Multiagent-based simulation of individual traffic in berlin

CUPUM 2005

Arnd Vogel, Kai Nagel

University of Technology, Berlin

Outline

- Introduction
- Multiagent-based simulations of traffic using MATSim
- Study region and available data
- Model setup
- Problems
- Conclusion

Introduction

- Multiagent-based simulation promise to become sophisticated tools for transportation planning
- Shift towards practical planning tool requires that existing data & knowledge can be used
- Project to evaluate implementation of a simulation model for Berlin based on existing data
- Preparatory work for a project funded by the VREF, investigating environmental effects of Urban Road Pricing Comparing Multi-agent based approach and classical" approach based on VISUM

Study focus

- Evaluate direct usage of available data
- Evaluate sharing of data between VISUM and MATSim
- Computational tractability
- Performance data

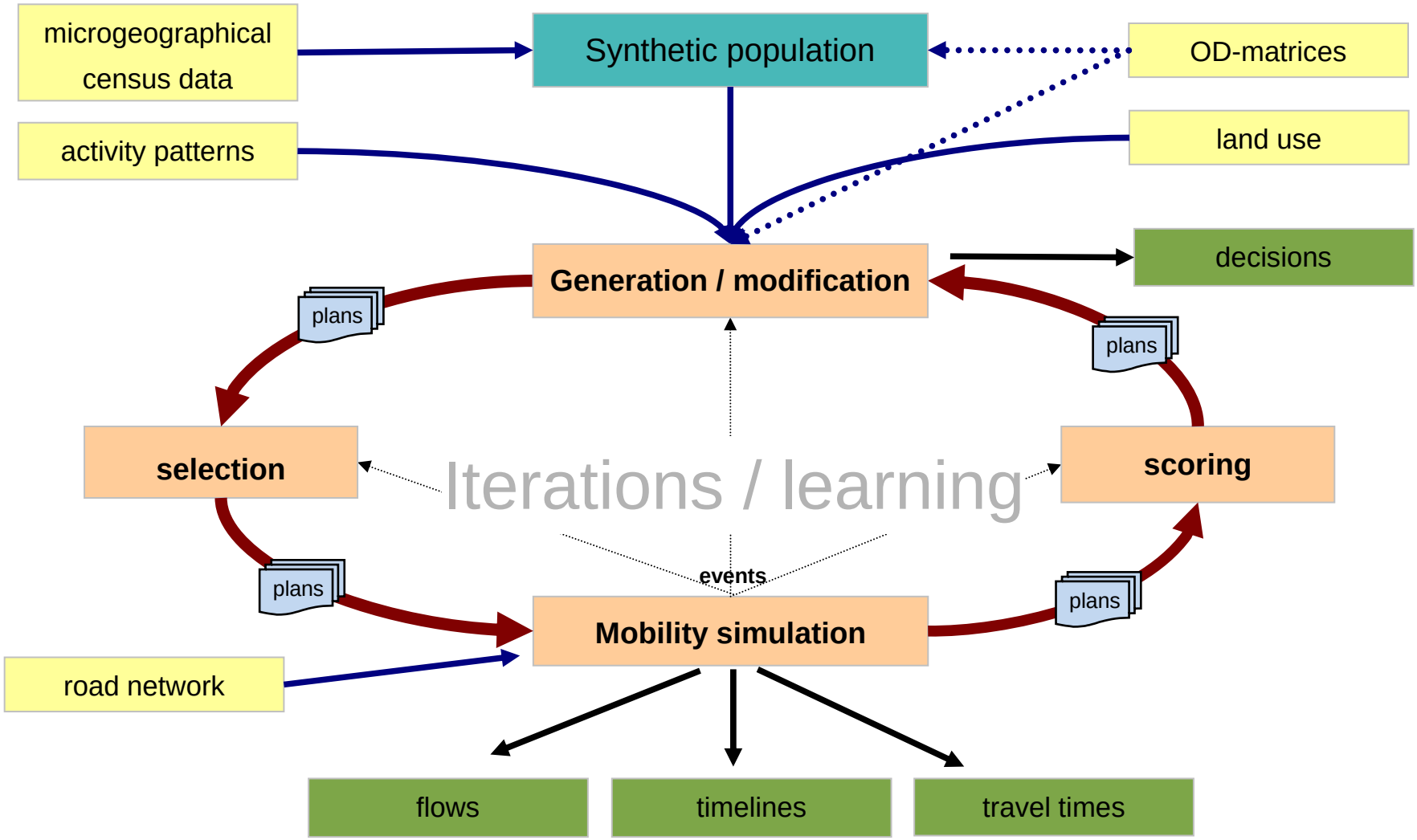
→ Report on initial model set up

→ No meaningful results so far, but problems and areas of future work identified

Multiagent-based simulations of traffic

- Steps of 4-step-process → person-based procedures
- Agents keep several versions of plans for one day
- Dynamic / “physical” traffic flow simulation
  - Simulate delays and congestions
  - MATSim → parallel Queue-model
- Scoring of plans
- Modification and selection of plans
- Iterative learning model

MATSim model structure



Multiagent-based simulations of traffic

- Plans (in XML) look like this

```
<person id="241" income="50000">
  <plan score="123">
    <act type="h" end_time="07:00" x100="7150"
         y100="2790" link="5834" />
    <leg mode="car" dept_time="07:00" trav_time="00:25">
      <route>1932 1933 1934 1947</route>
    </leg>
    <act type="w" dur="09:00" x100="0650"
         y100="3980" link="5844" />
    <leg mode="car" dept_time="16:25" trav_time="00:14">
      <route>1934 1933</route>
    </leg>
    <act type="h" x100="7150" y100="2790" link="5834" />
  </plan>
</person>
```

- Plans are „executed“ by the mobility simulation

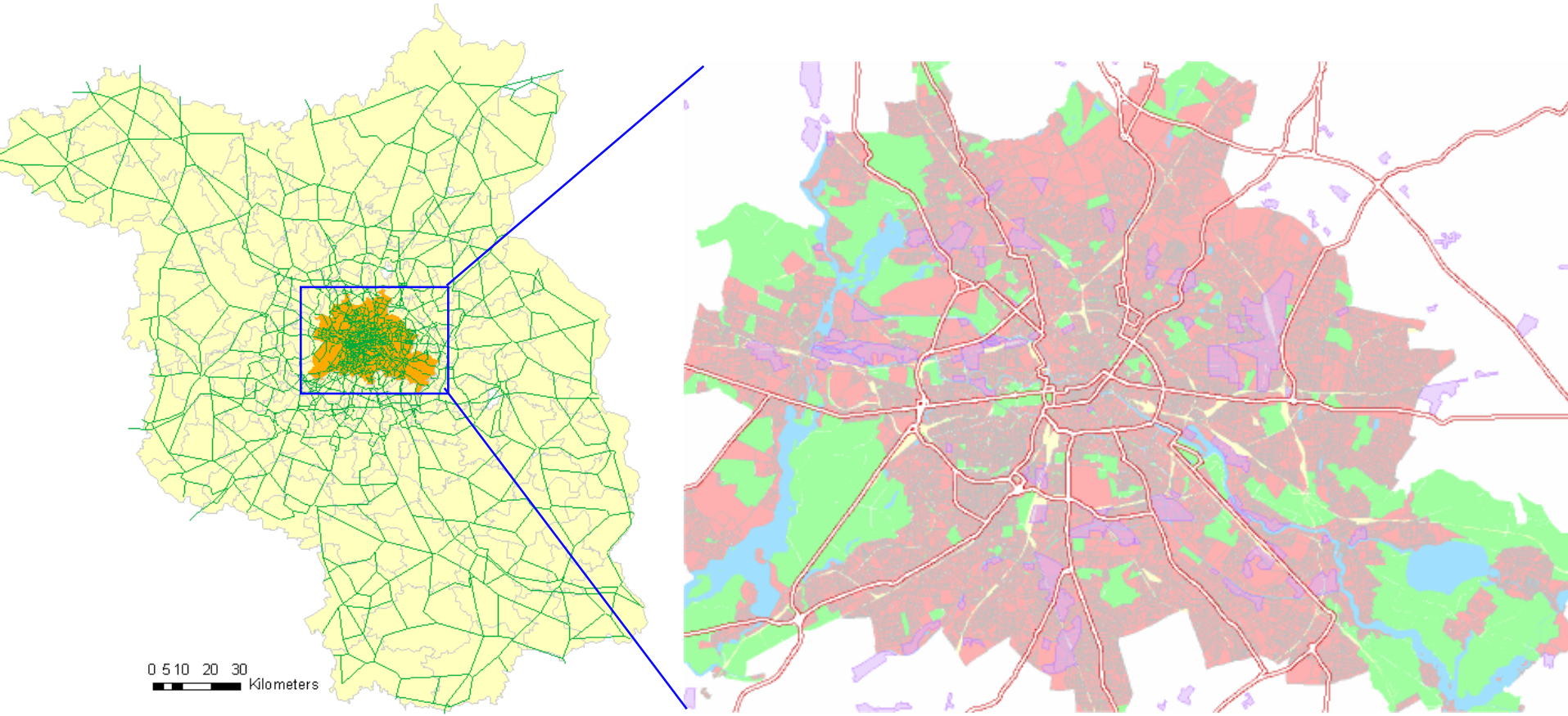
Multiagent-based simulations of traffic

- Scoring
  - Performance in Mobility Simulation
  - Penalties for being late at work etc.
- Simple modifications to plans
  - Routes → based on actual travel times
  - Departure times → e.g. depart 5 min later
  - Locations etc.
- Selection of plan for next iteration by Discrete Choice Model (→ some random)

The study region Berlin-Brandenburg

- Berlin and surrounding (Brandenburg)
  - 6 Mio inhabitants (Berlin 3.5 Mio, Brandenburg 2.5 Mio)
  - 150 km x 250 km
- Suburbanisation only began in late 1990ies
- Relatively low level of car ownership in Berlin,
  Modal Split of 40% car
- Congestion is not a real problem at the moment
  (but pollution/dust is a matter of concern)

The study region Berlin-Brandenburg



Transportation planning in Berlin

- Urban Planning Department
- Proprietary demand model
  (1020 traffic analysis zones, 885 Berlin, 135 Brandenburg)
- Based on trip frequencies and travel distances
  derived from household survey
- Assignment done with VISUM
- Only local traffic counts eg. before construction

Available data

- OD-matrices (different modes) only for 24h
- Basic census / land use data for 885 Berlin zones
- Road network for 1998 and for 2010 (planned)

Not (yet) available

- Time-dependent OD-matrices
- Data from household survey
  (household structure, employment, car availability)
- Network-wide traffic counts

Demand generation

- Goal is to generate initial plan for each person

```
<person id="241" income="50000">
  <plan score="123">
    <act type="h" end_time="07:00" x100="7150"
         y100="2790" link="5834" />
    <leg mode="car" dept_time="07:00" trav_time="00:25">
      <route>1932 1933 1934 1947</route>
    </leg>
    <act type="w" dur="09:00" x100="0650"
         y100="3980" link="5844" />
    <leg mode="car" dept_time="16:25" trav_time="00:14">
      <route>1934 1933</route>
    </leg>
    <act type="h" x100="7150" y100="2790" link="5834" />
  </plan>
</person>
```

- Different (increasingly complex) approaches

1. Directly convert the OD-matrix into trips
2. Create a synthetic population from census data
   and do primary activity location choice using OD-matrix
3. Create a synthetic population with “true” location choice

Model set up (so far)

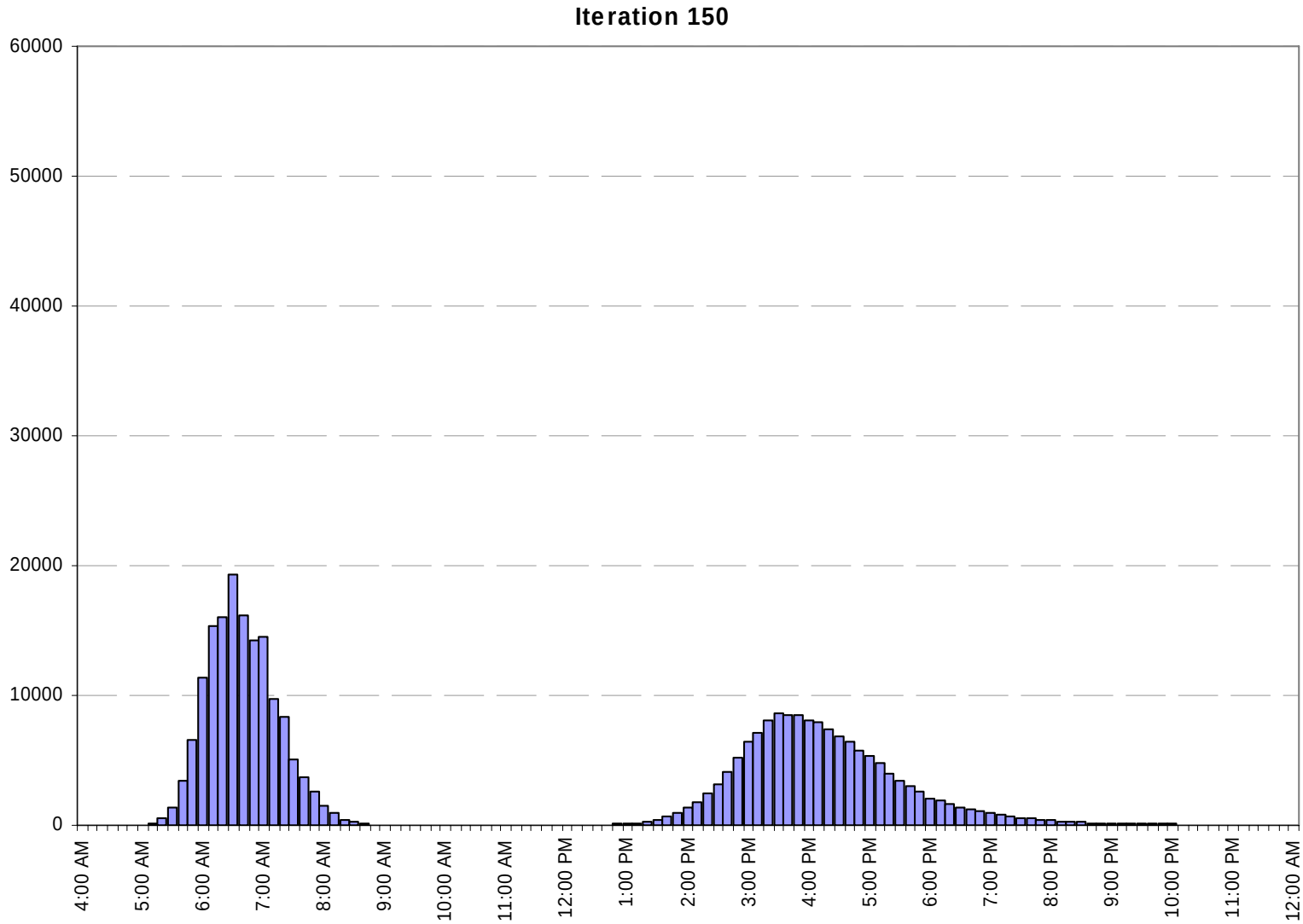
- Using simplest approach for demand generation (OD-matrix)
- Assuming all trips to be *home-work-home* and to start (initially) e.g. at 7am
- 8hrs work, must start between 7am and 9am
- Modify routes and departure times in iterations,
- 5% sample for faster testing (~150.000 agents) [with reduced network capacities]

Implications of the model set up

→ Using the OD-matrix lets return trips become "additional" agents

→No relationship to real traffic, but ok (and easily feasible) for initial testing of the network

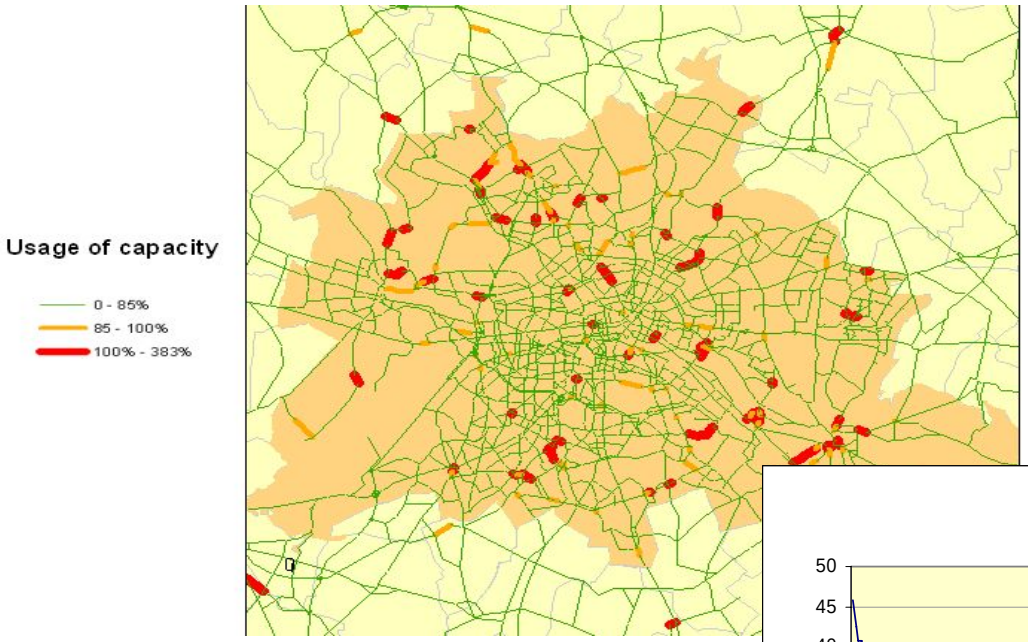
Variation of departure times

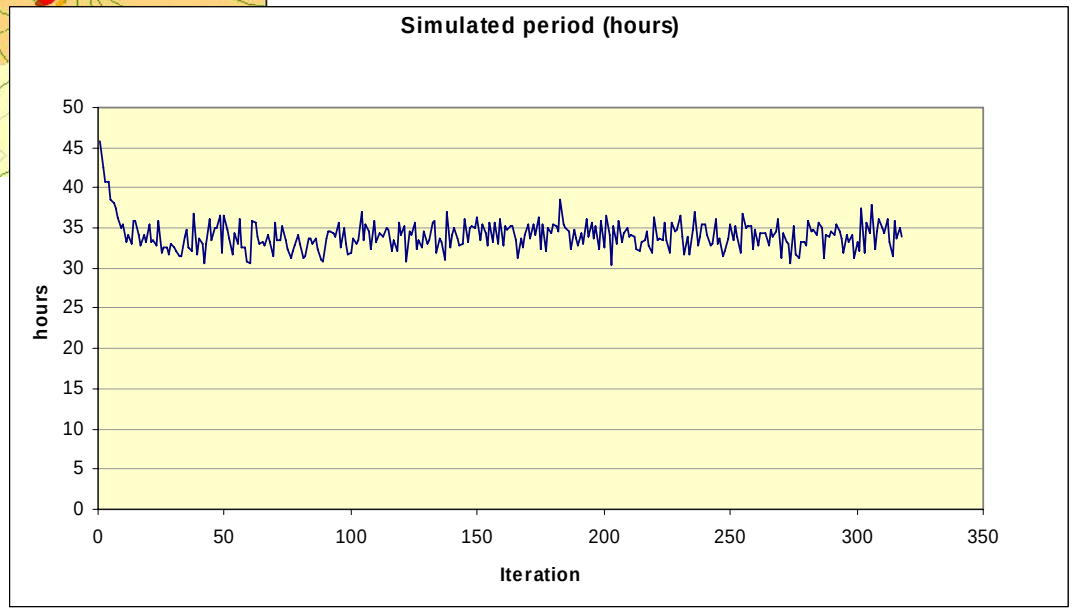


Problems during implementation

- Network errors and subtle codification
  - Links of length 0 coding one ways
  - very low capacities
- Just result in high usage of capacity in VISUM
- Especially in conjunction with “artificial” demand:
  Real problem for Queue model of MATSim Mobility Sim.
  - Short links are completely “blocked”
    (but are favored in initial routes…)
  - Persons travel longer than 24h (which breaks scoring function…)
- “reduced capacity” trick may amplify problem / not work

Effects of Network errors





Tracktability and Performance

- On a 2.2 GHZ, 2 GB machine,
  one iteration takes approx. 25 minutes
- Simulating 150.000 agents uses approx. 1 GB
- Mobility can handle larger scenarios without significant performance mobility in parallel mode (already implemented)
- In order to handle full Berlin scenario (> 3 Mio. agents), agent database must run in parallel mode
  (implementation in progress)

Conclusion & future work

- Applying MATSim to an unknown, real world database can raise serious problems…
- Performance on modern Single CPU PC is already good for running medium sized scenarios

Outlook on future work:

- Better approaches for demand generation ☺
- Work on software robustness
- Full parallel implementation needed for large scenarios

Thank You for Your attention…