Korcek, P., Sekanina, L., Fucik, O.: Advanced Approach to Calibration of Traffic Microsimulation Models using Travel Times, In: Journal of Cellular Automata, Vol. 8, No. 6, 2013, Philadelphia, US, p. 457-467, ISSN 1557-5969

Advanced Approach to Calibration of Traffic Microsimulation Models using Travel Times

PAVOL KORCEK*, LUKAS SEKANINA, OTTO FUCIK

Brno University of Technology
Faculty of Information Technology
IT4Innovations Centre of Excellence
612 66 Brno, Czech Republic

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.

Key words: 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

* email: `ikorcek@fit.vutbr.cz`

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 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 [5], [1]. 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. [12], [6]) 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).

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 × (gap(i) + acc(i + 1))
    else
        v_v(i) := 1/p_10 × (gap(i) + acc(i + 1))
    end if
end if
```

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

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 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.

	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.

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 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

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

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.

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 $(i \leq G)$

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

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.

decoded as real numbers in Table 2. All results come from the best solution of GA (after 6×10^5 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) 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 6×10^5 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

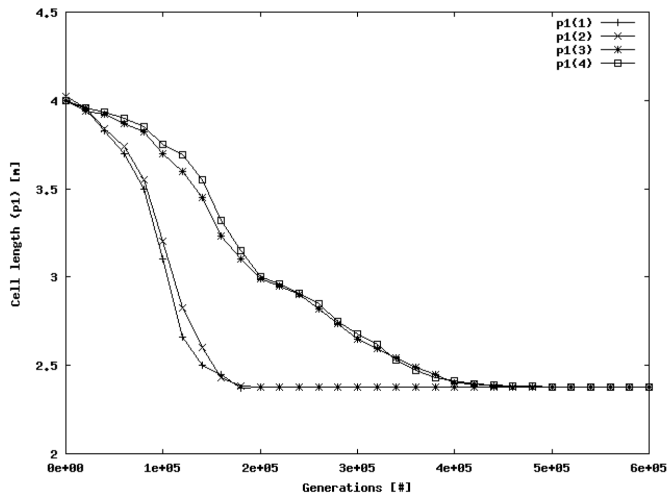

FIGURE 1
Parameter p_1 (cell length) in all generations as an average value out of 50 runs.

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 ($p_3 \dots p_{10}$) 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 2×10^5 generations to get stable results. On the other side, for incomplete data sets (3) and (4) it takes three times longer (i.e. 6×10^5 generations) to find a solution with a comparable quality.

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

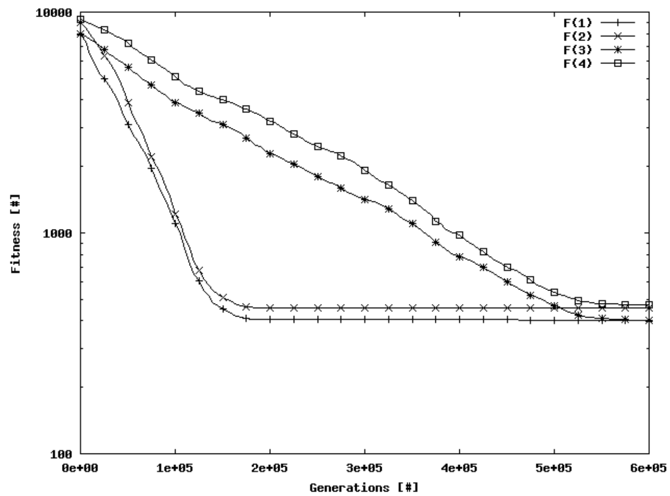

FIGURE 2
Fitness in all generations as an average value out of 50 independent runs.

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.

6 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.