review, in Transportation Research Part F: Traffic Psychology and Behaviour, vol. 2, no. 4, Elsevier, 1999, pp. 181-196.

[3] J. L. Tripp, H. S. Morveit, A. A. Hansson, and M. Gokhale, Metropolitan road traffic simulation on FPGAs, in 13th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'05), 2005, pp.117-126.

[4] K. Nagel and D. Strippgen, Using common graphics hardware for multi-agent traffic simulation with CUDA, in 2nd International Conference on Simulation Tools and Techniques (Simutools '09), ICST 2009, Belgium, 2009, pp. 1-8, ISBN: 978-963-9799-45-5.

[5] R. Barlovic, J. Esser, K. Froese, W. Knospe, L. Neubert, M. Schreckenberg, and J. Wahle, Online traffic simulation with cellular automata, in Traffic and Mobility: SimulationEconomics Environment, 1999, pp. 117-134.

[6] A. Dupuis and B. Chopard, Cellular automata simulations of traffic: A Model for the city of Geneva, in Networks and Spatial Economics, vol. 3, no. 1, Springer Netherlands, 2003, pp. 9-21.

[7] G. H. Bham and R. F. Benekohal, A high fidelity traffic simulation model based on cellular automata and car-following concepts, in Transportation Research Part C: Emerging Technologies, vol. 12, no. 1, Elsevier, 2004, pp. 1-32.

[8] P. Korcek, L. Sekanina, and O. Fucik, A Scalable Cellular Automata Based Microscopic Traffic Simulation, in IEEE Intelligent Vehicles Symposium 2011 (IV11), Baden-Baden, DE, IEEE ITSS, 2011, pp. 13-18, ISBN 978-1-4577-0889-3.

[9] P. Korcek, L. Sekanina, and O. Fucik, Cellular automata based traffic simulation accelerated on GPU, in 17th International Conference on Soft Computing (MENDEL2011), Brno, CZ, UAI FSI VUT, 2011, pp. 395-402, ISBN 978-80-214-4302-0

[10] B. R. Hellinga, Requirement for the Calibration of Traffic Simulation Models. Department of Civil Engineering, University of Waterloo.

[11] R. Dowling, et. al., Guidelines for Calibration of Microsimulation Models: Framework and Applications, in Transportation Research Record: Journal of the Transportation Research Board, Transportation Research Board of the National Academies, vol. 1876, 2004, pp. 1-9, ISSN: 0361-1981.

[12] V. Punzo and F. Simonelli, Analysis and Comparison of Microscopic Traffic Flow Models with Real Traffic Microscopic Data, in Transportation Research Record: Journal of the Transportation Research Board, Transportation Research Board of the National Academies, vol. 1934, 2005, pp. 53-63, ISSN: 0361-1981.

[13] M. Van Aerde, H. Rakha, Multivariate Calibration of Single Regime Speed-Flow-Density Relationships, in Vehicle Navigation and Information Systems (VNIS) conference, Seattle, 1995.

[14] D. Gazis, R. Herman and R. Rothery, Nonlinear follow-the-lead models of traffic flow. Operations Research, 1961. pp. 545-567.

[15] L. H. Immers and S. Logghe, Traffic Flow Theory, Katholieke Universiteit Lueven - Faculty of Engineering, Department of Civil Engineering, Section Traffic and Infrastructure, Course H111, en. version, 2002.

[16] H. Rakha and Y. Gao, Calibration of steady-state car-following models using macroscopic loop detector data, Final Report VT-2008-01, Virginia Tech Transportation Institute, 2010.

[17] H. Rakha and M. Arafeh, Tool for calibrating steady-state traffic stream and car-following models. in Transportation Research Board Annual Meeting, Washington, D.C, 2007.

[18] S. H. Demarchi, A new formulation for Van Aerde's speed-flow-density relationship, in XVI Congresso De Pequisa e Ensino em Transportes, 2002, Natal, Brazil.

[19] K. Nagel and M. Schreckenberg, A cellular automaton model for freeway traffic, in Journal de Physique I, vol. 2, Issue 12, 1992, pp.2221-2229.

[20] K. Nagel, High-speed microsimulations of traffic flow, Ph.D. Thesis, University of Cologne, 1995

[21] P. Chakroborty and A. K. Maurya, Microscopic analysis of cellular automata based traffic flow models and an improved model, in Transport Reviews, vol. 28, issue 6, 2008, pp. 717-734.

[22] D. Chowdhury, L. Santen and A. Schadschneider, Statistical physics of vehicular traffic and some related systems, in Physics Reports 329, 2000, pp. 199-329.

[23] G. H. Bham and R. F. Benekohal, A high fidelity traffic simulation model based on cellular automata and car-following concepts, in Transportation Research Part C: Emerging Technologies, vol. 12, no. 1, Elsevier, 2004, pp. 1-32.

[24] M. F. Aycin and R. F. Benekohal, Comparison of car-following models for simulation, Transportation Research Record, vol. 1678, Washington, 1999, pp. 116-127.

[25] D. E. Wolf, Cellular automata for traffic simulations, in Physica A, vol. 263, 1999, pp. 438-451.

[26] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley, 1989, ISBN: 0201157675.

Paper IV

Calibrating Traffic Simulation Model Using Vehicle Travel Times.

Calibration of Traffic Simulation Models Using Vehicle Travel Times

Pavol Korcek, Lukas Sekanina, and Otto Fucik

Brno University of Technology
Faculty of Information Technology
IT4Innovations Centre of Excellence
Bozetechova 1/2, 612 66 Brno, Czech Republic
{ikorcek,sekanina,fucik}@fit.vutbr.cz
http://www.fit.vutbr.cz

Abstract. In this paper, we propose an effective calibration method of the cellular automaton based microscopic traffic simulation model. We have shown that by utilizing a genetic algorithm it is possible to optimize various model parameters much better than a human expert. Quality of the new model has been shown in task of travel time estimation. We increased precision by more than *25 %* with regard to a manually tuned model. Moreover, we were able to calibrate some model parameters such as driver sensitivity that are extremely difficult to calibrate as relevant data can not be measured using standard monitoring technologies.

Keywords: traffic, simulation, cellular automaton model, calibration, travel time.

1 Introduction

A very important stage of development of any traffic model is its comparison with reality, namely calibration and validation. In [1], authors proposed an effective three-step process for the microscopic traffic model calibration. Another paper [2] gives some basic guidelines for calibration of microscopic simulation models in form of framework and applications. The developers usually calibrate and validate the model on their own using some data sets that they have access to and publish the results obtained. For example, in paper [3] authors tried to perform a simple calibration of ten microscopic traffic simulation models in a way that the models were calibrated and compared to each other with the GPS based field data from year 2004 in Japan. But it should be noted, that in almost all previous calibration approaches, some real data are desired in a form, which is generally not available. It was shown that it is important to find a few basic parameters for the model calibration [4]. Namely a driver sensitivity (e.g. reaction time), a jam density headway and free-speed (maximum speed when vehicle is not constrained) have to be determined. It was also stated that this process is neither a straight-forward nor an easy task. For example, while the free-speed is relatively easy to estimate in the field and generally lies between the

G.C. Sirakoulis and S. Bandini (Eds.): ACRI 2012, LNCS 7495, pp. 807–816, 2012.

speed limit and the design speed of the roadway, the jam density headway is more difficult to calibrate but typically ranges between 110 to 150 vehicles/km/lane. The driver sensitivity factor is extremely difficult to calibrate because it can not be measured using standard monitoring technologies (e.g. detection loops that work on magnetic-induction principle).

In this work we propose to utilize our cellular automaton (CA) based microscopic traffic simulation model, which was shown not only to be extremely fast to achieve multiple in real-time simulations (e.g. [6]), but also updated to eliminate unwanted properties of ordinary CA based models. The quality of this updated model has been previously evaluated by comparison with *Van Aerde* fundamental diagram [5]. Then we will also try to calibrate parameters of this model to field data that can be obtained from standard monitoring technologies. We will show, that it is possible to achieve a better precision on travel time estimation for a given road segment. Moreover, except CA model parameters, we will also optimize some parameters such as driver sensitivity which, as stated for example in [4], are extremely difficult to calibrate with other common techniques. The optimization/calibration will be performed by genetic algorithm (GA).

The rest of the paper is organized as follows. Section 2 introduces an updated cellular automaton based traffic simulation model. Then, in Section 3, the process of optimization of the model with selected GA is described in detail with all simulation model parameters. Experimental evaluations for our field data sets are then presented in Section 4. Finally, conclusions and suggestions for future work are given in the last Section 5.

2 Updated Local Transition Function

In our previous work [5], we updated the original local transition function [7] to a new form, where some brand new parameters can be found. The traffic simulation model is extended to eliminate unwanted properties of ordinary CA based models, such as stopping from maximum vehicle speed to zero in one time step. This is possible due to storing the previous (or the leading) vehicle velocity $v_v(i+1)$. When there is such vehicle, the following vehicle (i) is able to determine its positive or negative acceleration with $acc(i+1)$. According to Alg. 1, it is firstly determined, if investigated vehicle could accelerate (i.e. vehicle velocity $v_v(i)$ is not greater than maximal vehicles speed p_4 or given vehicle speed limit $v_{max}(i)$). If so, its speed-up is accomplished with probability p_7, so not all vehicles tend to always accelerate as in the original model [7]. Then, if there is a plenty of room for vehicle to get in (i.e. $gap(i) + acc(i+1) > v_v(i)$) or there is no previous vehicle in the same lane, collision avoidance mechanism is not performed. Similarly to the original CA local transition function, only deceleration based on probabilities could be applied in this situation. In case of small vehicle speeds ($v_v(i) < p_6$), deceleration is performed with probability p_5, otherwise ($v_v(i) > p_6$) with probability p_8.

Collision avoidance occurs only when there is no free room for vehicle i in the same lane to get in (i.e. $gap(i) + acc(i+1) <= v_v(i)$). Two basic situations may

Algorithm 1. Updated local transition function

if $v_v(i) < p_4$ and $v_v(i) < v_{max}(i)$ **then**
  $v_v(i) := v_v(i) + 1$ with probability p_7
end if
if $(gap(i) + acc(i + 1)) > v_v(i)$ **then**
  if $v_v(i) < p_6$ **then**
    $v_v(i) := v_v(i) - 1$ with probability p_5
  else
    $v_v(i) := v_v(i) - 1$ with probability p_8
  end if
else
  if $acc(i + 1) > 0$ **then**
    $v_v(i) := 1/p_9 \times (gap(i) + acc(i + 1))$
  else
    $v_v(i) := 1/p_{10} \times (gap(i) + acc(i + 1))$
  end if
end if

Ensure: Each vehicle i is advanced $v_v(i)$ times and $v_{prev}(i) := v_v(i)$.

occur. If the leading vehicle tends to accelerate $(acc(i+1) > 0)$, the actual vehicle speed $v_v(i)$ is reduced to $1/p_9 \times (gap(i)+acc(i+1)$. Otherwise, $(v_v(i+1) <= 0)$, actual vehicle speed $v_v(i)$ is reduced more strictly to $1/p_{10} \times (gap(i)+acc(i+1)$. It can be seen that these two parameters are more driver-based parameters than model oriented. We will try to find out if these ones could be determined statistically for a given road segment. Finally, each vehicle is advanced $v_v(i)$ sites and velocity updates must be also performed.

3 Optimization of the Model

Genetic algorithms (GA) are widely used in various areas of science and engineering to find solutions to optimization and design problems [8]. The main idea is to evolve a population (set) of candidate solutions to find better ones. A candidate solution is encoded as a chromosome which is an abstract representation that can be modified with standard genetic operators such as mutation and crossover. In this work, GA is used to find all parameters of the CA model in order to maximize the precision of the traffic simulator.

3.1 Parameters Encoding

In order to simplify GA, all simulation model parameters, which will be optimized, are encoded in binary form. In case of real numbers from a given interval (e.g. $[1, 0]$), the interval is divided into the N pieces of the same size. The value N depends on the number of bits used for encoding of the parameter.

Using a 6-bit value and the minimal length of the cell 0.125 m the maximal cell length is 8 m (64×0.125). The cell length is the first model parameter – p_1. One vehicle always occupies as many such cells as it fits into the 5.5 m (or nearer, but not smaller). For example, for the smallest cell length (0.125 m) it is exactly 44 cells. Bigger vehicles, such as trucks, occupy only two times bigger place (11 m). The second model parameter, p_2, is the simulation time-step (also the reaction time) with the minimal value of 0.05 and maximum value of 3.2 seconds encoded again using 6 bits. The cell neighbor, p_3, is encoded using 12 bits (e.g. when the cell length is at minimum then the maximum neighbor is $0.125 \times 4096 = 512$ m). The next parameter is maximal vehicles speed p_4 (encoded on 11 bits, i.e. 2048 possible values for a chosen reaction time and cell lenght) giving, as in the original model [7], the number of cells per simulation step. The probability of slowing down is represented by p_5 (encoded on 8 bits) and slow speed boundary is encoded as p_6 ($1 - 512$ cells per simulation step on 9 bits). Then, the speed-up probability is denoted as p_7. The parameter p_8 is probability of vehicles slowing down in case of a vehicle speed greater than the slow speed p_6. Further model constants p_9 and p_{10} are coefficients of vehicle approximation in case of previous vehicle acceleration and previous vehicle slowing-down. Both parameters have minimal value of 1 and maximal value of 32 (encoded on 5 bits). All parameters with their respective minimal values, maximal values and step, are briefly summarized in Tab. 1.

Table 1. CA model parameters and values

	Bits used [#]	Min. value	Max. value	Step
p_1	6	0.125	8.000	0.125
p_2	6	0.05	3.20	0.05
p_3	12	p_1	$2^{12} \times p_1$	p_1
p_4	11	p_1/p_2	$2^{11} \times p_1/p_2$	p_1/p_2
p_5	8	0.00392	1.00000	0.00392
p_6	9	p_1/p_2	$2^9 \times p_1/p_2$	p_1/p_2
p_7	8	0.00392	1.00000	0.00392
p_8	8	0.00392	1.00000	0.00392
p_9	5	1	32	1
p_{10}	5	1	32	1
p_m	10	0.00097	1.00000	0.00097
p_c	4	0.06667	1.00000	0.06667

3.2 Chromosome

The proposed GA has an auto-evolution or also self-adaptation capability, which means that parameters of the algorithm (the probability of mutation p_m and crossover p_c) are also part of the chromosome. Hence the user is not forced to set them. The whole set of parameters is represented using one 92-bit number. It is important to note that each parameter of the chromosome is encoded using *Gray* *encoding* to ensure that the maximal Hamming distance between two successive values is only one. This setup does not allow big jumps between values in case of a single bit change. The first population ($X(0)$) consists of 60 such chromosomes ($|X(0)| = 60$) generated randomly.

3.3 Fitness Function

All chromosomes from population X_i are separately evaluated using the same fitness function. Firstly, a candidate CA road segment is constructed using the parameters obtained from a candidate chromosome. Then simulation is performed for that model. Incoming vehicles are generated depending on their time of arrival based on measurements from the field. Vehicles outgoing from the simulated road segment are simply removed and their travel time is recorded. Depending on the facility type, various vehicle types could be generated where possible. Whole simulation is executed until the same number of simulated vehicles as the number of vehicles in the field data is reached (N_x). After that, the fitness function F (see Eq. 3) is calculated as a sum of two functions. The error function E is defined as

$$E = \sum_{i=1}^{M} \left(\frac{|x_{mi} - x_{f_i}|}{x_{f_i}} \right), \tag{1}$$

where M is time interval (e.g. travel times of 50 to 51 seconds – in the scope of 1 second), x_{mi} and x_{f_i} are frequencies (or occurrence) of i-th travel time measured from the calibrated model and from the field data respectively. Then the penalty function P is

$$P = (cel\_length)^{-8}. \tag{2}$$

This penalization ensures that the solutions where the cell length is very small are not preferred due to noticeable slower simulation runtime. Moreover, it is multiplied by the number of vehicles – N (to add a constant error to every vehicle). Thus the fitness function is

$$F = E + (N \times P). \tag{3}$$

Finally, GA tries to minimize this fitness function F as better solutions are always with lower fitness value.

3.4 Creating a New Population

Selection: After evaluation of all chromosomes from the population $X(i)$ is complete, some of them are selected for next operations using a tournament selection with base 2 giving a new population $X_S(i)$, where $|X_S(i)| = 30$.

Crossover: Two-point crossover is applied between two randomly selected individuals giving a new set $X_C(i)$ (where $X_C(i) \subset X_S(i)$ and $|X_C(i)| = 30$). The first point of crossover operation is between the p_3 and p_4 parameter and the second one right after p_{10} parameter, to allow alternation of the model and the GA parameters individually. This operator is applied with the average probability calculated from two chosen chromosomes (p_c).

Mutation: On all chromosomes from $X_C(i)$ a mutation operator (i.e. changing bit $0 \to 1$ or $1 \to 0$) is applied with the probability (p_m) taken from evaluated individual, which gives a brand new population $X_M(i)$ of the same size.

Population Recovery: Finally, a new population of 60 individuals $X(i+1)$ is selected from the previous population $X(i)$ and the $X_M(i)$ population. This ensures that the best solution will always survive (i.e. elitism is present) [8].

4 Experimental Results

4.1 Field Data

In order to evaluate proposed method, field data have been utilized. Our field data comes from 2431 meters long road segment between two bigger villages in the Slovak Republic with a maximum allowed speed of 50 km/h. This segment is a bit crocked one and there is no allowance for another vehicle advancement due to the local restrictions. The particular segment is on the way to the country seat, so the road is utilized mostly by drivers going to work and back on ordinary business days, but traffic is not strictly homogenous here. This road segment is also a part of the route between two biggest cities in the region and statistically given 5% of traffic comes from bigger vehicles (e.g. busses, trucks, etc.).

Data set was obtained using standard monitoring technology (i.e. detection loops and detection cameras) for every day and night of the year 2010. Therefore, it was possible to measure travel time for vehicles on given road segment. To be able to get frequency of individual travel times (that is used for model comparison), we decided to round these travel times to 1 second scope. Based on this, it is possible to get frequencies of travel times for different intervals (e.g. morning travel times, one day travel times, week travel times, etc.). We utilized two such data sets, where travel time for every single vehicle is present. Frequencies of travel time from ordinary business day (Tuesday, 18/5/2010) (1) and from the last business day (Friday, 21/5/2010) (2) of the same week has been selected. First data set (1) has average travel time of 197.74 seconds for 6702 vehicles (N_1). Second data set (2) has about 11.21 seconds greater average travel time for 8511 vehicles (N_2). It is also important to note that there were sometimes short-term traffic jams during the second selected day. Both data sets for different week day are shown in Fig. 1.

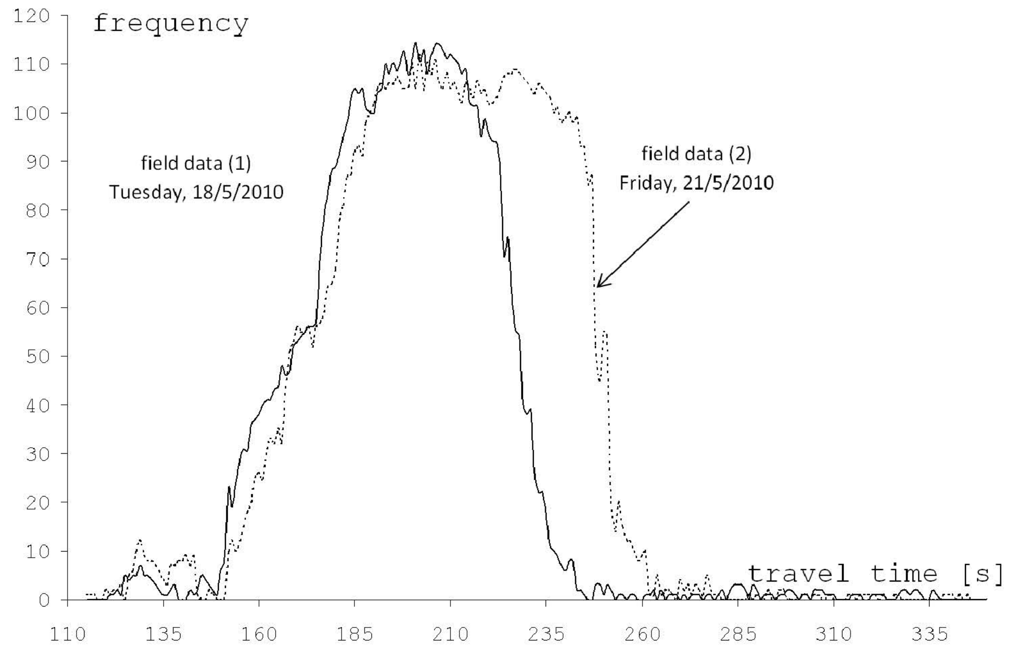

Fig. 1. Frequencies of travel times for two days

4.2 Calibrated Model

All parameters of the CA based microscopic traffic simulation model ($p_1 \ldots p_{10}, p_m$ and p_c), which were evolved for our data sets separately are shown decoded as real values in Tab. 2. All come from the best solution of the last generation (220 000) of GA. Tab. 2 also shows parameters of our previously manually tuned and updated CA model as introduced in [5] and in [6] (in the first column of the table). Some of those manually updated values, are generally not available (GA parameters) or have a bit different meaning in our previous model. Such an example is the low speed boundary value p_6, which is identical with maximal vehicles speed p_4. This is caused by absence of the first parameter in the updated model, because slowing down was performed for all available vehicles (with probability p_5). Also all vehicles in the updated model tend to always accelerate, so $p_7 = 1.0$.

In order to check whether all evolved values are not only a result of stochastic nature of GA we made a simple convergence test and it was discovered, that all parameters tend to evolve to one particular value during generations of GA. Due lack of space we do not illustrate this test results here.

The cell length parameter (p_1) is nearly 2 times less in contrast to our previously manually updated model. This also means that single vehicle is represented using two such cells. The evolved reaction time (p_2) of 1.5 seconds corresponds to the minimal increment of 7.43 km/h (p_1/p_2 as seen in Tab. 1). This parameter is also slightly different compared to our previous model. However, very important finding is that these parameters (p_1 and p_2) converged to the same value for both data sets as they are strictly model oriented. All other parameters ($p_3 \ldots p_{10}$)

Table 2. Parameters and errors for updated model and models evolved for data sets

	Updated model	Model for (1)	Model for (2)
p_1	5.500 m	2.375 m	2.375 m
p_2	1.200 s	1.15 s	1.15 s
p_3	60.5 m	194.75 m	166.25 m
p_4	181.5 $\frac{km}{h}$	81.78 $\frac{km}{h}$	89.22 $\frac{km}{h}$
p_5	0.3000	0.1059	0.4118
p_6	181.5 $\frac{km}{h}$	29.74 $\frac{km}{h}$	59.48 $\frac{km}{h}$
p_7	1.00000	0.8314	0.7569
p_8	n/a	0.1451	0.4549
p_9	12	2	2
p_{10}	12	2	3
p_m	n/a	0.00196	0.00293
p_c	n/a	0.66667	0.66667
E_t on (1)	28.19%	**7.63%**	15.21%
E_t on (2)	36.40%	19.23%	**6.35%**

are a bit different in between two given data sets. The first such parameter is cell neighbor (p_3). It is 194.75 m (i.e. 82 cells) for the first data set (1) and 166.25 m (i.e. 70 cells) for the second data set (2). The maximum allowed speed (p_4) is for both models higher than a local speed restrictions. This represents the real situation at the road segment as some drivers do not keep the maximum speed limit here.

For the model calibrated to (1), the probability of slowing down (p_5) is 0.1059 for vehicle speeds lower than the evolved boundary (p_6) of 29.74 km/h. The same slowing down (in case of speed lower than 59.48 km/h) occurs for 41.18% in the model calibrated to (2). On the other hand, the probability of acceleration (p_7) is higher for (1). The parameter (p_8) of slowing down in case of speeds greater than the evolved boundary speed (p_6) is nearly the same as the previous one (p_5). This could indicate that it would be possible to somehow interoperate both of these parameters and simplify the simulation model.

Parameters p_9 and p_{10} are surprisingly very small and also quite similar. However, based on their convergence tests, we claim that these parameters (i.e. driver sensitivity) can be also statistically obtained for a desired road segment and/or time.

Tab. 2 also shows the average travel time error E_t for one time interval computed as

$$E_t = \frac{E}{M}, \tag{4}$$

where M is the number of time intervals for a given data set. We also measured this error for our manually updated model with additional maximal speed re-adjustment for exact local conditions (e.g. maximal speed). It can be seen that all three new calibrated models, which were obtained using described GA, are significantly better on a particular data set in comparison to our manually updated model (compared previously only with fundamental diagrams). Moreover, all new models are also better when compared to different data sets. This finding is very important for future travel time estimation using simulations.

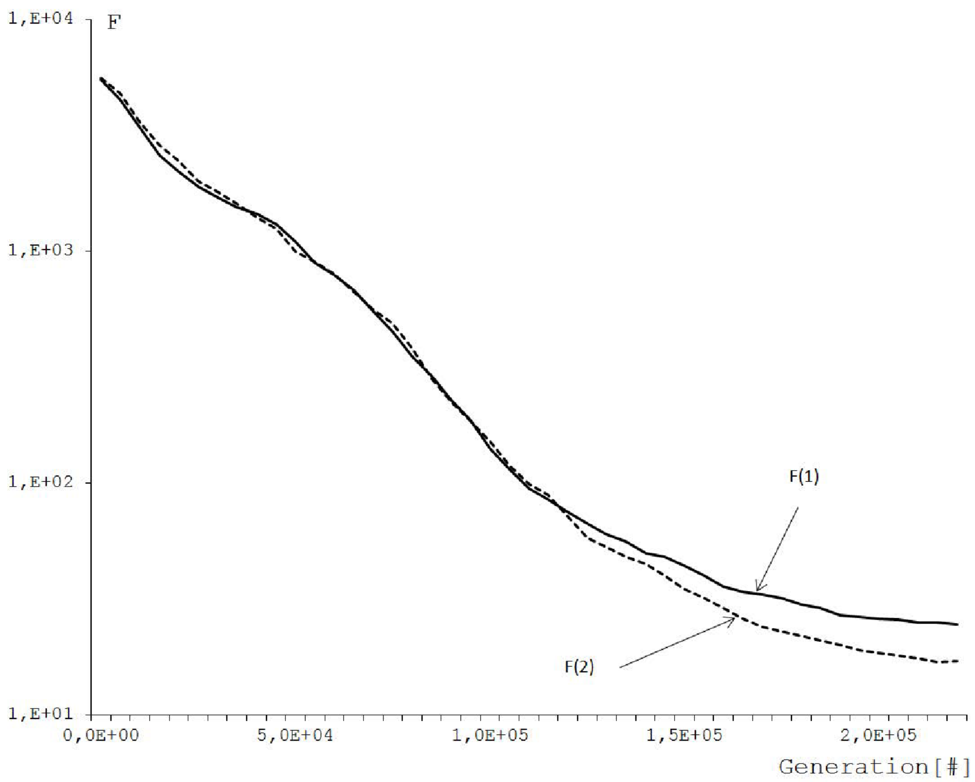

Fig. 2. Fitness $F(1)$ and $F(2)$ in all generations as an average value out of 50 independent runs when calibrating to data set (1) and (2) respectively

It is also important to note, that completing all runs for one data set (50 runs of 220 000 generations) takes more than three days running at *Intel Xeon CPU5420 @ 2.5 GHz* due to need for performing simulations. Fig. 2 shows the average fitness value $F(1)$ and $F(2)$ for 50 successive runs and for data set (1) and (2) respectively. Note that y-axis is in the logarithmic scale. It can be also seen, that quality tends to increase (lower fitness) during evolution of 220 000 generations which is ensured by elitism. After that number of generations, the quality of population is not changing dramatically. Our genetic algorithm was tuned to always find a reasonable solution after this number of generations. The whole tuning process will be described in the forthcoming paper.

5 Conclusions

In this work, we proposed an effective calibration method for a simple microscopic traffic simulation model. The proposed model is based on the cellular automaton, which can easily be accelerated. We utilized genetic algorithm for model parameters optimization, that was able to find all parameters of the CA model for a given field data. We increased the precision of simulator in average by more than 1/4 in comparison with our previously updated and manually tuned model. Furthermore, new evolved models have better stability compared to original model (i.e. model calibrated to (1) utilizable for (2) and vice versa). Therefore, the proposed methods seem to be promising for calibration in the task of travel time estimation of pre-selected road segments of interest.

In our future work, it would be very interesting to derive how much data has to be used for a proper model calibration in the case when a sufficient amount of data is not available. This could be very important in the travel time estimation using such calibrated cellular automaton based models.

Acknowledgments. This work has been partially suported by the Czech Science Foundation under projects P103/10/1517 and GD102/09/H042, by the IT4Innovations Centre of Excellence project CZ.1.05/1.1.00/02.0070, by the research program MSM 0021630528 and FIT BUT grant FIT-S-12-1.

References

1. Hellinga, B.R.: Requirement for the Calibration of Traffic Simulation Models. Department of Civil Engineering, University of Waterloo (2009)
2. Dowling, R., et al.: Guidelines for Calibration of Microsimulation Models: Framework and Applications. Transportation Research Record: Journal of the Transportation Research Board, Transportation Research Board of the National Academies 1876, 1–9 (2004) ISSN: 0361-1981
3. Punzo, V., Simonelli, F.: Analysis and Comparison of Microscopic Traffic Flow Models with Real Traffic Microscopic Data. Transportation Research Record: Journal of the Transportation Research Board, Transportation Research Board of the National Academies 1934, 53–63 (2005) ISSN: 0361-1981
4. Van Aerde, M., Rakha, H.: Multivariate Calibration of Single Regime Speed-Flow-Density Relationships. In: Vehicle Navigation and Information Systems (VNIS) Conference, Seattle (1995)
5. Korcek, P., Sekanina, L., Fucik, O.: A Scalable Cellular Automata Based Microscopic Traffic Simulation. In: IEEE Intelligent Vehicles Symposium 2011 (IV11), pp. 13–18. IEEE ITSS, Baden-Baden (2011) ISBN 978-1-4577-0889-3
6. Korcek, P., Sekanina, L., Fucik, O.: Cellular automata based traffic simulation accelerated on GPU. In: 17th International Conference on Soft Computing (MENDEL 2011), Brno, CZ, UAI FSI VUT, pp. 395–402 (2011) ISBN 978-80-214-4302-0
7. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. Journal de Physique I 2(12), 2221–2229 (1992)
8. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley (1989) ISBN: 0201157675

Paper V

Advanced Approach to Calibration of Traffic Microsimulation Models Using Travel Times.

Journal of Cellular Automata, Vol. 8, pp. 457–467
Reprints available directly from the publisher
Photocopying permitted by license only

Advanced Approach to Calibration of Traffic Microsimulation Models Using Travel Times

PAVOL KORCEK*, LUKAS SEKANINA AND OTTO FUCIK

Brno University of Technology, Faculty of Information Technology
IT 4 Innovations Centre of Excellence, 612 66 Brno, Czech Republic

Received: March 15, 2013. Accepted: August 1, 2013.

An effective calibration method of the cellular automaton based traffic microsimulation model is proposed in this paper. It is shown that by utilizing a genetic algorithm it is possible to calibrate different parameters of the model much better than a traffic expert. Moreover, using this process it is also possible to find several model parameters that are extremely difficult to calibrate as relevant data can not be measured using standard monitoring technologies or complete data sets are often not available. The quality of the new calibrated models is discussed in the task of vehicle travel time estimation. The precision of simulations is increased over three times compared to a manually tuned model. The average error rate is *10.75 %* in comparison with several field travel time data.

Keywords: Traffic, simulation, cellular automaton model, genetic algorithm, calibration, travel time.

1 INTRODUCTION

Traffic microsimulation models distinguish and trace every single vehicle or driver on the road. In cellular automaton (CA) based models each CA cell represents a specified road segment, e.g. 7.5 meter long [11]. The cell contains information if it is occupied by vehicle, and if so, also vehicle speed is known. This speed is updated according to the CA local transition function (see later). Such CA based models were shown to be able to capture all basic phenomena that occur in traffic flows [4], not only in the field of vehicular

* E-mail: ikorcek@fit.vutbr.cz

traffic flow modeling, but also in other fields such as pedestrian behavior, escape dynamics, etc.

Any traffic simulation model has to be calibrated and validated prior to its real deployment [1, 5]. The researchers usually do this on their own using some data sets that they have access to and publish the results obtained. For example, authors of recent papers (e.g. [6, 12]) utilized the field data gathered from global positioning system (GPS) for calibration. They showed, that the calibrated microsimulation models exhibit an average error of about 20%. Nevertheless, it should be noted that for exploiting such calibration approach, the GPS data is generally not available. Even if it is available, the data can be used only if it were measured in the same or at least similar traffic facility type (e.g. highway vs. local road). However, in paper [3] the authors tried to benchmark microsimulation models with more common traffic field data. Equally to our approach, they used vehicle travel times that can be obtained from standard monitoring technologies (i.e. inductive loops placed in the road and traffic detection cameras). They showed that it is possible to get travel time estimation error of about 16% for the best traffic microsimulation model [10].

In this work, we propose to utilize our CA based microsimulation model to efficient and fast traffic simulation. This model was shown not only to be capable of achieving multiple in real-time simulations (e.g. [7]), but it was also updated to eliminate unwanted properties of ordinary CA based models such as stopping vehicle from maximal speed to zero in one simulation step. The quality of this updated model has been previously evaluated by comparison with *Van Aerde* traffic fundamental diagrams [8]. We will show, that by careful calibration of key model parameters it is possible to achieve a better precision of travel time estimation compared to other models for a given road segment. Moreover, except CA model parameters, we will also calibrate some parameters (such as driver sensitivity) that, as stated for example in [13], are extremely difficult to optimize with other common techniques. The calibration will be performed by genetic algorithm (GA). This paper extends our previous work [9] in two directions. The first one consists in the form of data utilized in the calibration process. We are using exact vehicle travel time values with no other post-processing instead of travel time frequencies. The second difference is that in the used data sets certain travel time values are absent. The objective is to simulate real life situations by means of these incomplete data sets (e.g. to simulate traffic sensor errors).

The rest of the paper is organized as follows. Section 2 introduces the updated local transition function of CA based model. The calibration process utilizing the genetic algorithm is described in Section 3. Then, in Section 4, experimental evaluations for our field data sets are presented and discussed. Finally, conclusions and suggestions for future work are given in Section 5.

2 UPDATED LOCAL TRANSITION FUNCTION

In our previous work [8], we updated the CA local transition function originally consisting of only four successive steps (i.e. acceleration, slowing down, randomization, and car motion) [11] to a new form, where some brand new parameters can be found (p_j in Algorithm 1).

Our traffic model is extended to eliminate unwanted properties of ordinary CA based models, such as stopping from maximum vehicle speed to zero in one step. This is possible due to storing the previous (or leading) vehicle velocity $v_v(i+1)$. If there is such vehicle, the following vehicle (i) is able to determine its positive or negative acceleration by means of function $acc(i+1)$.

According to Algorithm 1, it is firstly determined if investigated vehicle (i) could accelerate (i.e. vehicle velocity $v_v(i)$ is not greater than the maximal vehicle speed p_4 or given vehicle speed limit $v_{max}(i)$). If so, its speed-up is accomplished with probability p_7, so not all vehicles tend to always accelerate as in the original model [11]. Then, if there is a plenty of room for vehicle to get in (i.e. $gap(i) + acc(i+1) > v_v(i)$) or there is no previous

---

Algorithm 1 Updated local transition function of CA microsimulation model.

---

if $v_v(i) < p_4$ and $v_v(i) < v_{max}(i)$ **then**
  $v_v(i) := v_v(i) + 1$ with probability p_7
end if
if $(gap(i) + acc(i+1)) > v_v(i)$ **then**
  if $v_v(i) < p_6$ **then**
    $v_v(i) := v_v(i) - 1$ with probability p_5
  else
    $v_v(i) := v_v(i) - 1$ with probability p_8
  end if
else
  if $acc(i+1) > 0$ **then**
    $v_v(i) := 1/p_9 \times (gap(i) + acc(i+1))$
  else
    $v_v(i) := 1/p_{10} \times (gap(i) + acc(i+1))$
  end if
end if

Ensure: Each vehicle i is advanced $v_v(i)$ times and $v_{prev}(i) := v_v(i)$.

---

vehicle in the same lane, collision avoidance mechanism is not performed (see later). Similarly to the original CA local transition function, only deceleration based on probabilities could be applied in this situation. In case of small vehicle speeds ($v_v(i) < p_6$), deceleration is performed with the probability p_5, otherwise ($v_v(i) > p_6$) with the probability p_8. Collision avoidance occurs only when there is no free room for the vehicle in the same lane to get in (i.e. $gap(i) + acc(i+1) \leq v_v(i)$). Two basic situations may occur. If the leading vehicle tends to accelerate ($acc(i+1) > 0$), the actual vehicle speed $v_v(i)$ is reduced to $1/p_9 \times (gap(i) + acc(i+1))$. Otherwise ($acc(i+1) \leq 0$), actual vehicle speed $v_v(i)$ should be reduced more strictly to $1/p_{10} \times (gap(i) + acc(i+1))$. It can be seen that parameters p_9 and p_{10} are more driver-based than model-oriented. We will try to find out if they could be determined statistically for a given road segment. Finally, each vehicle is advanced $v_v(i)$ sites and the velocity updates are performed.

There are also some other parameters that are not shown in Algorithm 1. It is the cell length – p_1, reaction time (simulation step) – p_2 and the cell neighborhood – p_3.

3 CALIBRATION OF THE CA BASED MODEL

Genetic algorithms are widely used in various areas to find solutions to hard optimization and design problems [2]. The main idea is to evolve a population (set) of candidate solutions to find better ones. A candidate solution is encoded as a chromosome which is an abstract representation that can be modified with standard genetic operators such as mutation and crossover. In this work, GA is used to find and calibrate all parameters of the CA model in order to maximize the precision of traffic simulations.

3.1 Parameters Encoding

In order to simplify GA, all simulation model parameters that will be calibrated are encoded in the binary form. Real numbers are encoded as fixed-point numbers. All encoded parameters with their respective minimal values, maximal values and step, are briefly summarized in Table 1.

3.2 Chromosome

The proposed GA has a self-adaptation capability, which means that the parameters of the algorithm (the probability of mutation p_m and crossover p_c) are also part of the chromosome. Hence the user is not forced to set them. The whole set of parameters is represented using one 92-bit number. It is important to note that each parameter of the chromosome is encoded using *Gray encoding* to ensure that the maximal Hamming distance between two

	No. of bits used	Min. value	Max. value	Step
p_1	6	0.125	8.000	0.125
p_2	6	0.05	3.20	0.05
p_3	12	p_1	$2^{12} \times p_1$	p_1
p_4	11	p_1/p_2	$2^{11} \times p_1/p_2$	p_1/p_2
p_5	8	0.00392	1.00000	0.00392
p_6	9	p_1/p_2	$2^9 \times p_1/p_2$	p_1/p_2
p_7	8	0.00392	1.00000	0.00392
p_8	8	0.00392	1.00000	0.00392
p_9	5	1	32	1
p_{10}	5	1	32	1
p_m	10	0.00097	1.00000	0.00097
p_c	4	0.06667	1.00000	0.06667

TABLE 1
CA microsimulation model parameters and values.

successive values is one. This setup does not allow big jumps between values in case of a single bit change. The first population ($X(0)$) consists of 60 such chromosomes ($|X(0)| = 60$) generated randomly.

3.3 Fitness Function

All chromosomes from population X_i are separately evaluated using the same fitness function. Firstly, a candidate CA road segment is constructed using the parameters obtained from a candidate chromosome. Then a simulation is performed for that model. Incoming vehicles are generated depending on their time of arrival and with their actual speed based on the measured value from the field. Vehicles outgoing from the simulated road segment are simply removed, but their travel time is recorded.

The whole simulation is executed until the number of simulated vehicles is the same as the number of vehicles in the field data. After that, the fitness function $F_x()$ is calculated as a sum of error function E_x and penalty function P_x (all for given data set x). The error function is defined as

$$E_x = \sum_{i=1}^{N_x} \left(\frac{|y_{m\,i} - y_{f\,i}|}{N_x} \right), \tag{1}$$

where N_x is the number of travel time samples in the data set x, $y_{m\,i}$ and $y_{f\,i}$ are travel time values of the i-th vehicle measured from the new calibrated model and from the field data respectively. The penalty function

$$P_x = (cell\_length)^{-8} \tag{2}$$

Algorithm 2 Genetic algorithm based calibration procedure.

$i = 0$
Generate population $X(i)$ randomly, $|X(i)| = 60$
Evaluate all candidates from $X(i)$ with F_x using simulations
repeat
1. Create $X_S(i)$ using *tournament selection* from $X(i)$
2. Create $X_C(i)$ using *crossover operator* on $X_S(i)$
3. Create $X_M(i)$ using *mutation operator* on $X_C(i)$
4. *Evaluate* all candidates from $X_M(i)$ with F_x using simulations
5. Create $X(i+1)$ by selecting 60 best individuals from $X_M(i) \cup X(i)$
6. $i := i + 1$

until $(i \leq G)$

ensures that the solutions where the cell length is very small are not preferred due to the slower simulation runtime. GA tries to minimize the fitness function in which better solutions are always those with lower fitness values.

3.4 New Population

After evaluation of all chromosomes from the population $X(i)$ is complete, some of them are selected for next operations using a tournament selection with base 2 giving a new population $X_S(i)$, where $|X_S(i)| = 30$. Two-point crossover is applied on two randomly selected individuals giving a new set $X_C(i)$ (where $X_C(i) \subset X_S(i)$ and $|X_C(i)| = 30$). The first point of the crossover operation is between parameters p_3 and p_4, and the second one right after p_{10} parameter, to allow alternation of the model and the GA parameters individually. This operator is applied with the probability calculated as the average of p_c values. On all chromosomes from $X_C(i)$, a mutation operator (i.e. bit inversion) is applied with the probability (p_m) taken from evaluated individual, which gives a brand new population $X_M(i)$ of the same size.

Finally, a new population of 60 individuals $X(i+1)$ is selected from the previous population $X(i)$ and the $X_M(i)$ population. This ensures that the best solution will always survive (i.e. the elitism is present) [2].

Described GA procedure is repeated until specified number of generations (G) is exhausted as shown in Algorithm 2.

4 EXPERIMENTAL RESULTS

4.1 Field Data

The field data have been utilized in order to evaluate the proposed method. Our data comes from a 2431 meter long road segment between two bigger villages in the Slovak Republic with the speed limit of 50 km/h. This segment is a bit crocked one and there is no allowance for another vehicle advancement due to the local restrictions.

The data was obtained using traffic monitoring system for every day and night over the year 2010. Therefore, it was possible to measure travel time for vehicles on this road segment. We utilized travel times from ordinary business day (Tuesday, 18/5/2010) (1) and travel times from a day with much denser traffic (Friday, 21/5/2010) (2). The first data set (1) has an average travel time of 197.74 seconds for 6702 vehicles (N_1) and the second data set (2) has about 11.21 seconds longer average travel time for 8511 vehicles (N_2). Moreover, we derived two more data sets. We decided to withdraw every second travel time sample from both previous data sets. Therefore, four data sets are utilized in experiments: original (and full) data set (1) and (2), data set (3) derived from (1) containing only 3351 travel times (N_3) for N_1 vehicles, and data set (4) derived from (2) containing only 4256 travel times (N_4) for N_2 vehicles.

4.2 Calibrated Models

All parameters of the CA based microscopic traffic simulation model ($p_1 \ldots p_{10}$, p_m and p_c) that were evolved for all four data sets separately are shown decoded as real numbers in Table 2. All results come from the best solution of GA (after 6e5 generations).

Table 2 also shows the parameters of our previously manually updated CA model (in the first column of the table) as introduced in [8] and [7]. Some of those manually updated values are generally unavailable (GA parameters)

Parameter	**Prev. model [8]**	**(1)**	**(2)**	**(3)**	**(4)**
$p_1[m]$	5.500	2.375	2.375	2.375	2.375
$p_2[s]$	1.200	1.15	1.15	1.15	1.15
$p_3[m]$	60.5	194.75	166.25	190.00	171.00
$p_4[\frac{km}{h}]$	181.5	81.78	89.22	74.35	81.78
p_5	0.3000	0.1098	0.4078	0.1137	0.4510
$p_6[\frac{km}{h}]$	181.5	22.30	52.04	29.74	59.48
p_7	1.0000	0.8353	0.7608	0.8117	0.7529
p_8	n/a	0.1490	0.4471	0.1451	0.4745
p_9	12	2	2	2	3
p_{10}	12	3	3	3	4
p_m	n/a	0.0012	0.0029	0.0029	0.0039
p_c	n/a	0.6667	0.6667	0.6667	0.6667
$E_{(1)}[\%]$	31.57	5.86	**9.89**	5.91	**10.64**
$E_{(2)}[\%]$	37.04	**11.19**	5.28	**11.27**	5.47

TABLE 2
Parameters and average errors for models evolved for different data sets.

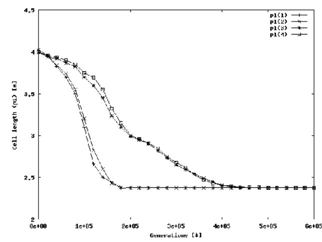

FIGURE 1
Parameter p_1 (cell length) in all generations as an average value out of 50 runs.

or have a bit different meaning in this model. Such an example is the low speed boundary value p_6, which is identical with maximal vehicles speed p_4. This is caused by the absence of the first parameter in this manually updated model, because slowing down was performed for all available vehicles equally (with probability p_5). Also all vehicles in that model tend to always accelerate (the probability p_7 is 1.0).

In order to check whether some of evolved values are not only a result of the stochastic nature of GA, we made a simple convergence test. Figure 1 shows the evolution of parameter p_2 (the cell length) during 6e5 generations as an average value out of 50 independent runs of GA. Nevertheless, it can clearly be seen that this parameter tends to converge to one particular value in all data sets. A similar test was performed for every one evolved parameter, but due to lack of space we do not illustrate them here.

The cell length (parameter p_1) is nearly twice shorter than our previously manually updated model. This also means that a single vehicle has to be represented using two cells. The evolved reaction time (p_2) of 1.15 seconds corresponds to the minimal increment of 7.43 km/h (p_1/p_2). These parameters are also slightly different compared to our previous model (i.e. 5 meters and 1.2 seconds). However, a very important finding is that both parameters (p_1 and p_2) converged to the same value for all data sets as they are strictly model-oriented and they do not depend on the measurement time of the field data.

All other parameters ($p_3 \ldots p_{10}$) fluctuate among data sets. The first such parameter is the cell neighbor (p_3). It can be seen that it is greater in the model calibrated for the first day (2) even if half calibration data is available (3). Then the maximum allowed speed (p_4) is for both models higher than the local speed restriction. This clearly represents the real situation at the road segment as some drivers do not keep the maximum speed limit. For the model calibrated to (1) and also to (3), the probability of slowing down (p_5) is quite smaller for vehicles where the speed is lower than the evolved boundary (p_6) compared to the rest two models. On the other hand, the probability of acceleration (p_7) is a bit higher for (1) and (3). The parameter (p_8) of slowing down in case of speeds greater than the evolved boundary speed (p_6) is nearly the same as the previous one (p_5). This could indicate that it would be possible to somehow interoperate both of these parameters and simplify the simulation model.

The values of both parameters p_9 and p_{10} are very small and also quite similar. However, based on their convergence tests (i.e. the test whether the parameters tend to evolve to one particular value during the progress of GA), we claim that these parameters (i.e. driver sensitivity) can also be statistically obtained for a desired road segment.

Table 2 shows the average travel time error E_x (in percent) for the single vehicle. The first data set (1) can be read as the calibration data and the second data set (2) as a validation data for the first model and vice versa. Therefore, we achieved precision of 11.19 %, 9.89 %, 11.27 % and 10.64 % in the models calibrated subsequently to (1), (2), (3) and (4). All models were validated subsequently on data set (2), (1), (2) and finally again on (1). As we utilized the exact values of vehicle travel times it is quite easy to derive an average travel time error (in seconds) for each vehicle.

We also calculated this error for our manually updated model with additional model re-adjustment to local conditions (e.g. maximal speed). Naturally, the error for the new optimized models is much lower (in average by 3.19 times) compared to the manually tuned one.

Figure 2 shows the average fitness value $F(1)$, $F(2)$, $F(3)$ and $F(4)$ for 50 successive runs and for all data sets respectively. Note that the y-axis is in the logarithmic scale. It can clearly be seen that the quality tends to increase (lower fitness value) during generations. Depending on the data set, after a certain number of generations, the quality of population is not changing significantly. It can be seen that for complete data sets (1) and (2) it takes approximately 2e5 generations to get stable results. On the other side, for incomplete data sets (3) and (4) it takes three times longer (i.e. 6e5 generations) to find a solution with a comparable quality.

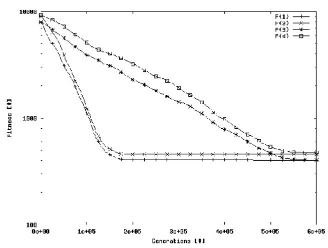

FIGURE 2
Fitness in all generations as an average value out of 50 independent runs.

5 FINAL CONCLUSIONS

In this work, we proposed an effective calibration method for CA based microscopic traffic simulation model. The proposed model is based on the cellular automaton, which can easily be accelerated and, therefore, used for large-scale real-time simulations. We utilized the genetic algorithm for the model parameters calibration which was able to find all parameters of the CA model for a given field data. Using this process, we increased the precision of the model more than three times compared to the manually tuned model and in average by 10.75 % compared to the field data. This is also a significantly better result than relevant findings in the same area [3].

Furthermore, new evolved models have better generalizing capability and hence the model calibrated to (1) can be utilized for (2) or vice versa. This finding is very important for future vehicle travel time estimations using microsimulation. The proposed methods seem to be promising for calibration in the task of travel time estimation in the pre-selected road segments of interest. We also discovered, that at the cost of three times longer calibration time, it is possible to reach almost the same error even if not all the field data is available.

ACKNOWLEDGMENTS

This work has been partially supported by the Czech Science Foundation under Natural Computing on Unconventional Platforms project P103/10/1517, the IT4Innovations Centre of Excellence project CZ.1.05/1.1.00/02.0070, and the Transport Systems Development Centre (RODOS) project TE01020155.

REFERENCES

[1] Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H. N., and Wenm, Y. (2007). Calibration of microscopic traffic simulation models: Methods and application. *Journal of the Transportation Research Board*, 1999:198–207.

[2] Bentley, P. J. (ed.). (1999). *Evolutionary Design by Computers*. Morgan Kaufmann, San Francisco CA.

[3] Brockfeld, E., Kuhn, R., Skabardonis, A., and Wagner, P. (2003). Towards a benchmarking of microscopic traffic flow models. *Journal of the Transportation Research Board*, 1852:124–129.

[4] Chowdhury, D., Santen, L., and Schadschneider, A. (2000). Statistical physics of vehicular traffic and some related systems. *Physics Reports*, 329:199–329.

[5] Hellinga, B. R. (1998). Requirements for the calibration of traffic simulation models. In *Proceedings of the Canadian Society for Civil Engineering CSCE*, volume 4, pages 211–222.

[6] Kesting, A., and Treiber, M. (2008). Calibrating car-following models by using trajectory data: Methodological study. *Journal of the Transportation Research Board*, 2088:148–156.

[7] Korcek, P., Sekanina, L., and Fucik, O. (2011). Cellular automata based traffic simulation accelerated on GPU. In *Proceedings of the 17th International Conference on Soft Computing*, pages 395–402. BUT.

[8] Korcek, P., Sekanina, L., and Fucik, O. (2011). A scalable cellular automata based microscopic traffic simulation. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2011*, pages 13–18. IEEE.

[9] Korcek, P., Sekanina, L., and Fucik, O. (2012). Calibration of traffic simulation models using vehicle travel times. In *Cellular Automata*, volume 7495 of *Lecture Notes in Computer Science*, pages 807–816. Springer Berlin Heidelberg.

[10] Krauss, S. (1998). *Microscopic modeling of traffic flow: Investigation of Collision Free Vehicle Dynamics*. PhD dissertation, University of Cologne.

[11] Nagel, K., and Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. *Journal de Physique I*, 2(12):2221–2229.

[12] Punzo, V., and Simonelli, F. (2005). Analysis and comparison of microscopic traffic flow models with real traffic microscopic data. *Journal of the Transportation Research Board*, 1934:53–63.

[13] Van Aerde, M., and Rakha, H. (1995). Multivariate calibration of single regime speed-flow-density relationships. In *Proceedings of the Vehicle Navigation and Information Systems Conference*, pages 334–341. IEEE.