International Journal of Modern Physics C, Vol. 8, No. 5 (1997) 1025–1036

MICROSCOPIC SIMULATION OF URBAN TRAFFIC BASED ON CELLULAR AUTOMATA

J. ESSER and M. SCHRECKENBERG
Fachbereich 10/Physik-Technologie
Gerhard-Mercator-Universität-Duisburg
47048 Duisburg/Germany
E-mail: esser@comphys.uni-duisburg.de

Received 21 May 1997
Revised 2 June 1997

Saturated capacities in traffic systems evoke increasing interest in simulations of complex networks serving as laboratory environment for developing management strategies. Especially for urban areas questions concerning overall traffic control have to be considered with regard to their impacts on the whole network. Modeling traffic flow dynamics using cellular automata allows us to run large network traffic simulations with only comparatively low computational efforts. We present a traffic simulation tool for urban road networks which is based on the Nagel–Schreckenberg Model. Arbitrary kinds of roads and crossings are modeled as combinations of only a few basic elements. Furthermore parking capacities are considered as well as circulations of public transports. The vehicles are driven corresponding to route plans or at random depending on the available data. The application of this network simulation covers investigations on the field of traffic planning as well as online simulations based on real-time traffic data as basis for dynamic traffic management systems.

Keywords: Traffic; Cellular Automata; Complex Systems.

1. Introduction

While for globally aimed questions in the field of traffic management like dynamic vehicle routing it is self-evident to be considered on the whole network, even local aspects like changes in priority regulation at single crossings have to be investigated on large scales to avoid undesired global effects due to mutual correlations. Testing control strategies in reality is usually infeasible or at least extremely demanding in time and costs; therefore, the application of simulation tools as laboratory environment is desired.

Designing a simulation tool for network traffic requires an overall compromise: To facilitate collection of individual vehicle data like travel times or number of stops due to priority rules, a microscopic approach is appropriate. In addition one may have to minimize running times, since, e.g., in checking out different vehicle routing or traffic light control strategies complex scenarios have to be simulated again and again such that large computation times are very cumbersome and inefficient. Furthermore, for online simulations based on real-time traffic data run speeds of at least real-time are a necessary requirement.

We present a microscopic simulation tool for vehicular traffic in urban road networks, which was developed in the framework of the Northrhine Westfalia Cooperative FVU[a].[1] The underlying traffic flow dynamics is based on the cellular automaton introduced by Nagel and Schreckenberg.[2] This approach has proven to be very efficient: It is possible to simulate even the whole German autobahn network in real-time[3,4] and within the project TRANSIMS[5] microscopic traffic simulations are carried out for the Dallas/Fort Worth area.[6]

The outline of this paper is as follows: First the cellular automaton model is briefly described. In the main part we focus on the simulation tool and explain the essential elements like network representation, vehicle guidance and measure instruments. The final chapter contains some remarks on a current application of the simulation tool for the city of Duisburg; a more detailed presentation of this project can be found in Ref. 7.

2. The Cellular Automaton Model

The Nagel–Schreckenberg model[2] was originally defined on a single-lane road. The road is subdivided into cells, which can be either empty or occupied by one vehicle. Every vehicle has a non-negative integer velocity. For one update of the road the following four steps are performed simultanously for all vehicles:

1. Acceleration: $\mathsf{v} \longrightarrow \min(\mathsf{v}+1, \mathsf{v}_{\max})$
2. Avoiding crashes: $\mathsf{v} \longrightarrow \min(\mathsf{v}, \mathsf{gap})$
3. Randomization: if rand() $< \mathsf{p}_{\mathrm{dec}}$ then $\mathsf{v} \longrightarrow \max(\mathsf{v}-1, 0)$
4. Update: Each vehicle is advanced v cells.

Here gap denotes the number of empty cells in front of a vehicle and $\mathsf{v}_{\max}$ the maximum velocity. The randomization (Step 3) takes into account that individual driving behaviors for different vehicles result in non-deterministic dynamics of vehicle motions in reality. The resolution is taken to be 7.5 m per cell representing the average space a vehicle takes in a jam, but can be suitably adjusted with regard to the considered problem[8]; using this cell width each time step in the simulation corresponds to one second in reality. For modeling traffic on multi-lane roads a set of rules for lane changes is added to the fundamental rules.[9–11]

Detailed investigations showed that despite its simplicity the cellular automaton model is capable of reproducing macroscopic traffic flow features including realistic lane changing behaviors.[9,12] Also a continuous extension[13] of the basic discrete model and a modified set of rules resulting in more realistic breaking and accelerating behavior[14,15] was developed to enable modeling of additional more detailed vehicle features (e.g., exhaust emission).

[a] Forschungsverbund Verkehrssimulationen und Umweltwirkungen.

3. Simulation Elements

The overall simulation tool consists of different elements. In this section these are explained in detail.

3.1. Network representation

The road network is described as a composition of nodes and edges representing crossings and roads. To avoid any misunderstanding: Whenever we use the expression "edge" we refer to directed edges representing one direction of motion on a road; i.e., one road usually consists of two (oppositely directed) edges. The basic idea is to define different types of edges. This approach causes arbitrary complex crossing types to be represented as combinations of at most three basic edge types described now.

Single-lane edge: Simply, here the basic cellular automaton is applied. In addition there can be connected turning sections for each direction at the end of the edge. Vehicles that have to use such a section drive on to the beginning of it and if the vehicle is hindered in entering, it has to wait at a predefined position until a change is possible.

Multi-lane edge: Especially at the end of multi-lane roads the lane changing behavior strongly depends on the desired driving direction. For this purpose, multi-lane edges are subdivided into different regions, which are exemplarily shown for the case of two lanes in Fig. 1. Up to the position defined by the length DirectionChange lane changes are unconstrained. In the following section only changes according to the destination directions are allowed. Here the probability for risky lane changes (i.e., only the neighboring site is checked) increases approaching the change lock position, where vehicles have to wait if they are not on the destination lane. Waiting vehicles at the change lock positions can cause deadlocks, when two hinder each other in changing the lane; these deadlocks are cleared by exchanging the vehicles.

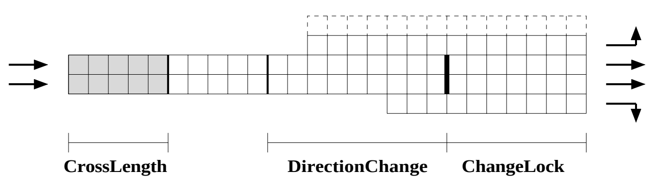

Fig. 1. Structure of a two-lane edge with additional turning sections.

Transfer edge: Usually highways also play an important role for a realistic description of urban traffic. Therefore an additional element is incorporated to simulate highway drive-ups and also crossings with large spatial extension. Transfer edges are one- or multi-lane edges which merge in a destination edge (Fig. 2). The vehicles drive on up to the merge section at the end of the transfer edge, where they

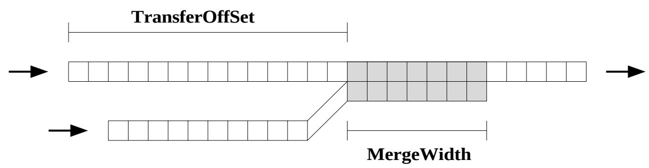

Fig. 2. Structure of a transfer edge for realizing highway drive-ups and crossings with large spatial extension.

change under the condition that there is sufficient distance to the successor on the destination edge. If hindered while changing they drive on to the end of the transfer road and if necessary wait. The length TransferOffSet denotes the position on the destination road where the merge section starts.

Figure 3 shows a crossing as combination of different edge types. The numbers at the end of the edges represent signal groups of the corresponding traffic lights and will be used below. Optionally the first cells on an edge can be defined as the crossing section (shaded in the figure) to model the spatial extension roughly. In this case for each possible direction it is predefined if vehicles entering the edge are put onto the first cell of the edge or beyond the crossing section. This also allows us to consider cases of jammed crossing regions in combination with priority rules described in the following section.

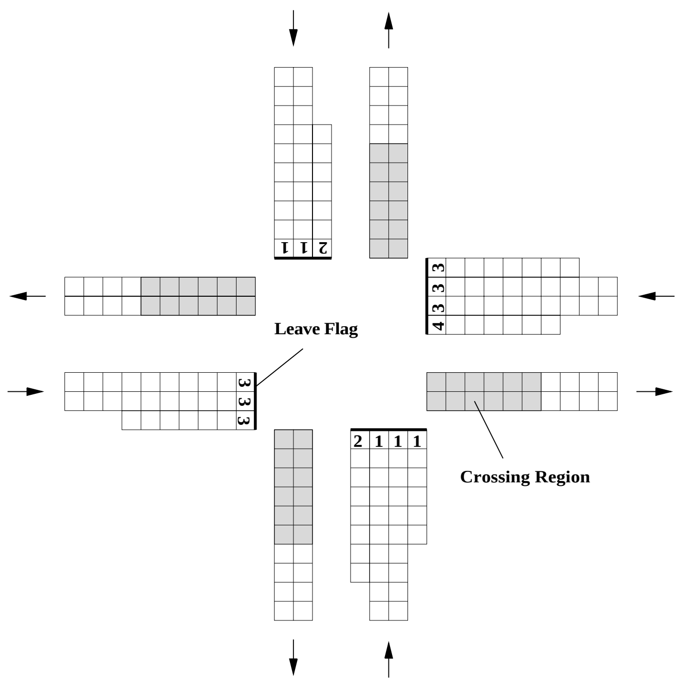

Fig. 3. Representation of crossings.

3.2. Priority regulation

One striking feature of urban road networks is the high density of intersections, which causes essential correlations among tra c streams on di erent edges. It is important for the simulation to have realistic throughputs at the crossings, which again requires us to consider complex priority rules and especially tra c lights in a realistic manner.

Each edge in the network has a driving-direction-dependent leave flag at its end, by which vehicles may be prevented from advancing any further. Tra c lights are realized by switching these flags corresponding to prede ned time tables. In addition a hierarchy can be de ned to consider crossings where special directions are controlled by more than one light (e.g., occasionally active green-light arrows for right turns). To facilitate handling of tra c lights the simulation tool contains an editor for setting green and red phases within a switch matrix. Figure 4 shows an exemplary switch matrix for the crossing in Fig. 3, where the signal groups (i.e., groups of synchronously switched tra c lights) are numbered with regard to the numbers at the end of the edges. The tra c lights are switched with an overall program period of 50 seconds and each second is represented by an empty or dark shaded rectangle corresponding to green and red signals respectively. There exist no yellow signals in the simulation; these are interpreted as green.

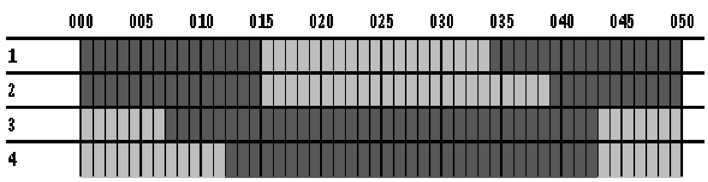

Fig. 4. Tra c light switch matrix.

An essential parameter with regard to crossing throughputs is the time headway between two vehicles crossing the stop line after a tra c light switched to green. It is assumed to be about two seconds, but varies (0:5 sec) with the road gradient, driving direction, lane width, weather and the vehicle position in the queue.[16] Already application of the basic discrete version of the Nagel{Schreckenberg model allows us to reproduce realistic time headways. Figure 5 shows the headways plotted against the deceleration probability p_{dec} (maximum velocity v_{max} = 2) measured for queues with an average length of 10 vehicles standing bumper-by-bumper before the tra c light switches green. In reality the gap even increases for the rst vehicles in the queue.[17] Also this e ect can be reproduced using a higher resolution for the cellular automaton, but is usually assumed to be negligible. Altogether with regard to Fig. 5, deceleration probabilities of about $p_{dec} = 0{:}2$ yield realistic time headways.

Priority rules are realized by switching leave flags depending on vacant cells on other edges. Here an additional important parameter is the number gap_{prior} of cells

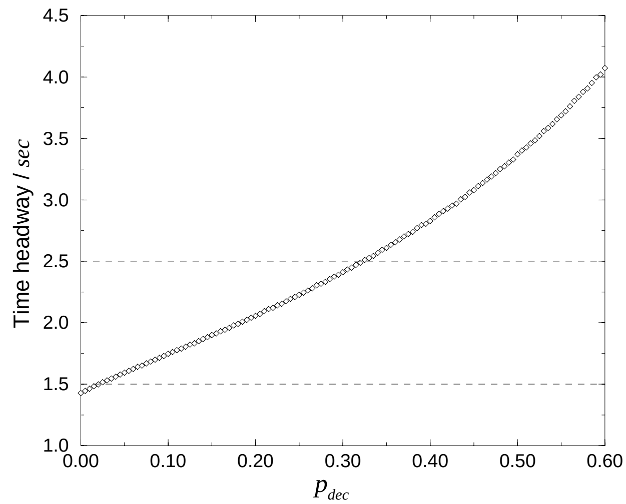

Fig. 5. Time headway between vehicles crossing the stop-line at a tra c light.

to be scanned for checking priorities in units of v_{max}. For this aim, we consider a non-priority (minor) tra c stream crossing a priority (major) stream. In Fig. 6 the maximum minor flow q_{minor} is plotted against the major flow q_{major} for di erent gap_{prior} (v_{max} = 2 and p_{dec} = 0:2). Comparison with corresponding relations for real tra c in Refs. 16 and 18 leads to values for gap_{prior} = 4 in the range of 4 − 6 for realistic gap acceptance. It slightly depends on the value of q_{major}, but is assumed to be constant in the simulation.

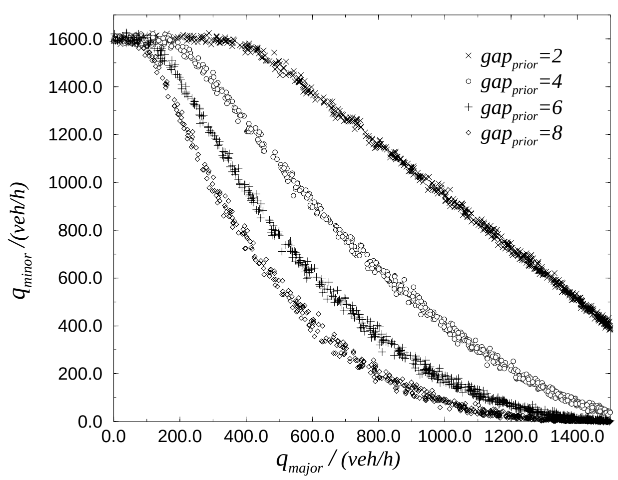

Fig. 6. Maximum flow of a non-priority stream crossing a priority stream at an unsignalized intersection.

It is possible that vehicles waiting at a crossing hinder each other in advancing further; in this case one of these is randomly chosen to drive on offending priority rules to clear the deadlock. A similar problem arises if a vehicle has priority, but cannot drive on (e.g., due to red traffic light or a crowded destination edge); then its priority is neglected for other vehicles.

In reality turning vehicles are often hindered in driving further, e.g., by pedestrians crossing the destination road, which can reduce the throughput at certain crossings essentially. This effect is considered by blocking turning vehicles, for which driving on is allowed by traffic lights and priority rules, with predefined probabilities. Furthermore no-stop and stop priority rules are distinguished; for the latter it is additionally checked if vehicles had zero velocity standing on the last cell of the edge for at least one time step before they are allowed to drive on.

One essential problem with regard to the implementaion of a network simulation is the large number of priority rules. Therefore the network editor if necessary automatically derives priority rules from the geometric node structure (i.e., angles between intersecting edges) as far as it is clearly defined.

3.3. Vehicle guidance

One of the fundamental questions with regard to network traffic simulations concerns the question of how the vehicles are guided through the network. In a randomly driven simulation at each crossing destination directions for the vehicles are choosen at random corresponding to turn counts. Running the simulation route plan driven means that each vehicle makes its turning decision corresponding to individual route plans, which are derived from origin-destination information. In the simulation the guidance mode is specified for each vehicle separately. Combination of both for example allows us to check routing strategies on random background traffic representing typical traffic states.

3.4. Vehicle types

The vehicles are characterized by their maximum speed, length (number of occupied cells) and a probability for carrying out risky lane changes (i.e., lane changes without considering safety distances). In the present version the deceleration probabilities are not individual, but attributed to the edges to tune throughputs at the crossings (see Sec. 3.2).

There are two special kinds of vehicles: Vehicles can be guided periodically along predefined routes following time tables to simulate public transport. Additional bus or tram stops are defined, where these have to wait for a certain time and in this case are taken out of the network for that time. The second type covers hindrances like accidents or road works. These are represented by vehicles of special length and $v_{max} = 0$.

3.5. Sources and sinks

An essential characteristic of urban traffic is the fact that there are no predefined sources and sinks; rather vehicles are allowed to enter or leave the network at nearly every arbitrary position. For that reason sources and sinks can be linked to every cell in the network. At sources vehicles are characterized by type, guidance mode (see Sec. 3.3) and — if necessary — additional information like if the vehicle follows a dynamic routing system or not. They are inserted corresponding to source rates or trip plans; the latter means that at the beginning a list of vehicles with departure times, vehicle characteristics and route plans is prepared.

In addition flow check points as source-sink combination where the number of vehicles is adjusted to counting loop data are located in the network to adjust the number of vehicles to counting loop data. For these check points statistics covering number of added and deleted vehicles are collected separately. It should be mentioned that tuning traffic flows by adding or deleting vehicles in traffic streams especially for complex road networks is not trivial but a topic of current research.[7,19]

Route plan driven vehicles are taken out of the network after reaching the destination node. Probabilities for leaving the network at the next node are attributed to every edge for randomly driven vehicles. Especially for urban areas the consideration of parking capacities is desired. In the simulation these are considered as special source-sink pairs, which are filled up with regard to predefined capacities and from which — as far as they are not empty — vehicles can enter the network. This allows us to develop strategies for parking control systems. For the city of Duisburg it is planned to connect the simulation directly to a traffic monitoring system, which provides the vehicle numbers of the main parking capacities (parking-garages etc.) to incorporate these sources directly in the online simulation. It is also possible to define parking-lanes on the roads, which are used as usual lanes and in addition vehicles can be put on them and marked as parking. This feature incorporates capacity reduction of roads due to parking vehicles.

3.6. Measuring instruments

A necessary requirement for traffic management systems is information about the traffic state. For this purpose, three instruments are incorporated in the simulation: At measure points local data like number and average speed of vehicles are collected separately for different vehicle types and averaged over predefined time intervals. Edges can also be grouped together representing measure regions, by which global measurements are carried out. The individual travel time is measured for every vehicle in the network. For more detailed information it is possible to guide vehicle probes through the network. This provides for the collection of additional point to point data for special routes (e.g., data with regard to the driving comfort like number and duration of stops due to priority rules and traffic lights). Furthermore overall network data like number of vehicles, average speed and edge usage are stored.

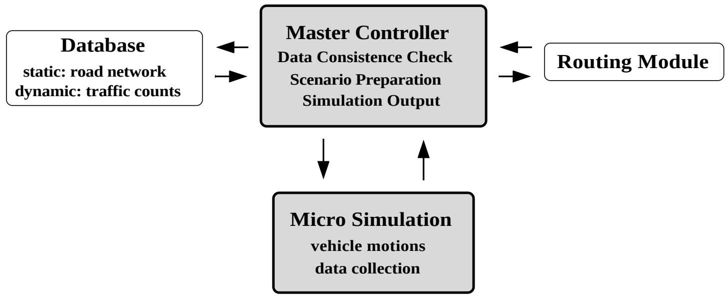

Fig. 7. Overall design.

3.7. Simulation control

The overall simulation tool consists of two principal processes (Fig. 7).

The master controller maintains the overall coordination: It checks static and dynamic network data read from the database for consistency and initializes the scenarios. During the simulation it receives and updates time dependent data like turn counts, handles the simulation output including updates of the graphics and | if necessary | provides dynamic data to the routing module, which updates of route plans for the vehicle guidance system. The actual network dynamics is carried out by the micro simulation process, i.e., vehicle motions, source and tra c light updates and data collection for statistics.

The advantage of this subdivision is twofold: On the one hand it speeds up the simulation, since in addition the micro simulation can be parallelized independently. Furthermore in practice it facilitates overall handling, because it is possible to re-arrange e.g., graphical output or data formats without caring about the complex structure of the micro simulation.

4. Application

Currently tra c simulations for the inner city of Duisburg are developed applying the simulation tool. Here we focus on presenting the present tra c state using online tra c data stemming from induction counting loops provided by the tra c control center of the municipality and updated every minute. Figure 8 shows the considered network covering 107 nodes, 280 edges; the total lane length amounts to about 165 kilometers (22059 cells). Inclusive of online source and turn count updates, priority regulation and data collection for statistics it is possible to simulate in 20 minutes on a PC (Pentium P133) a whole day of typical tra c.

Figure 9 shows the number of vehicles and the average speed during an online simulation for the time interval from 12.23 am on the 1st until 12.22 am on the 2nd of April in 1997. The extended minimum in the night and the two maxima at about 8 am and 5 pm are well discerned. Due to missing origin-destination information with appropriate time resolution the simulation is driven randomly, while

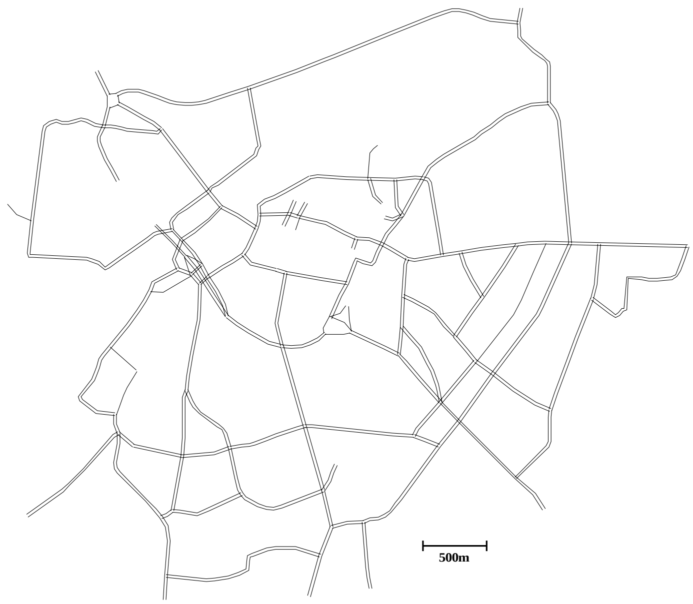

Fig. 8. Simulated road network for the city of Duisburg.

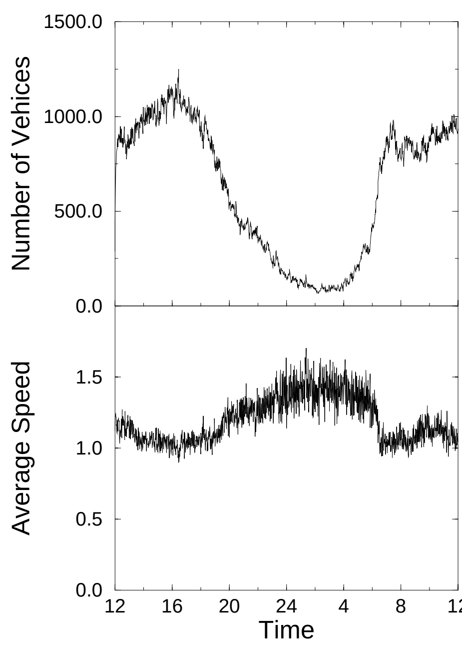

Fig. 9. Number of vehicles in the network and average speed during an online simulation run.

turn counts are derived from real-time tra c data. The present tra c state for the city of Duisburg is presented in the Internet.[20] For more details with regards to the online simulation and the problem of reproducing tra c states in complex networks based on local tra c counts see Ref. 7.

5. Summary

We presented a simulation tool for urban tra c. The microscopic dynamics based on the Nagel{Schreckenberg cellular automaton permits the simulation of large networks in multiple real-time. Within the network model complex crossings including realistic tra c lights and priority rules are considered as well as parking capacities and circulations of public transports. In combination with real-time tra c counts this tool serves as a useful laboratory environment for designing and checking dynamic tra c management systems.

Acknowledgments

The authors are thankful to C. Gawron, S. Krau, M. Rickert and P. Wagner of the tra c group at the center for parallel computing in Cologne and the German Aerospace Research Establishment, and K. Nagel from the TRANSIMS project at the Los Alamos National Laboratory for support with their experience on this topic by many fruitful discussions and hints.

This work was supported by the Northrhine Westfalia Cooperative (Forschungsverbund Verkehrssimulation und Umweltwirkungen, FVU).

References

1. NRW-FVU Home Page, http://www.zpr.uni-koeln.de/Forschungsverbund-Verkehr-NRW/.
2. K. Nagel and M. Schreckenberg, *J. Physique I* 2, 2221 (1992).
3. M. Rickert and P. Wagner, *Int. J. Mod. Phys.* C7(2), 133 (1996).
4. PAMINA (PArallel MIcroscopic Network Algorithm) Home Page, http://www.zpr.uni-koeln.de/ mr/PAMINA/.
5. TRANSIMS Home Page, http://studguppy.tsasa.lanl.gov.
6. M. Rickert and K. Nagel, *Int. J. Mod. Phys.* C8, 3, 483 (1997).
7. J. Esser and M. Schreckenberg, Deriving Tra c State Information in Urban Road Networks with Microscopic Simulations Based on Local Measurements, in preparation.
8. Ch. L. Barrett, S. Eubank, K. Nagel, S. Rasmussen, J. Riordan, and M. Wolinsky, Issues in the Representation of Tra c Using Multi-Resolution Cellular Automata, Los Alamos National Laboratory Technical Report: LA/UR 95-2658.
9. M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour, *Physica* A231, 534 (1996).
10. K. Nagel, P. Wagner, and D. Wolf, Lane-changing rules in two-lane tra c simulation using cellular automata: II. A systematic approach, in preparation.
11. P. Wagner, K. Nagel, and D. Wolf, *Physica* A234, 687 (1996).
12. P. Wagner, Tra c Simulations Using Cellular Automata: Comparison with Reality, in *Tra c and Granular Flow*, D. E. Wolf, M. Schreckenberg, and A. Bachem (eds.) (World Scienti c, Singapore, 1996).
13. S. Krau, P. Wagner, and C. Gawron, *Phys. Rev.* E54, 3707 (1996).

14. S. Krau, P. Wagner, and C. Gawron, Metastable states in a microscopic model of tra c flow, to appear in *Phys. Rev.* E55, 5 (1997).
15. S. Krau, Towards a uni ed view of microscopic tra c flow theories, to be presented at IFAC symposium 97 *Transportation Systems*.
16. W. Brilon, M. Gro man n, and H. Blanke, Verfahren für die Berechnung der Leistungsfähigkeit und Qualität des Verkehrsablaufes auf Stra en, Schriftenreihe *Forschung Straenb au und Stra enverkehrste chnik*, Heft 669 (1994).
17. M. Ahn, Veränderung der Leistungsfähigkeit städtischer Hauptverkehrsstraen über die Tageszeit, Schriftenreihe Lehrstuhl für Verkehrswesen, Ruhr-Universität Bochum, Heft 4 (1987).
18. Highway Capacity Manual, Transportation Research Board, Special Report *Tra c Flow Theory*, http://stargate.ornl.gov/trb/tft.html.
19. S. Krau, in preparation.
20. Duisburg Online Simulation Home Page,
http://www.comphys.uni-duisburg.de/OLSIM/.