Very Fast Traffic Simulation

and

Individual Routing using Parallel Supercomputers

Achim Bachem
Christoph Moll
Kai Nagel
Thomas Pfenning
Marcus Rickert

email: mr@zpr.uni-koeln.de

http://www.zpr.uni-koeln.de/GroupBachem/VERKEHR.PG/

EURO XIV

Section WEA 5.1

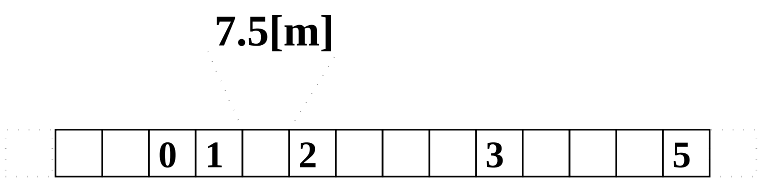
Center for Parallel Computing
University of Cologne and

Topics

- Underlying CA Model
- Fast Single Lane Net Simulation
  Using Route Plans
- Extended CA Features
  (Multi Lane, Route Plans)
- Performance
- Outlook

Single Lane CA

Nagel / Schreckenberg



figures denote vehicles with v=0,1,...,5

1 Accelerate:

$$v := \max(v_{max}, v+1)$$

2 Avoid crash:

$$v := \min(gap, v)$$

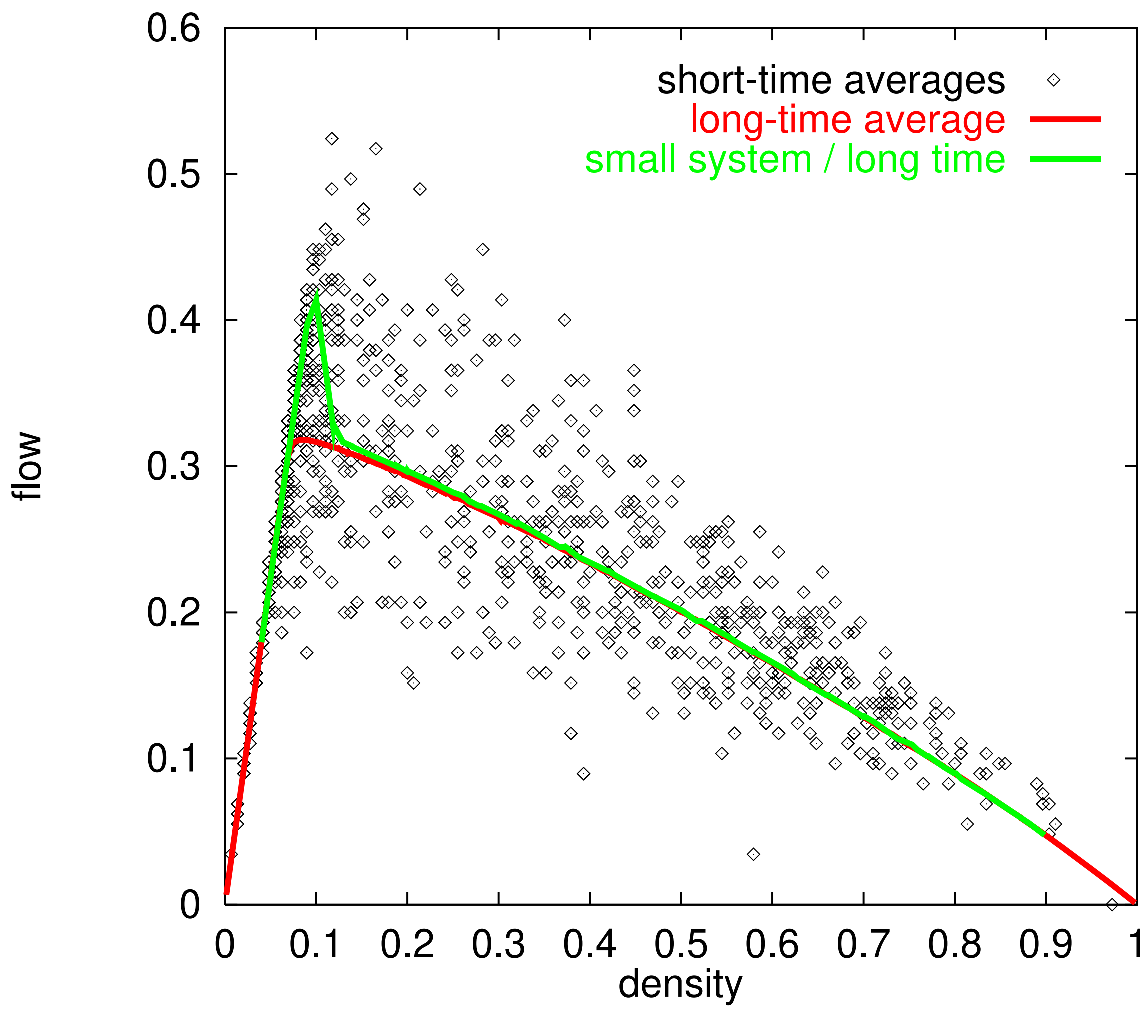
3 Randomize:

$$rand() < p_{dec} \Rightarrow v := \max(v-1, 0)$$

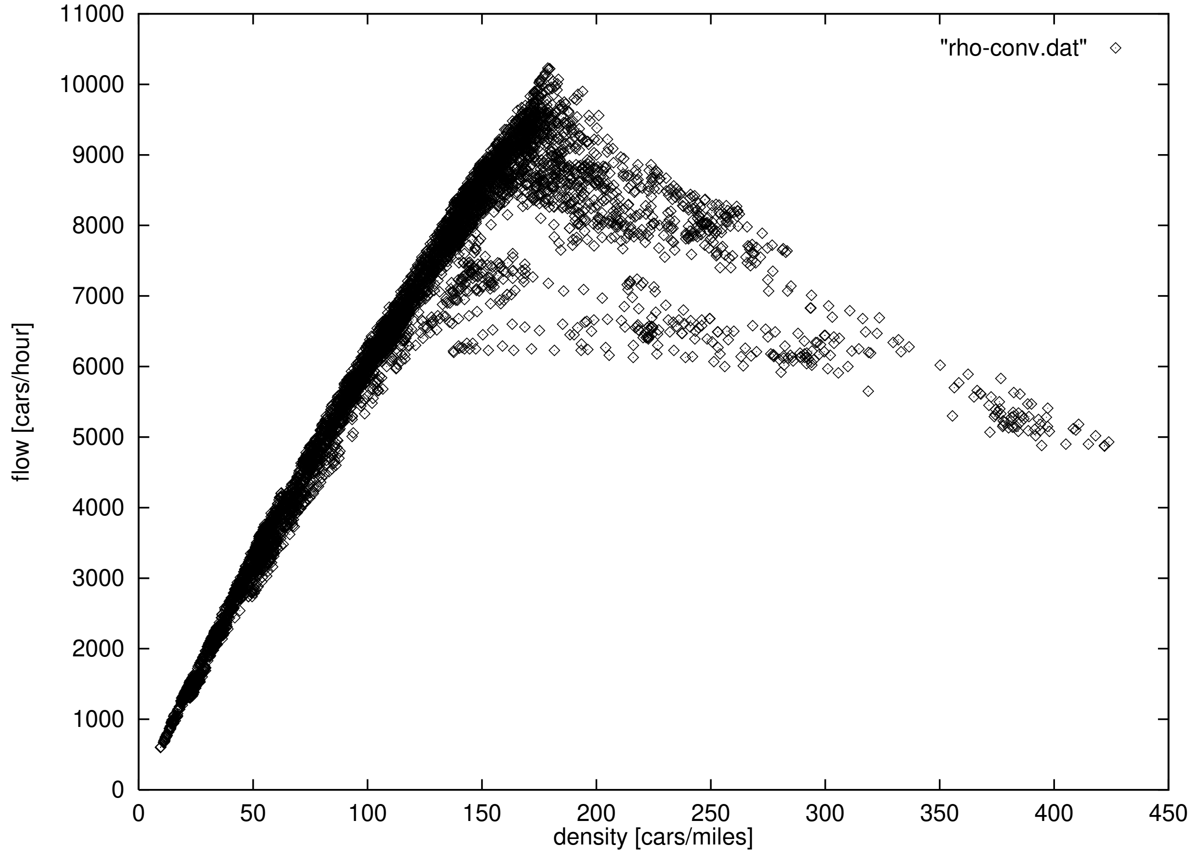
- Perform parallel update

Fundamental Diagram

system setup	sites	steps
large	10,000	1,000,000
scatter	10,000	100
finite size	100	100



Real Life Traffic



Data from a Californian Highway

Models

$$\text{SL} = \text{Microscopic single lane}$$
$$\text{ML} = \text{Microscopic multi lane}$$
$$\text{MESO} = \text{Mesoscopic}$$

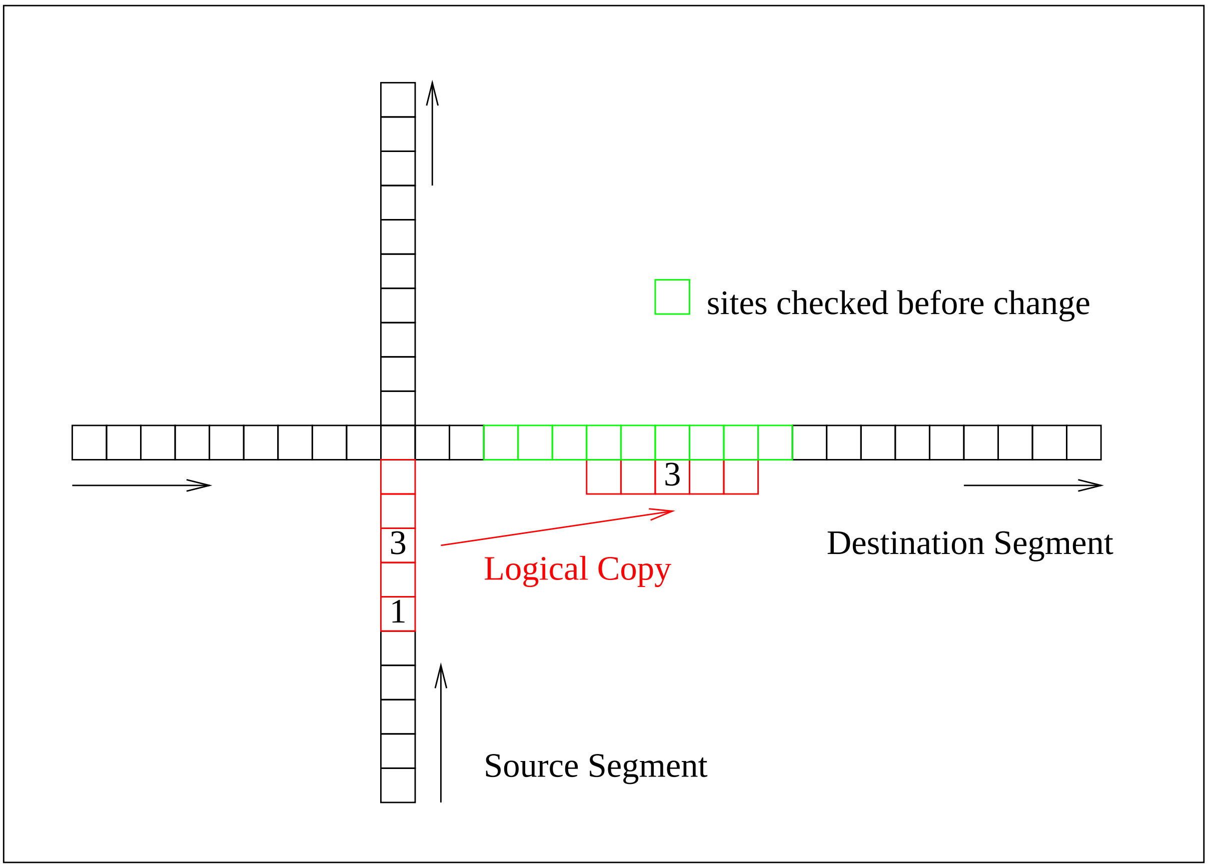
feature	SL	ML	MESO
individual vehicles	yes	yes	no
densities	no	no	yes
routes	yes	no	no
trans-prob.	no	yes	yes
HF-intersections	no	yes	-
vehicle types	no	yes	yes

Testbed: State NRW

- 550 nodes and 580 edges
- 19 boundaries
- 2 * 2000 single lane kilometers
- 523200 CA sites

Intersection Functionality

- as simple as possible
- model ramp metering
- unfortunately: vehicles block following traffic quite easily



Preprocessing and Setup

Problem: no realistic simulation data available. So use of reasonable artificial test cases.

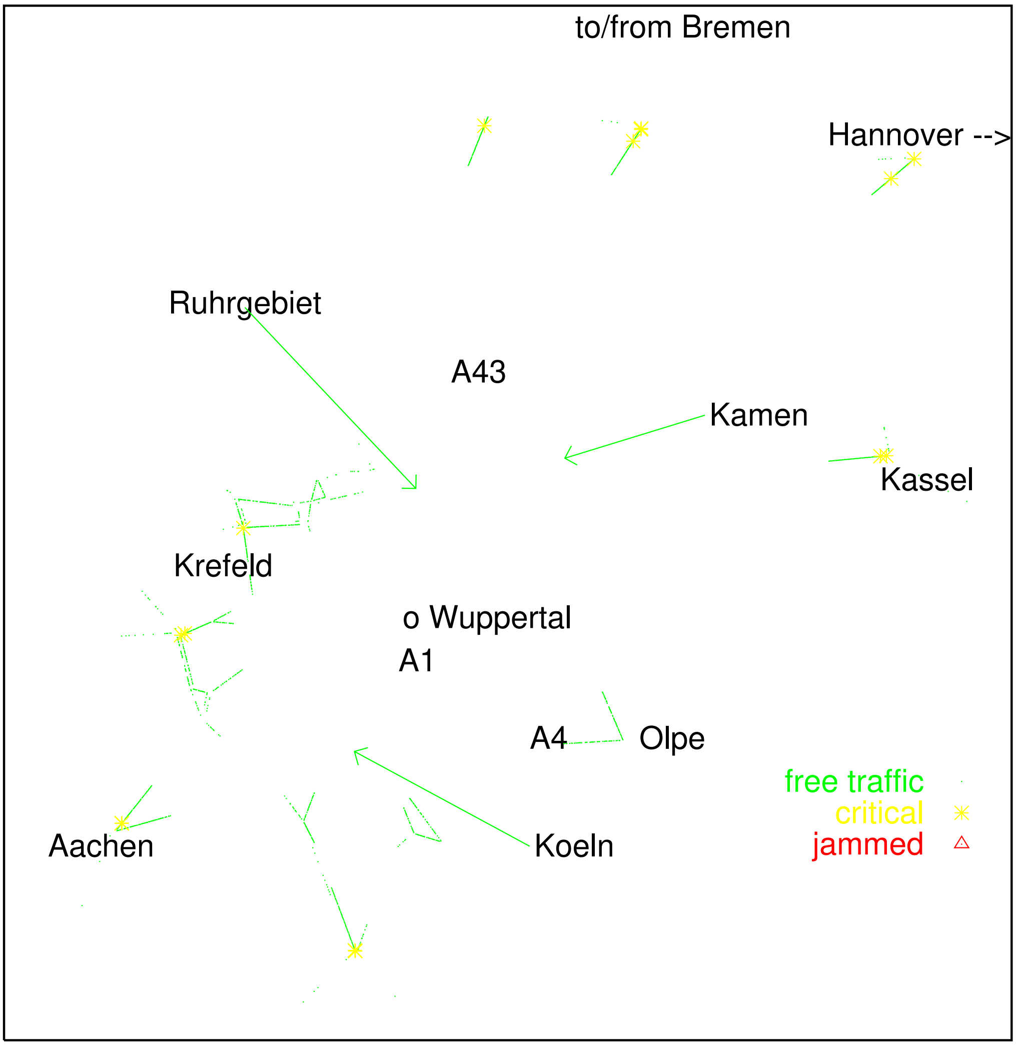
- Preprocessing: Compute the ten shortest routes for each origin destination pair.
- Boundary segments produce maximum flow into the system.
- Each vehicle has a route plan connecting two boundary segments.

Choice Of Routes

$P(dest) \propto distance$ short trips

$P(dest) \propto distance^4$ long trips

After 400 seconds:



Learning Process

1. 1st to 10th day: select one of the pre-computed routes, simulate, and store travel time.

2. Take the quickest route, simulate, and store new travel time t_{arr} according to:

$$
\begin{aligned}
t_{rem}(route) &= t_{arr} \\
t_{rem}(route) &= c t_{rem} + (1-c) t_{arr}
\end{aligned}
$$

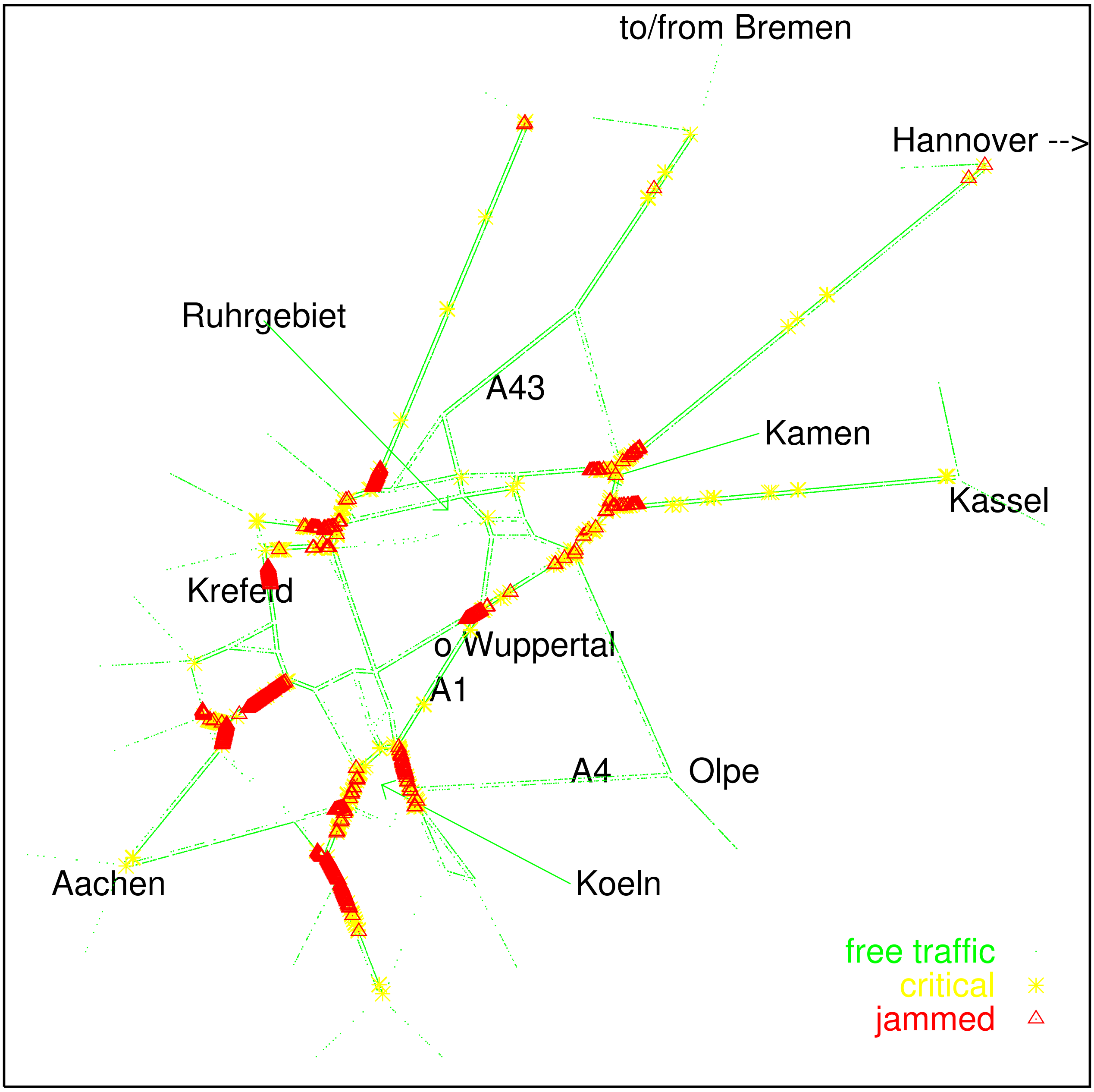
   $c =$ learning constant

3. Repeat 2

Traffic Simulation

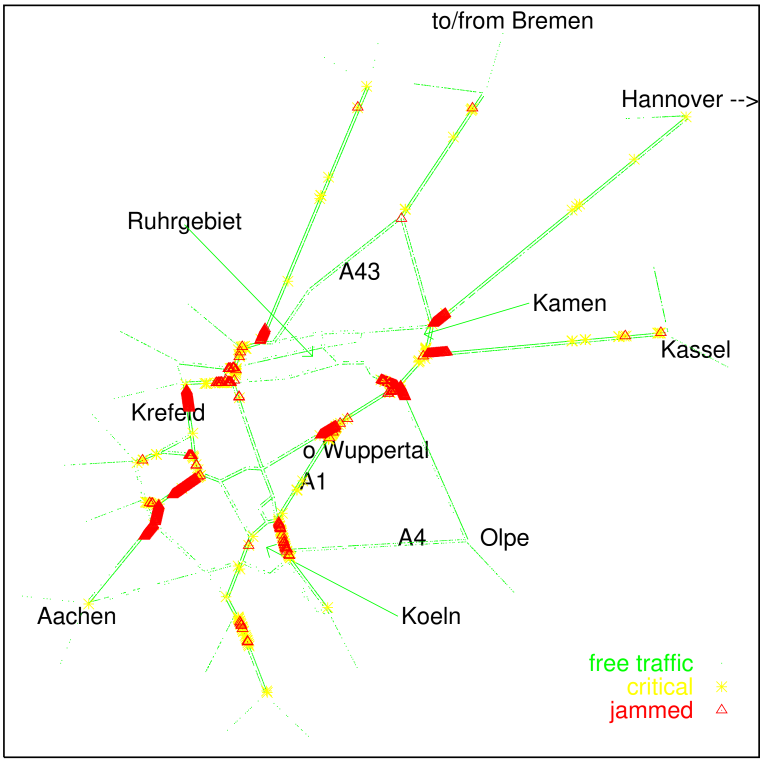
Day 15 after 100 minutes

4th order distribution

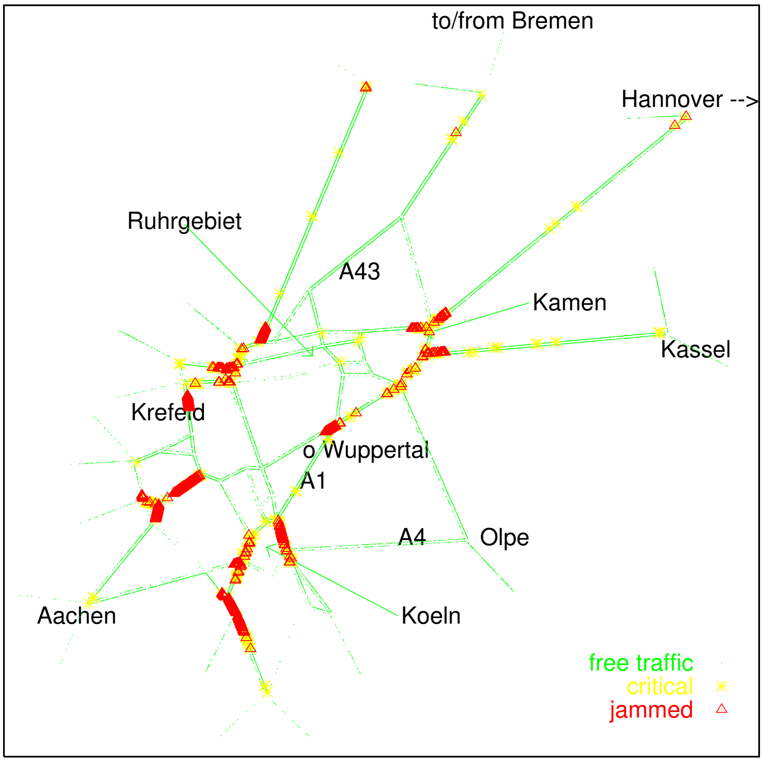


Traffic Simulation

Day 1



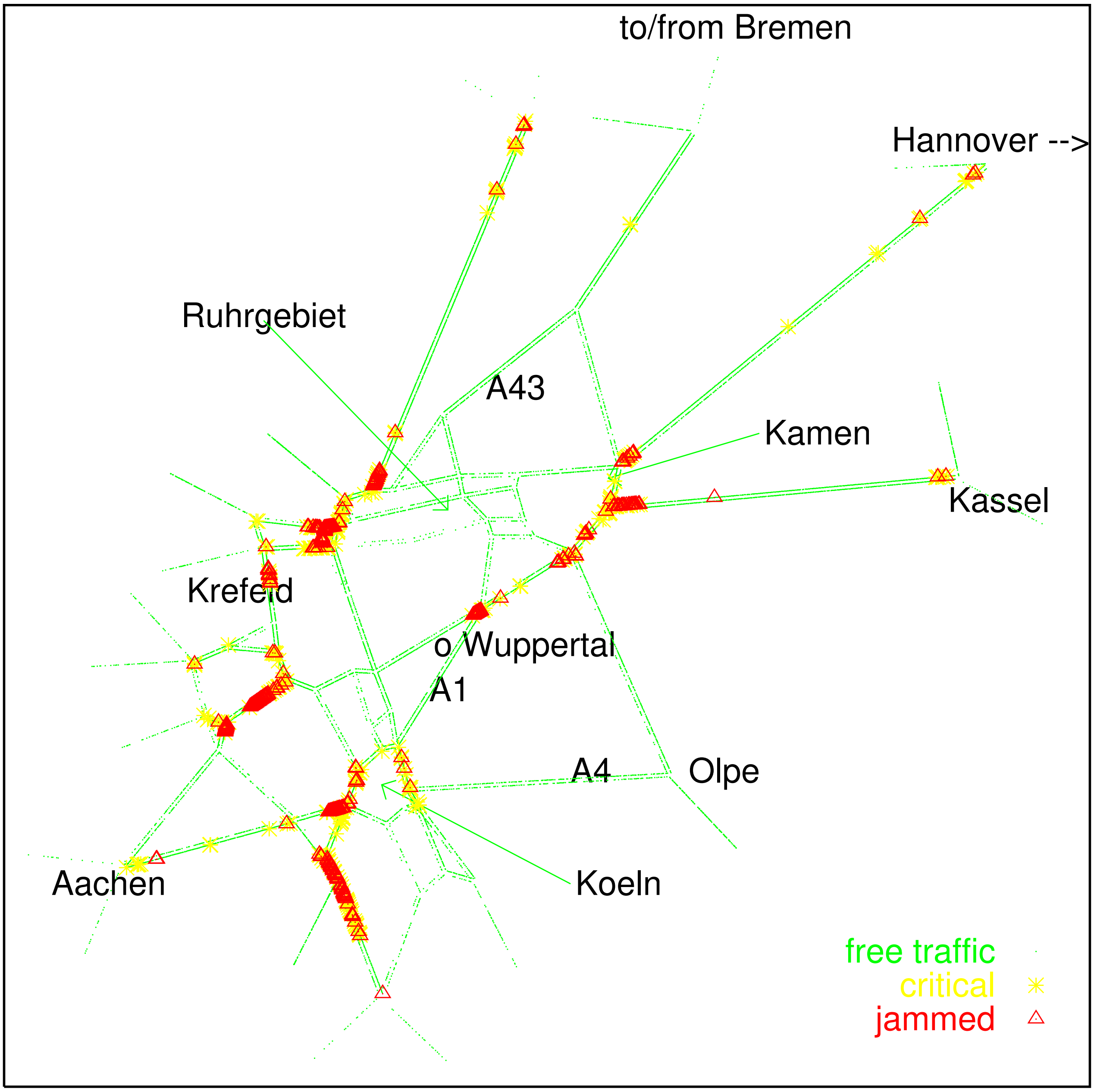
Day 15



Traffic Simulation

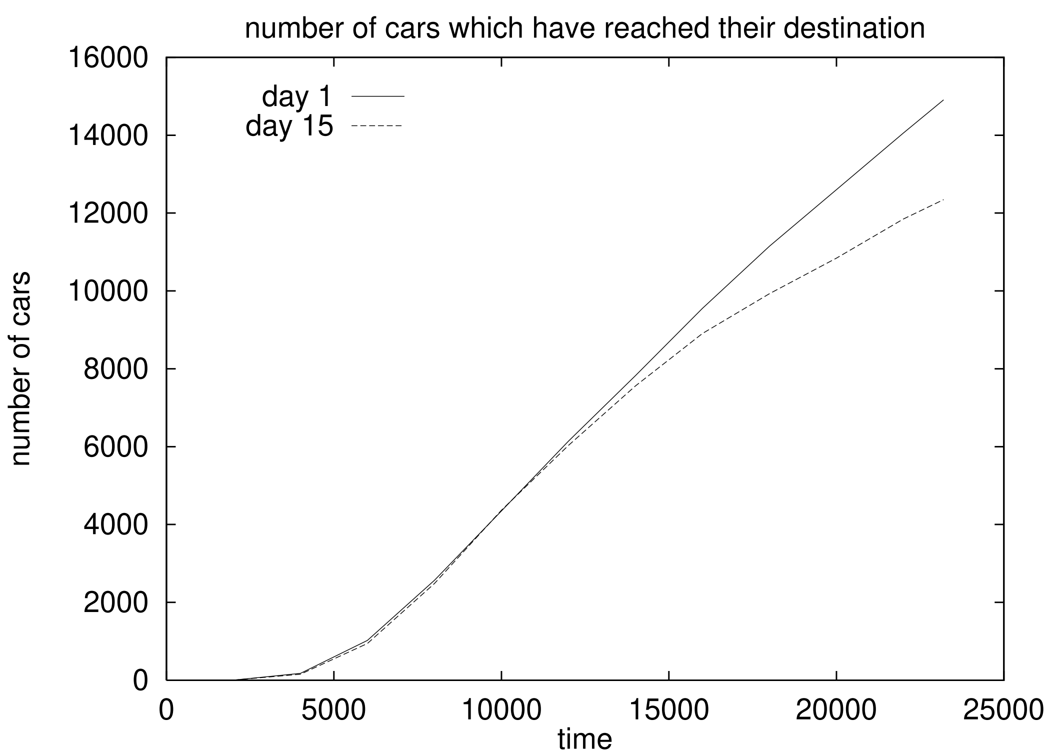
Day 15 after 100 minutes

linear distribution



Results (part I)

- Learning process results in a more homogeneous usage of edges
- System throughput decreases.
  Example of
  Nash-Equil. $\neq$ System Optimum



(preliminary result)

Results (part II)

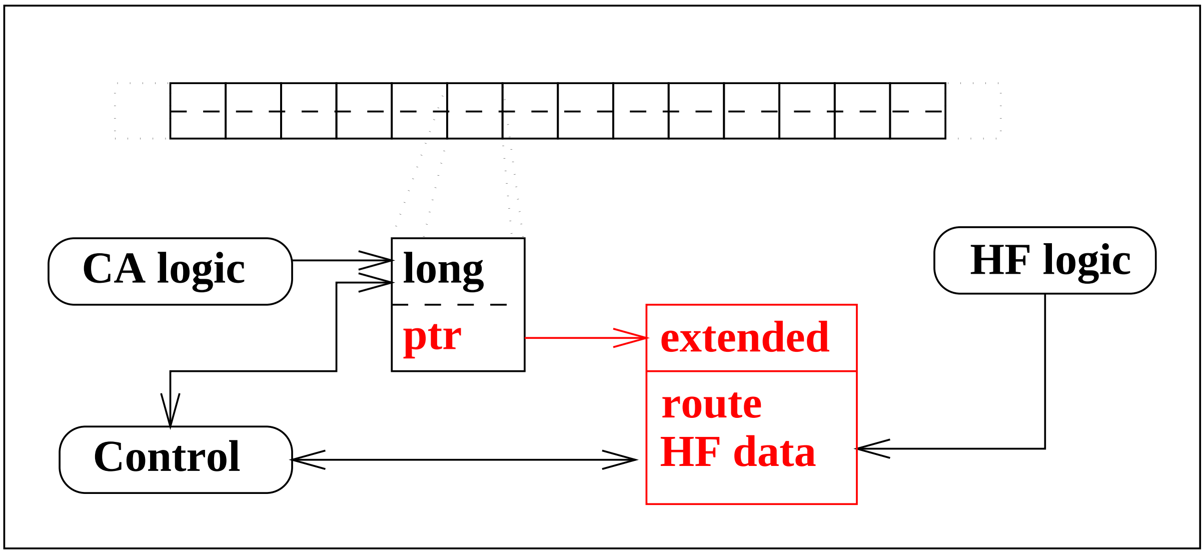
- Learning 'success' does not depend on learning constant within context and fidelity of this simulation. Remembering the latest travel time is sufficient.
- System was able to reproduce the bottlenecks known from real life traffic
- Bottlenecks are more or less independant of route distribution. Only bottlenecks in the interior are less severe for short trips.

Modifications of 'Classical' CA

- sites have individual characteristics (speed limit, passing prohibition)
- multi lane with lane changing rules
- different types of vehicles (acceleration, maximum velocity)
- refinement of CA parameters (acceleration and free–flow–velocity)
- variable vehicle lengths and possibly variable site lengths (e.g. 3.75 [m]).

CA: 'Passive' View

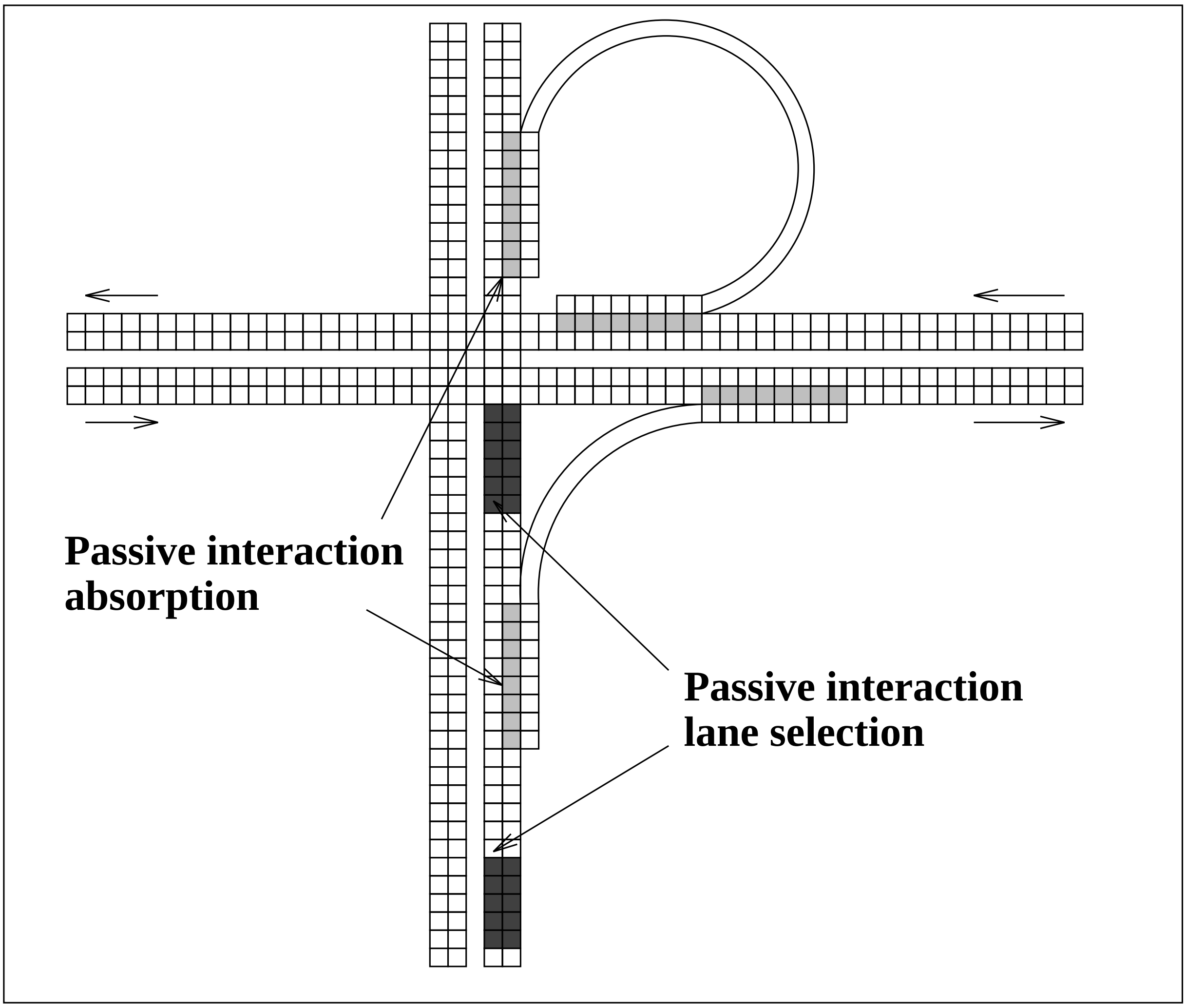
CA vehicle has additional pointer to extended data (e.g. route plan, data for High Fidelity model). External objects use the extended data to modify the base data whenever neccessary, but as infrequently as possible.



Extended data available for both models (CA, HF) ⟶ easy switching between models in hybrid simulation.

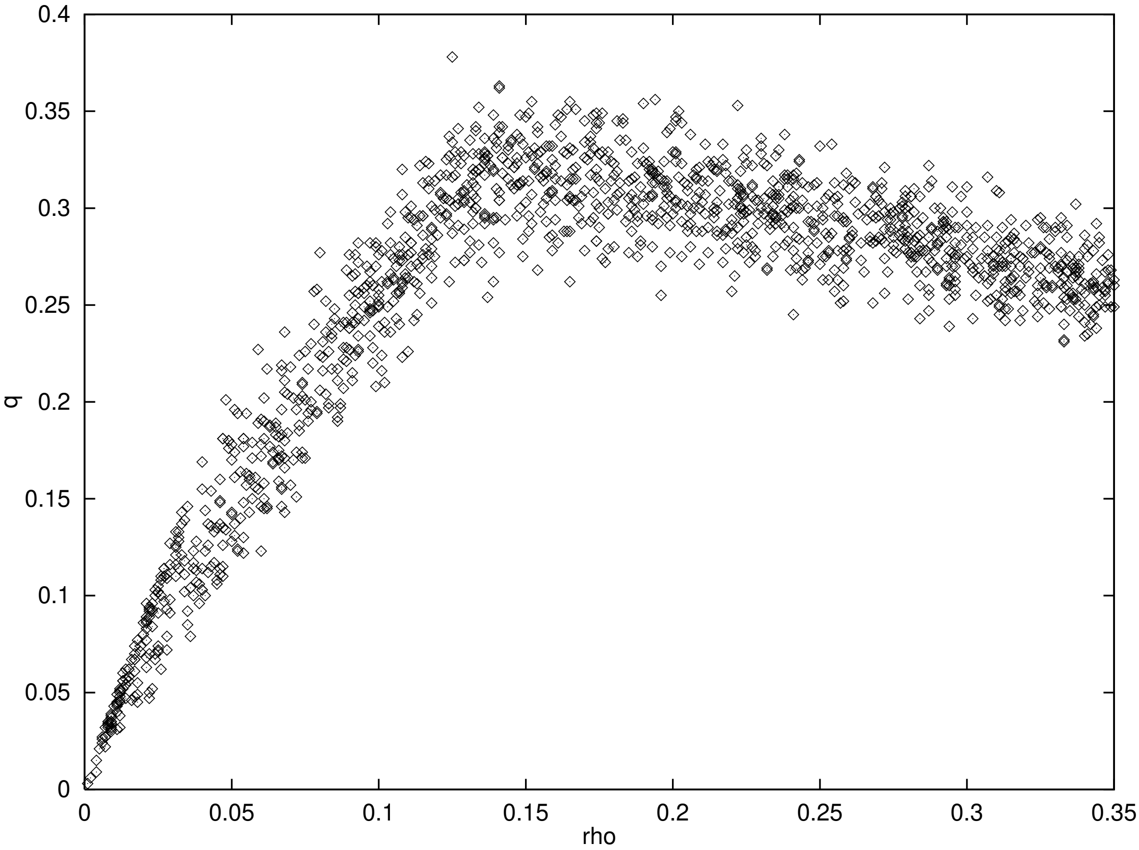
CA: Passive View

Example: intersection



Twolane CA

Example: Trucks and Passenger Cars



80 % fast ($v_{max} = 5$)
20 % slow ($v_{max} = 3$)

Performance

hardware	single	net
Sparc 10	1.9 MUPS	1.2 MUPS
5 * Sparc 10	8.9 MUPS	
Paragon		
- NRW,16		11 MUPS
- FRG,64		38 MUPS

MUPS = million updates per second

For Paragon with 64 nodes:

290,000 [km]

in realtime!

Outlook

- Develop multi lane CA logic for segments and intersections
- Combine CA and High Fidelity simulation models in a hybrid simulation
- Integrate the micro simulation into the context of traffic planning and traffic control
- Do online rerouting
- Use shared memory instead of message passing for future implementations.