Cellular automata models for transportation applications

Kai Nagel

Institute for Scientific Computing, Swiss Federal Institute of Technology Zürich (ETHZ),
8092 Zürich, Switzerland

Abstract. This paper gives an overview of the use of CA modes for transportation applications. In transportation applications, the CA dynamics is embedded within several other concepts, such as the fact that the dynamics unfolds on a graph instead of on flat 2d space, or multi-agent modeling. The paper also discusses the the limits of the CA technology in traffic.

1 Introduction

Cellular automata methods have their applications primarily in areas of spatio-temporal dynamics. Transportation simulations, with travelers and vehicles moving through cities, fall into that category. There are however also important differences between a "standard" CA simulation and those used in traffic. These differences are that in traffic, the dynamics is normally considered as unfolding on a graph instead of on flat space, and that particles in transportation simulations are better characterized as "intelligent agents". These aspects will be discussed in Secs. 3 and 4. This is followed by a discussion of the limits of the CA technology and relations to other methods (Sec. 5), and a short outlook on a simulation of "all of Switzerland" (Sec. 6). The paper is concluded by a summary.

2 CA rules for traffic

In CA models for traffic, space is typically coarse-grained to the length a car occupies in a jam ($\ell = 1/\rho_{jam} \approx 7.5$ m), and time typically to one second (which can be justified by reaction-time arguments [1]). One of the side-effects of this convention is that space can be measured in "cells" and time in "time steps", and usually these units are assumed implicitly and thus left out of the equations. A speed of, say, $v = 5$, means that the vehicle travels five cells per time step, or 37.5 m/s, or 135 km/h, or approx. 85 mph.

Deterministic traffic CA Typical CA for traffic represent the single-lane road as a 1-dimensional array of cells of length ℓ, each cell either empty or occupied by a single vehicle. Vehicles have integer velocities between zero and v_{max}. A possible update rule is [2]

Car following:	$v_{t+1} = \min\{g, v_t + 1, v_{max}\}$
Movement:	$x_{t+1} = x_t + v_{t+1}$

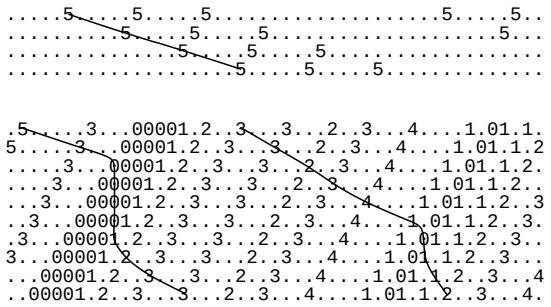

Fig. 1. Sequence of configurations of CA 184. Lines show configurations of a segment of road in second-by-second time steps; traffic is from left to right. Integer numbers denote the velocities of the cars. For example, a vehicle with speed "3" will move three sites (dots) forward. Via this mechanism, one can follow the movement of vehicles from left to right, as indicated by some example trajectories. TOP: Uncongested traffic. BOTTOM: Congested traffic.

The first rule describes deterministic car-following: try to accelerate by one velocity unit except when the gap is too small or when the maximum velocity is reached. g is the gap, i.e. the number of empty cells between the vehicle under consideration and the vehicle ahead, and v is measured in "cells per time step".

This rule is similar to the CA rule 184 in the Wolfram classification [3]; indeed, for $v_{max} = 1$ it is identical. This model has some important features of traffic, such as start-stop waves, but it is unrealistically "stiff" in its dynamics.

For this CA, it turns out that, after transients have died out, there are two regimes, depending on the system-wide density ρ_L (Fig. 1):

- Laminar traffic. All vehicles have gaps of v_{max} or larger, and speed v_{max}. Flow in consequence is $q = \rho\, v_{max}$.
- Congested traffic. All vehicles have gaps of v_{max} or smaller. It turns out that their speed is always equal to their gap. This means that $\sum v_i = \sum g_i = N_{veh} \times \langle g \rangle$. Since density $\rho = 1/(\langle g \rangle + 1)$, this leads to $q = \rho\, v = 1 - \rho$.

The two regimes meet at $\rho_c = 1/(v_{max} + 1)$ and $q_c = v_{max}/(v_{max} + 1)$; this is also the point of maximum flow.

Stochastic traffic CA (STCA) One can add noise to the CA model by adding a randomization term [4]:

Car following:	$v_{t+\frac{1}{2}} = \min\{v_t + 1, g_t, v_{\max}\}$
Randomization:	$v_{t+1} = \begin{cases} \max\{v_{t+\frac{1}{2}} - 1, 0\} & \text{with probability } p_n \\ v_{t+\frac{1}{2}} & \text{else} \end{cases}$
Moving:	$x_{t+1} = x_t + v_{t+1}$

t and $t+1$ refer to the actual time-steps of the simulation; $t+\frac{1}{2}$ denotes an intermediate result used during the computation.

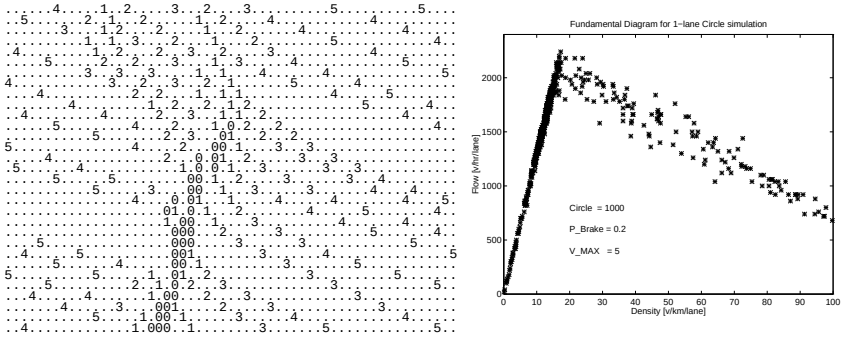

Fig. 2. Stochastic CA. LEFT: Jam out of nowhere leading to congested traffic. RIGHT: One-lane fundamental diagram as obtained with the standard cellular automata model for traffic using $p_{noise} = 0.2$; from [6].

With probability p_n, a vehicle ends up being slower than calculated deterministically. This parameter simultaneously models effects of (i) speed fluctuations at free driving, (ii) over-reactions at braking and car-following, and (iii) randomness during acceleration periods.

This makes the dynamics of the model significantly more realistic (Fig. 2). $p_{noise} = 0.5$ is a standard choice for theoretical work (e.g. [5]); $p_{noise} = 0.2$ is more realistic with respect to the resulting value for maximum flow (capacity), see Fig. 2 (right) [6].

Slow-to-start (s2s) rules/velocity-dependent randomization (VDR) Real traffic has a strong hysteresis effect near maximum flow: When coming from low densities, traffic stays laminar and fast up to a certain density ρ_2. Above that, traffic "breaks down" into start-stop traffic. When lowering the density again, however, it does not become laminar again until $\rho < \rho_1$, which is significantly smaller than ρ_2, up to 30% [7, 8]. This effect can be included into the above rules by making acceleration out of stopped traffic weaker than acceleration at all other speeds, for example by making the probability p_n in the STCA velocity-dependent: If $p_n(v = 0) > p_n(v \geq 1)$, then the speed reduction through the randomization step is more often applied to vehicles with speed zero than to other vehicles. Such rules are called "slow-to-start" rules [9, 10].

Time-oriented CA (TOCA) A modification to make the STCA more realistic is the so-called time-oriented CA (TOCA) [11]. The motivation is to introduce a higher amount of elasticity in the car following, that is, vehicles should accelerate and decelerate at larger distances to the vehicle ahead than in the STCA, and resort to emergency braking only if they get too close. The rule set is easier to write in algorithmic notation, where $v := v + 1$ means that the variable v is increased by one at this line of the program. For the TOCA velocity update, the following operations need to be done in sequence for each car:

1. if ($g > v \cdot \tau_H$) then, with probability p_{ac}: $v := \min\{v + 1, v_{max}\}$;
2. $v := \min\{v, g\}$

3. if ($g < v \cdot \tau_H$) then, with probability p_{dc}: $v := \max\{v-1, 0\}$.

Typical values for the free parameters are $(p_{ac}, p_{dc}, \tau_H) = (0.9, 0.9, 1.1)$. The TOCA generates more realistic fundamental diagrams than the original STCA, in particular when used in conjunction with lane-changing rules on multi-lane streets.

Dependence on the velocity of the car ahead The above rules use gap alone as the controlled variable. More sophisticated rules will use more variables, for example the first derivative of the gap, which is the velocity difference. The idea is that if the car ahead is faster, then this adds to one's effective gap and one may drive faster than without this. In the CA context, the challenge is to retain a collision-free parallel update. Ref. [12] achieved this by going through the velocity update twice, where in the second round any major velocity changes of the vehicle ahead were included. Ref. [13] instead also looked at the gap of the vehicle ahead. The idea here is that, if we know both the speed and the gap of the vehicle ahead, and we make assumptions about the driver behavior of the vehicle ahead, then we can compute bounds on the behavior of the vehicle ahead in the next time step.

Traffic breakdown An interesting topic is the transition from laminar to congested traffic. For the deterministic model, things are clear: The laminar regime is when all vehicles move at full speed; the congested regime is when at least one vehicle in the system does not move at full speed. Deterministic models can also display bi-stability, i.e. density ranges where both the laminar and the congested phase are stable. This is for example the case with deterministic slow-to-start models [14]. This characterization is the same as for deterministic fluid-dynamical models [15].

For stochastic models, things are less clear since even in the laminar regime there may be slow vehicles, their slowness caused by random fluctuations. Often, the analogy to a gas/liquid transition is used, meaning that traffic jams are droplets of the liquid phase interdispersed in the gaseous phase of laminar traffic. However, the question of a phase transition in stochastic models has not been completely settled [16–18]. The main problem seems to be that questions of meta-stability and of phase separation are not treated separately, although they should be, as our own recent investigations show [19].

Lane changing Lane changing is implemented as an additional sub-timestep before the velocity update. Lane changing consists of two parts: the reason to change lanes, and the safety criterion. The first one can be caused by slow cars ahead, or by the desire to be in the correct lane for a turn at the end of a link. The safety criterion means that there should be enough empty space around a vehicle which changes lanes. A simple symmetric implementation of these principles is:

- Reason to change lanes (incentive criterion): $g \leq v$ .AND. $g_o > g$, where g is the standard gap, and g_o is the gap ahead on the other lane. The rule means that a reason to change lanes is given when the gap on the current lane inhibits speed and when the gap on the other lane is larger. – This is a simple symmetric criterion based on gaps, more complicated and/or asymmetric criteria are possible [20].
- Safety criterion: $g_o \geq v$ .AND. $g_{b,o} > v_{b,o}$, where the index $_{b,o}$ refers to the vehicle coming from from behind on the other lane. This safety criterion is fulfilled if the

gap on the other lane is larger than the current velocity, and the backwards gap on the other lane is larger than the oncoming vehicle's velocity.

The safety criterion is in fact important in order to maintain laminar traffic [21], an aspect that should not be forgotten if one has spent considerable effort in designing rules for stable laminar high flow traffic on single lanes [9].

3 Dynamics on a graph

A big difference between typical CA models and those used for transportation applications is that the latter typically operate on a **graph**. A graph consists of nodes and links. Nodes for transportation applications have ID-numbers and geographical coordinates. Links connect two nodes, and they have attributes such as speed limit or number of lanes. Obviously, nodes correspond to intersections and links to the roads connecting them.

Traffic on links can be represented through 2d arrays, with one dimension being the length and the other one being the number of lanes, and using the driving models from Sec. 2. The only addition is to include lane changes for plan following, which forms an additional incentive to change lanes as discussed in Sec. 2. The remaining parts of the driving logic concern themselves with intersections.

Intersections An easy way to deal with intersections is to treat intersections as "black boxes" without internal dynamics. In this case, the prioritization is handled when vehicles are about to enter the intersection. There are two important cases: turning movements which are "protected" by traffic signals, and unprotected turns. These will be discuseed in the following.

Protected turns are straightforward, since the signal schedule is assumed to take care of possible conflicts. If vehicles can brake to zero speed in one time step (as is assumed in most CA models for traffic), then a yellow phase is not needed. The only other condition for a vehicle to move through an intersection is that there needs to be space on the outgoing link.

Unprotected turns (yield, stop, merging, etc.) are more advanced. In general, for each turning movement a list of conflicting lanes needs to be available, which is normally generated via pre-processing and is part of the network coding. A vehicle that wants to go through an intersection checks for each conflicting lane if there is a conflicting vehicle, and only if there is none and if in addition the outgoing link has space, then the vehicle can move.

The rules for conflicting lanes are normally treated in terms of gap acceptance, very similar to the safety criterion in lane changing. For example, one can demand that for each interfering lane, a conflicting vehicle needs to be at least $n \times v$ cells away, where n is a small number, and v is the speed of the conflicting vehicle. If the simulation has a time step of 1 sec, then n corresponds to the time gap of the conflicting vehicle in seconds. In reality, this time gap is of the order of 5 sec; in CA-based simulations, we found that 3 sec yields more realistic dynamics.

Unexpected side effects and calibration/validation Sometimes, an arbitrary rule, as plausible as it may be, can have unexpected side effects. For example, $g > n \times v$ means that with $v = 0$ the gap still needs to be larger than or equal to one. In contrast, with $g \geq n \times v$ the turn will be accepted when the gap is zero and the conflicting vehicle is not moving. The resulting differences in fundamental diagrams (see Fig. 3) are enormous. The latter turns out to model "zip-lock" dynamics, which is in fact the desired behavior under congested conditions.

In protected turns during the green phase as well as for unprotected turns which have the priority (such as a freeway link connecting to another freeway link at the position of an off-ramp), care has to be taken that free traffic flows unobstructed through the connection. This means, for example, that for CA logic with $v_{max} > 1$, up to v_{max} cells of the outgoing link need to be considered.

Care has also to be taken when different incoming links compete for space on the same outgoing link. Although in principle the prioritization given by traffic rules should take care of this, in practice such conflicts can rarely be completely avoided, for example because of small network coding errors. In order to have a robust implementation, it is thus desirable that vehicles reserve cells where they intend to go.

This can again lead to unexpected effects. For example, we noticed above that the condition $g \geq n \times v$ is very different from $g > n \times v$ under congested conditions. In TRANSIMS, however, it turns out that there is in fact no difference at all between the two rules. Why is that? The answer is that in TRANSIMS, vehicles with velocity zero on the main road reserve space on the outgoing link on the assumption that they might accelerate to speed one. In consequence, vehicles from the minor road cannot move to that same space, even if it turns out that the vehicle on the major road does not move after all.

In order to find out about such unexpected effects, driving logic should be systematically tested. In fact, there should be standardized test cases that each micro-simulation should do and which should be publicly available, e.g. on the internet. A minimum set of tests would consist of the fundamental diagrams for 1-lane, 2-lane, and 3-lane traffic (such as Fig. 2 right), and for all unprotected merging movements (such as Fig. 3). Such tests should be done with the production version of the code so that differences between the specification and the actual implementation could be detected. These tests should be available as an easy-to-configure run of the software package, and the results should be available on the Internet.

The simplest version of the TRANSIMS driving logic consists in fact of the rules described above. Most discussed variations, such as different gap acceptance at unprotected turns, or different p_{noise}, can be changed by global parameters.

The "roundabout" solution An elegant solution to many of these conflicts is the use of small roundabouts at intersections [22]. The advantage of roundabouts is that the high complexity of interfering lanes of standard intersections is decomposed into smaller sub-units, where in each sub-unit only the conflict between the roundabout and the incoming lane needs to be resolved.

Alternative implementation of graph dynamics The above description assumes that a street network is given as a graph. It was already said that this yields realistic descrip-

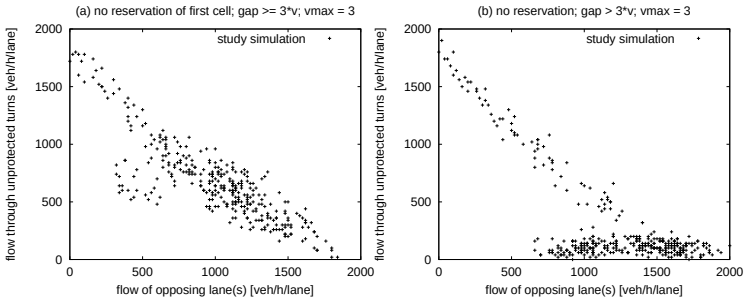

Fig. 3. Different yield rules. LEFT: Vehicles accept turn if $g \geq 3\,v$. RIGHT: Vehicles accept turn if $g > 3\,v$. Note the large difference in the congested regime.

tion at links, but may become problematic at nodes. The software package VISSIM [23] therefore dispenses with nodes and connects links via a second type of links, called connectors. Such connectors start somewhere on a link and end somewhere on a (usually different) link. There is no need that they start at beginnings or ends of links. Any vehicle which encounters an outgoing connector somewhere on the link can decide to go there, or to continue on the link. However, a vehicle need to select one of the connectors eventually, since there is nowhere to go once the vehicle reaches the end of the link.

Similarly, incoming traffic is modeled via connectors which end somewhere on the link, not necessarily at its beginning. There need to be rules which resolve conflicts between incoming vehicles and vehicles which are already on a link.

As a radically different approach, it is possible to dispense with the graph dynamics completely. The whole transportation system then is overlayed by a CA grid structure, and vehicles always move within cells. The typical artifacts with off-axis movement are compensated for by smoothing techniques. CITY-TRAFFIC [24] seems to be using this technology; we use a similar approach for the simulation of tourists hiking in the Alps [25].

4 Moving particles and moving agents

There is a stark difference between typical physics particle hopping models and the transportation models: in transportation, the particles are **intelligent agents**, meaning that they have individual and potentially complex rules of how they react to the environment. This means that in implementations some pointer to these complicated rule structures needs to be maintained when the particles move. This immediately excludes the use of single bit coding, since these pointers typically occupy 32 or even 64 bits of memory.

In consequence, a simple typical vehicle grid in a transportation looks as follows

```
class veh {
    int ID ;
    double Speed ;
```

```
    ...
};
veh* Road[200];  // memory allocation for 200 pointers
```

which means that `Road` consists of 200 pointers pointing to vehicles. Memory for the pointers is immediately allocated; memory for the vehicles is only allocated when it is used. For example, when the code decides to put a vehicle into cell number `ii`, the code fragment may look as

```
Road[ii] = new veh ; // memory allocation for vehicle
Road[ii]->ID = SomeID ;
Road[ii]->Speed = 0. ;
```

Movement is still done in a relatively standard way:

```
speed = int(Road[ii]->Speed) ;
Road[ii+speed] = Road[ii] ;
Road[ii] = NULL ;
```

The big advantage of this is that all information belonging to the vehicle is always being moved with it.

Clearly, many improvements to the above are possible or even recommended, such as using vectors instead of arrays, making clean definitions of constants such as "200", making IDs constant, explicitely defining constructors and destructors, etc.

The currently most important application of the agent technology in transportation simulations is that agents know where they are going. More precisely, it is possible to give each agent its full route, for example consisting of a sequence of nodes. A related but different area of research is to generate those strategic decisions for the agents. All this results in additional computational modules which are part of a complete transportation simulation package (e.g. [26]).

5 Limits of the CA technology and relations to other methods

More realistic representations A standard problem with CA methods is that they may be difficult to calibrate against realistic values. Take for example the STCA as described above in Sec. 2. The length of a cell is straightforward: this needs to be the space a vehicle occupies in a jam in the average. The time step is traditionally taken as 1 sec, which is justified by reaction time arguments [27]. This implies that speeds come in increments of 7.5 m/s; 5 cells per second $=$ 37.5 m/s $=$ 135 km/h is a convenient maximum speed. The remaining free parameter, p_{noise}, is now selected such that the maximum flow comes out at 2000 veh/sec; this results in $p_{noise} = 0.2$. Lane changing rules can be calibrated similarly, and can even reproduce the density inversion which happens on German freeways when they are close to capacity [28].

So far so good. The problems start if for some reason the above is not good enough. For example, the existing speed classes are not fine enough to resolve a difference between a 55mph and a 50mph speed limit, a common occurence in the U.S. Similarly, although the fundamental diagram comes out plausibly, acceleration of vehicles turns out to be too high, which is a problem for emissions calculations.

And it is difficult to resolve those problems via a clever choice of the probability p_{noise}. For example, increasing p_{noise} leads to lower acceleration (which is desired),

but also to lower throughput (which is not desired). A possible way out is to have p_{noise} dependent on the velocity: A small p_{noise} at low velocities together with a large p_{noise} at high velocities leaves the fundamental diagram nearly unchanged while leading to a much lower average acceleration. However, unfortunately such measures also change the fluctuations of the system – for example, such a reduced acceleration will lead to a much wider spread of the times that vehicles need to accelerate from 0 to full speed. Also note that in slow-to-start models, the modifications of p_{noise} are exactly the other way round.

As an alternative, it would be possible to make the resolution of the cells finer, for example to introduce cells of length 3.75 m and make vehicles occupy two cells. It is unclear if this would be worthwhile; it would certainly be slower than the standard method because twice the number of cells needs to be treated.

A possible method that seems to work well in many cases in practice are hybrid simulations. Here, one leaves the cellular structure intact, but allows for offsets of particles against the cellular structure. For directional traffic, it seems that one can ultimately completely dispense with the grid and work with a method that still has a 1 sec time resolution but a continuous resolution in space [27]. The reason why this works for traffic is that it is computationally relatively cheap to keep track of neighbors since a link is essentially one-dimensional. For higher-dimensional simulations, keeping some cellular structure is normally advantagous for that task alone – see for example the parallel code for molecular dynamics which turned out to also handle the problem of neighbor finding very efficiently.

(Even) less realistic representations Another problem with microscopic simulations often is that the necessary input data is not available. For example, for a CA-based traffic microsimulation one would need at least the number of lanes and some idea about the signal schedules. Most transportation network databases, in particular if they were put together for transportation planning, only contain each link's capacity. It is difficult to construct CA links so that they match a given capacity. The only way seems to be a heuristic approach, by selecting the right number of links and then to restrict the flow on the link for example by a (fake) traffic light. Still, this leaves many questions open. For example, signals phases need to be coordinated so that not two important incoming links try to feed into the same outgoing link at the same time. Furthermore, from the above it is not clear which incoming lane feeds into which outgoing lane (lane connectivities).

In consequence, there are situations where a CA representation is still too realistic, and a simpler representation is useful. A possibility to do this is the queue model. This is essentially a queuing model with added queue spillback. Links are characterized by free speed travel time, flow capacity, and storage capacity. Vehicles can enter links only when the storage capacity is not exhausted. Vehicles which enter a link need the free speed travel time to arrive at the other end of the link, where they will be added to a queue. Vehicles in that queue are moved accross the intersection according to the capacity constraint, and according to availability of space on the next link.

This describes only the most essential ingredients; care needs to be taken to obtain fair intersections and for parallelization [29]. Also, there are clearly unrealistic aspects of the queue model, such as the fact that openings at the downstream end of the link

are immediately transmitted to the upstream end. This has for example the consequence that queue resolution looks somewhat unrealistic: queues break up along their whole length simultaneously, instead of from the downstream end. Nevertheless, the queue simulation is an excellent starting point for large scale transportation simulations.

6 A simulation of all of Switzerland

One of the current main goals in our group is a simulation of "all of Switzerland". By this we mean a microscopic 24h simulation of a typical workday of all traffic in Switzerland. Fig. 4 contains an early result of this.

The network that is used was originally developed for the Swiss regional planning authority (Bundesamt für Raumentwicklung), and has since been modified by Vrtic at the IVT and by us. The network has the fairly typical number of 10 572 nodes and 28 622 links. Also fairly typical, the major attributes on these links are type, length, speed, and capacity.

Demand is obtained from a 24-hour origin-destination matrix with 3066 zones, also from the Bundesamt für Raumentwicklung. This matrix is converted to 24 separate hourly matrices by a several-step heuristic. In the long run, it is intended to move to activity-based demand generation. Then, as explained above one would start from a synthetic population, and for each population member, one would generate the chain of activities for the whole 24-hour period.

Routes are obtained via iterations between simulation and time-dependent fastest path routing. The simulation behind Fig. 4 is the queue simulation as described in Sec. 5.

7 Summary

This paper has outlined the most important elements of CA use in transportation applications. Besides the standard CA rules of "traffic on a link", the important aspects are, that the dynamics unfolds on a graph instead of on flat space, and that the particles are intelligent. Both aspects make simulation packages considerably more complicated, the first since intersections need to be modeled; the second since the "intelligence" of the travelers (route choice, destination choice, activity generation, etc.) needs to be modeled. Finally, the limits of the CA technology were discussed. These limits exist in two directions: (1) The driving logic of the CA rules may not be realistic enough, and making it more realistic may be computationally as expensive as moving to coupled map lattices (discrete time, contiuous state space). (2) The available real world data may not be detailed enough to feed a realistic CA-based micro-simulation.

Acknowledgments This paper and in particular the work on the simulation of "all of Switzerland" would not have been possible without the help of Kay Axhausen, Nurhan Cetin, Christian Gloor, Bryan Raney, Milenko Vrtic, Res Voellmy, and Marc Schmitt.

References

1. S. Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, University of Cologne, Germany, 1997. See www.zpr.uni-koeln.de.

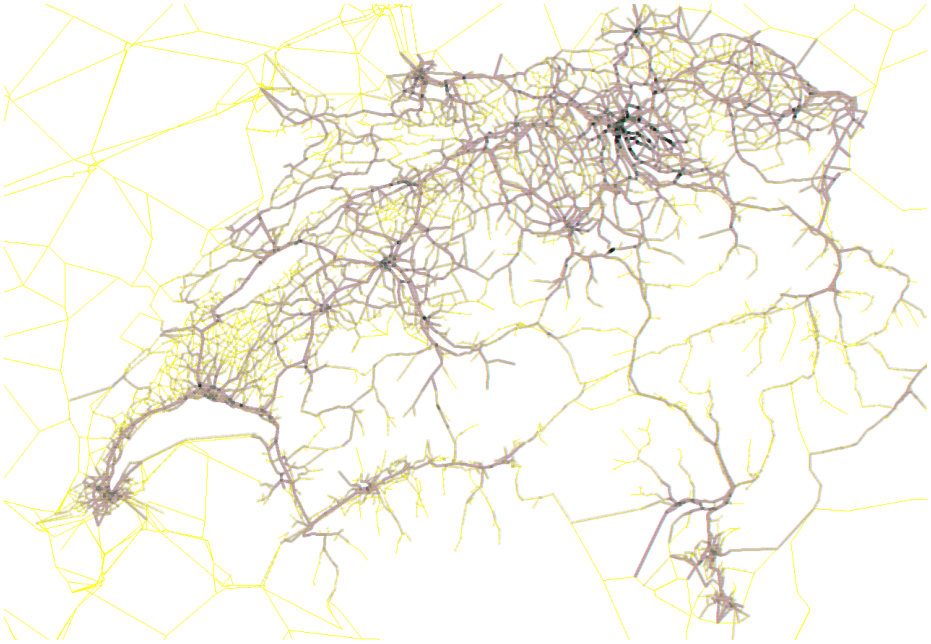

Fig. 4. Most of Switzerland at 8am, simulation result. The graphics shows individual vehicles, but they are so small that they cannot be distinguished. Areas in dark contain traffic jams.

2. K. Nagel and H. J. Herrmann. Deterministic models for traffic jams. *Physica A*, 199:254, 1993.
3. S. Wolfram. *Theory and Applications of Cellular Automata*. World Scientific, Singapore, 1986.
4. K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221, 1992.
5. D. Chowdhury, L. Santen, and A. Schadschneider. Statistical physics of vehicular traffic and some related systems. *Physics Reports*, 329(4–6):199–329, May 2000.
6. K. Nagel, P. Stretz, M. Pieck, S. Leckey, R. Donnelly, and C. L. Barrett. TRANSIMS traffic flow characteristics. Los Alamos Unclassified Report (LA-UR) 97-3530, Los Alamos National Laboratory, see transims.tsasa.lanl.gov, 1997.
7. B. S. Kerner and H. Rehborn. Experimental features and characteristics of traffic jams. *Physical Review E*, 53(2):R1297–R1300, 1996.
8. B. S. Kerner and H. Rehborn. Experimental properties of complexity in traffic flow. *Physical Review E*, 53(5):R4275–R4278, 1996.
9. R. Barlovic, L. Santen, A. Schadschneider, and M. Schreckenberg. Metastable states in cellular automata. *European Physical Journal B*, 5(3):793–800, 10 1998.
10. D. Chowdhury, L. Santen, A. Schadschneider, S. Sinha, and A. Pasupathy. Spatio-temporal organization of vehicles in a cellular automata model of traffic with 'slow-to-start' rule. *Journal of Physics A – Mathematical and General*, 32:3229, 1999.
11. W. Brilon and N. Wu. Evaluation of cellular automata for traffic flow simulation on freeway and urban streets. In W. Brilon, F. Huber, M. Schreckenberg, and H. Wallentowitz, edi-

tors, *Traffic and Mobility: Simulation – Economics – Environment*, pages 163–180. Aachen, Germany, 1999.
12. D.E. Wolf. Cellular automata for traffic simulations. *Physica A*, 263:438–451, 1999.
13. C. L. Barrett, M. Wolinsky, and M. W. Olesen. Emergent local control properties in particle hopping traffic simulations. In D.E. Wolf, M. Schreckenberg, and A. Bachem, editors, *Traffic and granular flow*, pages 169–173. World Scientific, Singapore, 1996.
14. M. Takayasu and H. Takayasu. Phase transition and $1/f$ type noise in one dimensional asymmetric particle dynamics. *Fractals*, 1(4):860–866, 1993.
15. B. S. Kerner and P. Konhäuser. Structure and parameters of clusters in traffic flow. *Physical Review E*, 50(1):54, 1994.
16. L. Roters, S. Lübeck, and K.D. Usadel. Critical behavior of a traffic flow model. *Physical Review E*, 59:2672, 1999.
17. M. Sasvari and J. Kertesz. Cellular automata models of single lane traffic. *Physical Review E*, 56(4):4104–4110, 1997.
18. D. Chowdhury et al. Comment on: "Critical behavior of a traffic flow model". *Physical Review E*, 61(3):3270–3271, 2000.
19. K. Nagel, Chr. Kayatz, and P. Wagner. Breakdown and recovery in traffic flow models. In Y. Sugiyama et al, editor, *Traffic and granular flow '01*. Springer, Heidelberg, in press.
20. K. Nagel, D.E. Wolf, P. Wagner, and P. M. Simon. Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E*, 58(2):1425–1437, 1998.
21. M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour. Two lane traffic simulations using cellular automata. *Physica A*, 231:534, 1996.
22. A. Dupuis and B. Chopard. Cellular automata simulations of traffic: a model for the city of Geneva. *Networks and Spatial Economics*, forthcoming.
23. Planung Transport und Verkehr (PTV) GmbH. See www.ptv.de.
24. City-traffic. Fraunhofer Institut Autonome Intelligente Systeme (AIS). See www.citytraffic.de.
25. See www.inf.ethz.ch/~nagel/projects/alpsim.
26. K. Nagel, J. Esser, and M. Rickert. Large-scale traffic simulation for transportation planning. In D. Stauffer, editor, *Annual Reviews of Computational Physics*, pages 151–202. World Scientific, 2000.
27. S. Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, University of Cologne, Germany, 1997. See www.zpr.uni-koeln.de.
28. P. Wagner. Traffic simulations using cellular automata: Comparison with reality. In D E Wolf, M.Schreckenberg, and A.Bachem, editors, *Traffic and Granular Flow*. World Scientific, Singapore, 1996.
29. N. Cetin and K. Nagel, in preparation.