PRELIMINARY RESULTS OF A MULTI-AGENT TRAFFIC SIMULATION FOR BERLIN

ULRIKE BEUCK, KAI NAGEL, MARCEL RIESER, DAVID STRIPPGEN

VSP, TU Berlin, Salzufer 17-19
Berlin, 10587, Germany
{beuck/nagel/rieser/strippgen}@vsp.tu-berlin.de

MICHAEL BALMER

IVT, ETH Zurich, ETH Hönggerberg
Zurich, CH-8093, Switzerland
balmer@ivt.baug.ethz.ch

This paper provides an introduction to multi-agent traffic simulation. It includes some description of where we are with respect to the implementation of a real-world Berlin scenario. It turns out that the difficulties there, because of data availability and because of the special situation of Berlin after the re-unification, are considerably larger than in previous scenarios that we have treated.

Keywords: traffic simulation; multi-agent simulation; large-scale real-world scenario.

1. Introduction

In recent years, microscopic traffic simulations have become an increasingly active field of research in transport science. "Micro" refers to the fact that all elements of the transport system, like roads, crossings, vehicles, and – most importantly – travelers (referred to as "agents") are resolved. This modeling approach is in contrast with the more aggregate models implemented in current transport planning software and used by transportation planners. While those programs have seen several decades of development and practical use, agent-based microscopic simulation systems are still relatively new and are mostly used in small and medium scale scientific scenarios rather than in real world applications. But new technologies, such as robust and fast object-oriented programming languages and high performance computing clusters make the applications increasingly realistic and increasingly large scale. This paper addresses the issue of applying such a model to a real world scenario of large dimensions.

2. The model

2.1. *Physical vs. mental level*

There are several ways from which a microscopic approach can be derived. One way is the attempt, often used in physics, to start from "first principles". First principles implies to start from individual particles, and indeed the possibility to do fast molecular dynamics [1] and fast cellular automata simulations [2] on the microscopic level was one of the driving forces for large scale microscopic traffic simulations [3].

When going down this path, one notices eventually two things:

- Building a microscopic simulation of vehicular traffic (or, for that matter, of pedestrians) needs some diligence and care, but is essentially possible.
  In terms of size: Even large urban systems rarely have more than 10^7 inhabitants, and rarely more than about 20% of those are simultaneously on the road. This makes for considerably smaller numbers of particles than in many physics applications, and moves the microscopic simulation of complete cities/regions into the realm of the computationally feasible.
  In terms of underlying dynamics: Despite considerable discussion about different traffic states and possible phase transitions [4; 5; 6], the absolutely most important elements of the dynamics are, in fact, rather constrained: There is mass conservation (especially if vehicles are tracked from parking to parking; see below); vehicles move on a quasi one-dimensional geometry (roads) most of the time; there are quite severe restrictions on acceleration and braking capabilities; and there are quite severe excluded volume restrictions (not more than about 150 vehicles fit on a kilometer of single-lane roadway when traffic is stopped). A consequence of this is that a rather simple theory of traffic – that of kinematic waves [7] – describes traffic rather well, by just using the equation of continuity plus an equation of state (relating velocity to density, the so-called "fundamental diagram") [8]. Any microscopic model that obeys the corresponding microscopic principles – mass conservation and velocity related to the distance to the car ahead – will reach a similar level of realism [9]. More complicated aspects, such as capacity drop [10], phase transitions (see above), or "synchronized traffic" [11], matter for the management of individual road segments, but they do not matter so much for where we currently are with simulations of large scale urban systems.
- The other thing that one notices is that the behavior of the "particles" (vehicles, pedestrians) is quite heavily influenced by behavioral aspects, i.e. by "what goes on in people's heads". This is, however, not so much the realm of physics, since one is not interested in, say, how 10^6 neurons together eventually make a decision, but instead in models that generate realistic human decisions within very short computing time. For typical computing

situations, a lane changing decision may not take more than $10\mu sec$ of CPU time, and a decision about a daily plan may not take more than a second of CPU time. That is, one is interested in models that describe the outcome of human decision-making reasonably well, without "looking at the neurons".
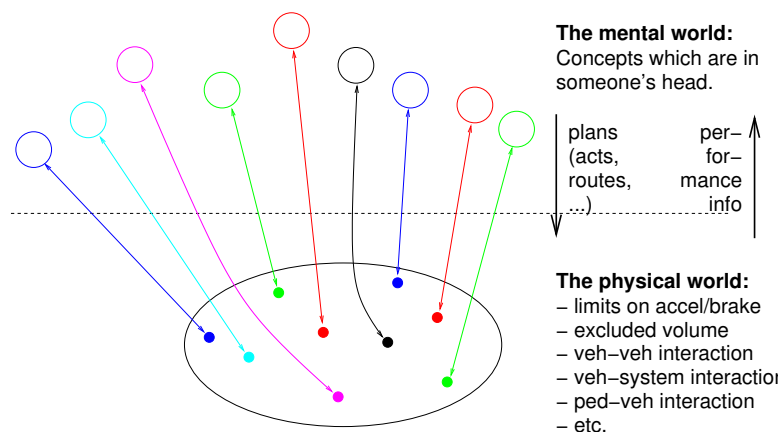

Fig. 1. Physical vs. mental level

Accordingly, it is useful to differentiate between the **physical layer** and the **mental layer** of a multi-agent simulation (MASim), see Fig.1. The physical layer essentially contains everything that can be observed. The mental layer essentially contains everything that goes on in people's heads.

2.2. *The physical level*

As just said, there is a variety of techniques available for the simulation of the physical level. These techniques include molecular dynamics [12], techniques based on differential equations [e.g. 13], coupled maps [14; 15; 16], cellular automata [17], methods where individual vehicles are moved with velocities based on link densities [18; 19], methods where individual vehicles are moved based on fluid-dynamical equations [20], and queue models [21]. Some packages based on these different techniques are SUMO [using coupled maps and more recently a queue model; 22], DYNASMART, DYNAMIT, METROPOLIS [all using a combination of velocities based on link densities and a queuing approach; 23; 24; 25], TRANSIMS, OLSIM [both using a cellular automata approach; 26; 27], or our own package MATSIM [using a queuing approach and more recently also vehicle movement based on fluid-dynamics; 28]. Since these are not the main focus of this paper, it shall suffice to have given these references.

4   *Beuck et al.*

2.3. *The mental layer*

As mentioned above, the mental layer models and simulates the human decision processes. Those include:

- accelerating, braking, lane changing
- turning decisions at intersections; route choice
- time choices (when to depart?)
- mode choices (which mode of transport to use?)
- location choices (where to do an activity?)
- activity pattern choice (which activities should be done at a given day, and in what sequence?)

This is an approximately hierarchical list, in the sense that decisions further down the list are made less often, and in consequence decisions further up the list depend on those further down the list. For example, in order to compute a route from home to work, one first needs to know where home and work are located.

Although the above hierarchy can be justified by empirical observation [e.g. 29], there is also considerable inverse causality between the levels. For example, location choice (for example for a shopping location) depends on the available modes of transport.

With respect to model implementation, the following seems to establish itself in the community:

- Driving behavior, such as accelerating, braking, or lane changing, is included into the physical layer. That is, it is not assumed to be part of any strategy, but rather assumed to be purely reactive.
- Routes are typically generated using some kind of shortest (or fastest) path algorithm. This is most probably due to the fact that a shortest/fastest path in a traffic graph is relatively cheap to compute by using the Dijkstra algorithm [30]; it is, in fact, difficult to devise heuristics that are faster than that exact algorithm. If several route alternatives are available, selection between them is often done using a so-called multinomial logit or probit model [31]. Some care needs to be taken to correctly deal with correlations between alternatives [32]: Assume three route alternatives, where two of them differ in just one link, and the third is very different. The intuitive split between those would be roughly $25\% : 25\% : 50\%$, while plain multinomial logit returns $33.3\% : 33.3\% : 33.3\%$. This is known as "independence from irrelevant alternatives (IIA-property)" [see, e.g., 31].
- The choices of times, modes, locations, and activity patterns are often done in one model, called activity-based demand generation (ABDG). There are two major strains of models: those based on econometrics/utility maximization, and those based on rules. Most real-world implementations are a combination of those two approaches [e.g. 33; 34; 35; 36; 37; 38].

The above assumes that the population and where it lives is given. That is, the (synthetic) population can be seen as a fixed boundary condition of such simulations. Synthetic populations are generated from demographic data [e.g. 39]. Some models, normally separate from traffic models, also consider the evolution of a population over time (including aging, birth, and death), and include residential choice into those models [e.g. 40; 41; 42].

In our work, we call the output of the mental layer **plans**, which can look like:

```
<person id=="123" gender="male" income="50k">
    <plan score="456">
        <act type="home" link="234" endTime="08:05" />
        <leg mode="car" expectedTravelTime="00:55" />
            <route> 25 35 46 63 </route>
        </leg>
        <act type="work" link="345" duration="09:00" />
        ...
```

That is, a plan is a full description of the agents' intentions. The above agent intends to leave home at 8:05, take a pre-specified route to work with an expected travel time of 55 min, work for 9 hours, etc. Although in much of our work, plans are fully specified, conceptually they do not need to be so: It is quite reasonable to assume that some elements (e.g. the time to leave work) is decided depending on circumstances (e.g. how much work there is), and other elements are modified on short notice (e.g. the route, in order to circumvent some exceptional congestion).

2.4. *Adaptation and learning; day-to-day vs. within-day replanning*

If one runs the sequential process of synthetic population generation, activity-based demand generation, and routing, the resulting plans are often not useful since they will not execute as expected. A typical obstacle is congestion, which is a consequence of too many plans attempting to use a certain element of the infrastructure at the same time. Congestion will make certain choices sub-optimal, in the sense that an agent could find a better solution by modifying its plan.

There are two principal ways to model replanning:

- Option 1, called **day-to-day replanning**. The physical level simulates a day, then the plans of the agents are adapted, the physical level simulations a new day based on the new plans, etc. In pseudo-code:

  ```
  for ( day = 1 ; day <= lastDay ; day++ ) {
      for ( time = 0 ; time<endOfDay; time++ ) {
          advance_physical_layer_by_one_time_step() ;
      }
      for ( agent in agents ) {
          agent.replan() ;
      }
  }
  ```

  Most of the syntax is hopefully clear; `for ( agent in agents )` means that the loop goes through all agents; `agent.replan()` means that the specific agent is now asked to potentially replan.

In this version, agents pre-plan their complete day before they leave home in the morning, and they can only re-consider their plan just before they start the next morning. Agents can therefore not react to unforeseen circumstances.

- Option 2, called **within-day replanning**. The physical level simulates a short time period, then all agents can re-plan, the physical level simulates another short time period, etc. In pseudo-code:

```
for ( day = 1 ; day <= lastDay ; day++ ) {
    for ( time = 0 ; time<endOfDay; time++ ) {
        advance_physical_layer_by_a_one_time_step() ;
        for ( agent in agents ) {
            agent.replan() ;
        }
    }
}
```

More intelligent/efficient implementations of this can be considered, such as agent replanning being triggered by certain conditions during the update of the physical layer.

In this version, agents can replan while they are on the way, thus being able to react to unforeseen circumstances. Important examples of unforeseen circumstances are fluctuations from one day to the next, or exceptional events, such as accidents.

The two options face different levels of implementation difficulties:

- Option 2 is easier to implement by a single programmer or by a tightly integrated programming team where all members of the team have agreed to use the same data structures (e.g. for the agents).
- Option 1 is easier to implement if there is pre-existing, non-integrated code, or if the programming team is not tightly integrated.

In consequence, within-day replanning is often implemented by single-person projects [43; 25; 22; 44] or by projects that can define and enforce their programming standards, while day-to-day replanning is often the result of a multi-person or multi-team project [24; 26; 42].

A direct consequence is that projects with within-day replanning are often somewhat limited in scope, since it is difficult to combine pre-existing work. Since, on the other hand, within-day replanning is an important aspect of reality, it seems critical to overcome that obstacle.

2.5. *Scores and "events"*

For most applications, replanning only makes sense if the agent attempts to obtain a "better" plan by replanning. This implies that one needs to be able to compare plans. We assume that the plan is scored by submitting it to the physical layer, and scoring the outcome. That is, the plan is seen as the description of the strategy

of the agent. The strategy is then interpreted and executed by the physical layer. If the strategy contains infeasible elements (e.g. a route that is not connected), it will fail completely. Even if the strategy is feasible, it may not be very good, since, for example, assumptions about travel times may be optimistic. The physical layer provides output in terms of **events**, which structurally look as follows:

```
<event time="08:05" agentID="123" eventType="leavingAct" act="home" />
<event time="10:00" agentID="123" eventType="arrivingAtAct" act="work" />
<event time="19:00" agentID="123" eventType="leavingAct" act="work" />
```

This refers to the agent as described earlier by the example plan. The agent leaves home as intended, but needs one more hour than intended to get to work. Consequentially, she will leave work one hour later than intended, since the duration of work is fixed by the plan. The scoring will be based on the longer duration of travel, and the later work start/work end times. If, as a result, she gets late to an appointment later, that will cause further score reduction. If the morning travel delay occurs regularly, she will learn, say, to depart earlier.

Scoring functions can be arbitrary, although in practice, it is currently easiest to remain close to the utility functions used in economics. There is some research into how to assign utilities to full daily plans [45].

2.6. *Co-evolution, dynamical systems, and evolutionary game theory*

Day-to-day learning can be described in terms of an evolutionary game. If, over the iterations, all agents end up with plans that they cannot unilaterally improve, then the system is at a **Nash equilibrium**. The plans can be seen as "strategies"; the execution of the plans in the physical level can be seen as "scoring the strategies" or "computing the fitness function". The concept even holds when the simulation of the physical level is stochastic; then "score" needs to be replaced by "expected score".

The day-to-day evolution of the system can be seen as a time-discrete dynamical system where many agents co-evolve. For such systems, some theory is available [46], although there is a gap between theoretically tractable systems in the traffic context and full-blown multi-agent simulations. For such theoretically tractable systems, one can show: If, in every iteration (day), a small fraction of the agents switches to what would have been the best expected plan in previous iterations ("**best reply**"), and if that system moves to a fixed point, than that fixed point is a Nash Equilibrium. There are, however, at least two caveats: (1) The implementation of "best expected score" is not easy to make operational because a simulation is computationally rather expensive, and averaging over several such simulations is even worse. (2) As is well known, dynamical systems do not need to converge to fixed points. They can instead converge to cycles, or to chaotic attractors [47].

3. Set-up of the Berlin simulations

Although the scenario focuses on the city of Berlin, in order to simulate average traffic conditions we have to model and simulate Berlin's surrounding as well but with a lower level of detail. All together the study region covers an area of 150 km x 250 km and has a population of about 6 million inhabitants. Network and demand are derived from data used and produced by the aggregated macroscopic model that Berlin's planning department is working with. In contrast to this official transport model used for mid and long term forecasts, in our simulations all travelers are resolved as agents generating trips while following their day plans. The following sub-sections describe the set-up of the Berlin scenario.

3.1. *Boundary conditions: The network*

As mentioned earlier, Berlin's planning department provided us with a road network of their transport model. This network has been used as part of the forecast model for the year 2015. Since we aim to model and simulate Berlin's current traffic of an average workday, we had to adapt the network manually in order to exclude modifications planned to be realized until 2015 (e.g. expansion of the inner city highway southward). The final network consists of almost 30,000 links connecting more than 10,000 nodes, described by their coordinates. For our simulation we need, for each link, the attributes free flow speed, length, number of lanes and flow capacity. The network does indeed contain these attributes, but the usefulness of the data is variable. For example, the number of lanes in uniformly set to one, presumably because the number of lanes does not matter for traditional assignment models. Link capacity is interpreted very differently by the aggregated model used by the planning department of Berlin and our multi-agent simulation. While in our simulation, capacity is understood as maximum outflow of a link in a given time period, the aggregated model does not treat a link's capacity as hard constraint. In traffic assignment suitable functions are used to relate capacity and flow with the resulting cost in terms of travel times. Thus, we had to adapt these capacity values that were the basis for a 24 hours static assignment. In a first step, we adjusted the 24 hours capacity values in order to derive 1-hour values based on the assumption that daily traffic basically occurs in a 12-hours period. In a second step, we converted the resulting theoretical 1-hour values into maximum values of outflow of a link in 1 hour to be used in our multi-agent simulation.

Free flow travel time is calculated as link length divided by the free flow speed of the link. Additionally, the storage of a link is constrained. The storage of a link is calculated as length times the number of lanes divided by the space a vehicle occupies in a jam (7.5 m). Because of the incorrect number of lanes (uniformly one, which is much too small for the wide roads of Berlin), the space capacity needed correction as well. For the time being, we assume a storage of 3-lane roads everywhere – note that this affects only the storage (maximum number of vehicles on link), not the flow capacity.

In order to speed up the Berlin scenario, the demand and the network capacities (both flow and storage) were scaled down to 10% of the actual values.

3.2. *Initial conditions: Initial plans from an activity-based demand generation program*

Initial plans have their source in an activity-based demand generation (ABDG) model [also known as *Berliner Personenverkehrs-Modell*; 48; 49; 50]. It has been used to calculate three daily (= 24-hour) OD-matrices used as input data for the static assignment used by Berlin's planning department, differentiated between personal travel, freight travel, and through traffic. However, the model is in fact a disaggregated activity- and behavior-oriented traffic demand generation model. The demand of 72 person groups with similar demographic attributes and homogeneous behaviors is calculated based on expectancies. The model was modified to output activity chains to be used to produce initial agents' plans for our multi-agent simulation [51].

Activity chains are grouped by the 72 person groups. Each activity chain contains information about the start location, up to four activities, and the frequency of occurrence of the activity chain. Activities are described by their type, location, and the transportation mode used to reach that location. The home location is start and end location of each activity chain (round trips). Information on location refers to traffic analysis zones (TAZ), since these represent sources and sinks of traffic streams in the macroscopic model. Before transforming activity chains into agents' plans, location information and data has to be disaggregated. Additionally, activity chains lack time information. For initial plans, all activities are assigned a random activity duration within a type-dependent range.

As a result, over 7 million "virtual" agents are generated from the round-trips in the ABDG data. Each of these agents has a plan corresponding to an activity chain generated by the Kutter model. Unfortunately, the number of these agents does not correspond to the number of real persons in the area, since persons who make more than one round trip per day are registered as separate "virtual" agents. This is due to the fact that the Kutter model treats round trips, not persons. That is, activity chains with intermediate home stops are treated as completely separate round trips, resulting in separate agents.

We then decreased the number of agents in our simulation to the agents using the car for transportation. As already mentioned, to speed up the Berlin scenario we also scaled network capacities as well as demand down to 10%, which gives a total of about 205,000 car travelers with complete day plans in our simulation.

3.3. *Mental layer: "planomat" and router*

As already described, an agent's plan is a description of its intentions, but it might not be executed as expected because of congestion effects. At the end of an iteration, a score is calculated for each plan, corresponding to how successful an agent was

performing its plan (see Sec.3.5). A certain percentage of agents can adapt their plans, before the next simulation of the physical layer starts.

Two strategy modules enable this day-to-day replanning. The first module is the router. Given locations, departure times and activity durations, an agent tries to find a better route in terms of minimum travel costs based on the previous iteration. The router is based on Dijkstra's shortest-path algorithm, and shortness is measured by travel costs in terms of travel times on the links of a route. Travel times depend on how congested the links are, and so they change throughout the day. The second strategy module is the so-called planomat [52]. Using this module, departure times or activity durations can be altered in order to optimize the score of the plan. Also, altering the activity sequencing and activity dropping are possible modifications but are not implemented yet.

3.4. *Physical layer: Queue simulation*

The physical layer is simulated using a queuing approach [21]. The agents' plans are executed, and according to the plans they are moved on the network. As output of the simulation, events are produced allowing to calculate travel time, speed, etc. In general, an agent is moved to the next link when it was on that link for at least the free flow travel time, according to the maximum outflow, and when there is space on the next lane. The mentioned networks attributes remain fixed, as mentioned above. More information about the queue simulation can be found in [53].

3.5. *Scoring*

Scoring a plan is a precondition so that agents learn. Different plans can be compared and an agent can pick the one with the highest value. A higher score implies that the agent makes better use of its day. A scoring function needs to be defined, which evaluates complete day plans. As scoring function, the traditional utility function based on the Vickrey bottleneck model is used [54], but modified to be consistent with complete day plans. Scoring is based on events information from the physical layer. Performing an activity is rewarded, travel times and late arrival are punished. The overall equation is:

$$U_{plan} = \sum_i U_{act,i} + \sum_i U_{trav,i} + \sum_i U_{late,i} \; . \tag{1}$$

We assume the utility of performing an activity as increasing logarithmically:

$$U_{act,i}(x) = \max[0, \alpha \cdot \ln(\frac{x}{t_0})] \; , \tag{2}$$

where x is the duration that one spends at the activity. We take $\alpha = \beta_{dur} \cdot t^*$, where β_{dur} is uniformly the same for all activities and only t^* varies between activity types. With this formulation, t^* can be interpreted as a "typical" duration, and β_{dur} as the marginal utility at that typical duration:

$$\left.\frac{\partial U_{act,i}}{\partial x}\right|_{x=t^*} = \beta_{dur} \cdot t^* \cdot \frac{1}{t^*} = \beta_{dur} \; . \tag{3}$$

t_0 can be seen as a minimum duration of an activity, but is better interpreted as a priority: All other things being equal, activities with large t_0 are less likely to be dropped than activities with small t_0. For details, see [55].

The utilities of traveling and of being late are both seen as disutilities which are linear in time:

$$U_{trav,i}(x) = \beta_{trav} \cdot x \qquad (4)$$

(where x is the time spent traveling) and

$$U_{late,i}(x) = \beta_{late} \cdot x \qquad (5)$$

(where x is the time an agent arrives late at an activity). β_{trav} is set to $-6Eu/h$, and β_{late} is set to $-18Eu/h$.

In principle, arriving early or leaving early could also be punished. There is, however, no immediate need to punish early arrival, since waiting times are already indirectly punished by foregoing the reward that could be accumulated by doing an activity instead (opportunity cost). In consequence, the effective (dis)utility of waiting is already $-6Eu/h$.

Similarly, that opportunity cost has to be added to the time spent traveling, arriving at an effective (dis)utility of traveling of $-12Eu/h$.

No opportunity cost needs to be added to late arrivals, because the late arrival time is already spent somewhere else. In consequence, the effective (dis)utility of arriving late remains at $-18Eu/h$.

These effective values are the standard values of the Vickrey model [54].

It would make sense to consider an additional punishment (negative reward) for leaving an activity early. This would describe, for example, the effect when there are, on a specific day, better things to do than to continue to work, but some kind of contract (e.g. shop opening hours) forces the agent to remain at work.

If a new plan is built, an agent will execute it in the next iteration in order to obtain a score. In general every plan is scored after being executed. A formerly good plan can be scored at lower values if conditions change, e.g. congestion effects.

3.6. *Details of the learning algorithm*

The simulation starts with initial plans. Executing all agents' plans simultaneously gives the agents' interactions in the network. By allowing the agents to re-adjust, they can learn from the previous iteration (feedback learning). The iterations will go on until the system does not show any further development. In other words, agents adapt to their environment and learn how to improve their plans over many iterations. In the simulation all agents learn at the same time, since their plans are executed simultaneously. This also means, that an agent's environment changes due to the effect of the other agents in the system. Thus a plan's score has to be updated.

An agent database keeps track of agents and their decisions, allowing them to chose a strategy based on their past actions. An agent can compare plans of

12   *Beuck et al.*

its repertoire by the score they got in previous iterations. In the course of the simulation, the agents learn to build good plans in order to realize their intentions and to use the transportation system efficiently. Agents add plans (their strategies) to their repertoire by making use of the behavioral modules. A new plan will be used immediately in order to assign a score to make it comparable to the plans already existing in the repertoire. It can be expected that the average plan score will increase during the simulation until reaching a level were the agents have found their individually best strategies.

The agents have three different possibilities to replan: route replanning, time replanning, choosing an already existing plan. As already mentioned, only a certain share of agents replan. The replanning probability is not fixed. The simulation starts with 30% of agents replanning; each of the replan options is adopted by 10% of the agents. This is a relatively high share of agents changing their behavior and by that changing the environment for the other agents as well. But this rather high replanning probability provides a quick learning process; the agents build a repertoire of plan alternatives. Later in simulation, the replanning probability is lowered to a value of 15% (5% for each replan option). This gives better average scores because of reduced fluctuations (see Fig.2 and related text).
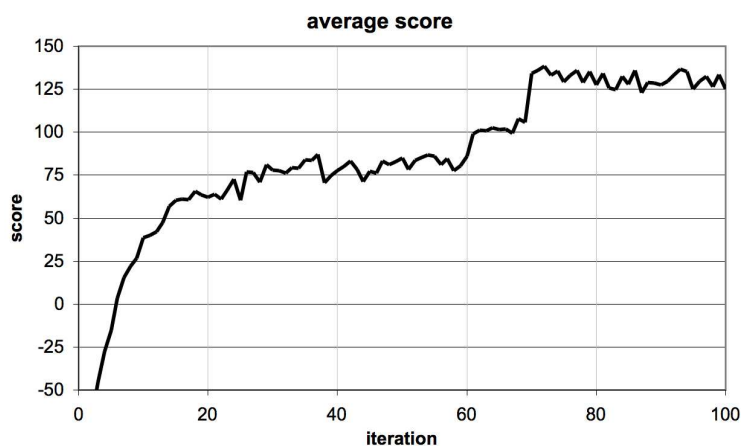
4. Preliminary results

Fig. 2. Average score as a function of the iteration number

The average score gives an overview of the iterations' progress (Fig.2). As expected, the average score is very low at the stage of initial plans, meaning the agents, in average, have not yet found good solutions for themselves. But the agents learn how to improve their situation by using different routes or changing their timing.

The higher replanning probability in the beginning allows a large share of agents to learn. When the average score does not show further improvements but oscillates, the replanning probability was set to half of the original value. At iteration 60 the reconfiguration was set, which can be also seen in the figure by improving scores around iteration 60. Also with the lower replanning probability fluctuations occur, but this can be also observed in real traffic.
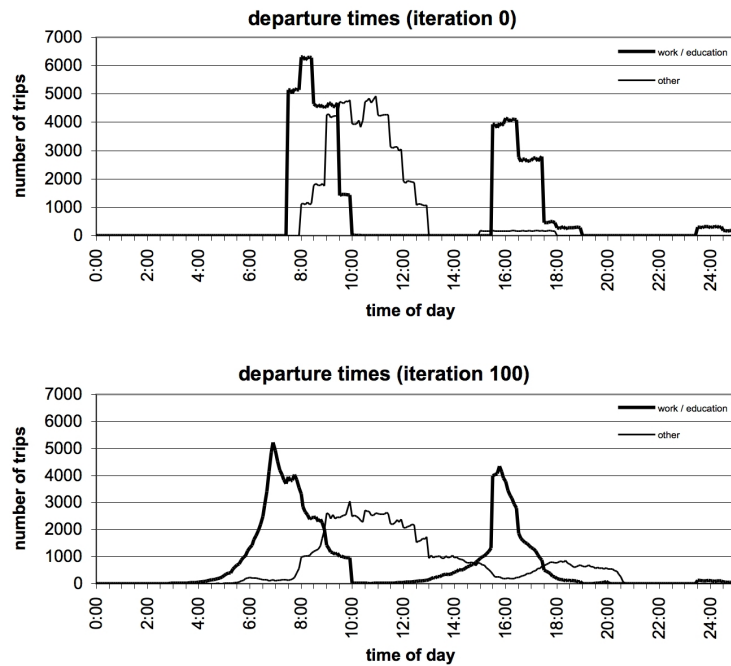

Fig. 3. Departure times distributions

Fig.3 shows the departure time distribution of two trip types: to or from work or education, and all others. The top plot shows the initial (iteration 0) departure times, based on heuristic expert knowledge, encoded in the initial conditions. The bottom plot shows the departure times after one hundred iterations. One notices the following effects:

- The initial rectangular shapes are replaced by more plausible smooth shapes.
- Travelers have, in average, moved to *earlier* departure times, with a peak before 7:00. We are, at this point, unable to judge if this is realistic.
- Those trips that are not coupled to office hours have moved to less congested time windows. Notice, in particular, the "dip" of those departures around 16:00, clearly avoiding the rush period.

14   *Beuck et al.*

Fig. 4. Daily flows of vehicles. Green: less than 10,000/day; yellow: around 20,000/day; red: more than 30,000/day

Fig.4 shows daily flows of vehicles. Flows below 10,000/day are displayed in green, flows of 20,000/day are displayed in yellow, flows above 30,000/day are displayed in red. Flows in between are displayed in interpolated colors. The figure shows the result of iteration 100. Since the simulation uses only 10% of the population, the numbers from the simulation were multiplied by 10 in order to have the same scale as real world numbers. – One observes that the pattern in the south-western sector is significantly different from the pattern in the north-eastern sector: While in the south-western half there is considerable traffic on the peripheral freeway, the patterns in the north-eastern sector are considerably more radial. This is due to extended freeway construction in the western sector during the division of the city, and the lack of such construction in the eastern sector.

It is, unfortunately, at this point difficult to say anything beyond the above. After spending considerable effort cleaning up other issues, such as related to network data or initial demand, our current issue are gridlocks as shown in Fig.5. The problem here is that all links along the loop are full, and all vehicles that are at the respective downstream ends of links want to enter the next link of the loop. In this situation, no vehicle along the loop can move, which is why it is called gridlock. Such situations can in principle occur along any closed loop of the network graph, but have a much higher probability along short loops.

We have been aware of the gridlock issue for many years, and have conventionally resolved it by the introduction of "lost vehicles": Vehicles that could not move for a
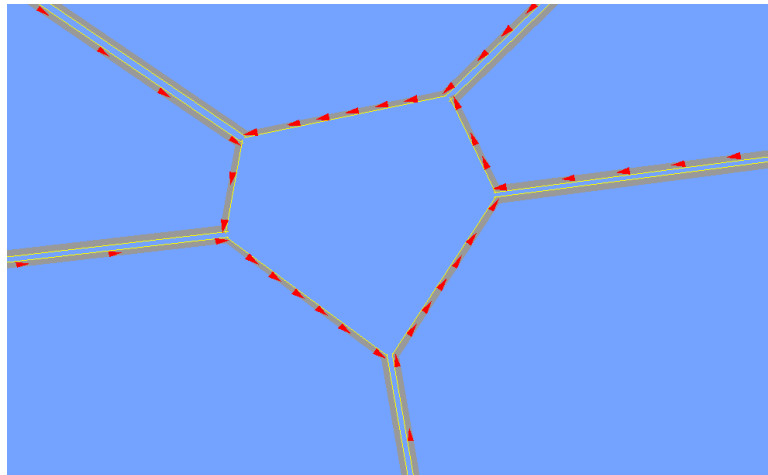
Fig. 5. Gridlock at *Grosser Stern* in Berlin

certain amount of time were taken out of the simulation. This approach, however, does not seem to be sufficient for the Berlin simulations, and we are currently investigating other solutions.

5. Discussion

Considerable work was necessary to adapt the network data to our purposes. Although the data requirements of the queue model are not particularly difficult (apart from the number of lanes, all information is the same as for traditional planning models), it turns out that the queue model is more sensitive to data errors than the conventional models. This is due to the hard limit on the capacity: In a conventional static assignment model, short links with reduced capacities have very little effect, whereas in the queue model, they cause large spillbacks. This effect occurs in all dynamic models with hard capacity limits. Our hope is that some of these issues will improve with the increasing availability of standardized commercially maintained network data. For the time being, however, such data are useful for routing and guidance, but do not possess reliable attributes (such as capacity) for traffic flow simulations.

Our demand generation suffers from the fact that the base model generates round trips, not daily plans. In consequence, a person who has, say, the activity chain home-work-home-leisure-home will be divided into two "virtual" agents, one with activity chain home-work-home and the other with activity chain home-leisure-home. There is no reason why those two virtual agents should perform their trips in a sequential order, so in general they will, wrongly, not do so. This issue is due to the orientation of the demand generation towards daily travel, without consideration of the time-of-day. It will probably be necessary to devise a completely new method

of demand generation.

An additional problem is that our simulations currently lack commercial travel and long-distance travel. Commercial travel, in particular, increases the overall demand.

Quite in general, it is difficult to get temporally consistent data – the data that we are currently using as input comes from many different years. In most places, things do not change that quickly, and it is sufficient to have the road network data and the traffic counts from the same year. Berlin, however, is a quickly-changing city due to the re-unification, and in consequence, such differences matter considerably.

6. Summary and conclusion

This paper provided an introduction to multi-agent traffic simulation. It included some description of where we are with respect to the implementation of a real-world Berlin scenario. It turns out that the difficulties there, because of data availability and because of the special situation of Berlin after the re-unification, are considerably larger than in previous scenarios that we have treated.

References

[1] D.M. Beazley, P.S. Lomdahl, N. Gronbech-Jensen, R. Giles, and P. Tamayo. Parallel algorithms for short-range molecular dynamics. In D. Stauffer, editor, *Annual reviews of computational physics III*, pages 119–176. World Scientific, 1995.

[2] D. Stauffer. Computer simulations of cellular automata. *J. Phys. A*, 24:909–927, 1991.

[3] K. Nagel and M. Rickert. Parallel implementation of the TRANSIMS microsimulation. *Parallel Computing*, 27(12):1611–1639, 2001.

[4] B.S. Kerner, S.L. Klenov, and D.E. Wolf. Cellular automata approach to three-phase traffic theory. *J. Physics A: Math. General*, 35(47):9971–10013, 2002.

[5] D. Helbing, A. Hennecke, and M. Treiber. Phase diagram of traffic states in the presence of inhomogeneities. *PRL*, 82(21):4360–4363, 1999.

[6] D. Jost and K. Nagel. Probabilistic traffic flow breakdown in stochastic car following models. *Transportation Research Record*, 1852:152–158, 2003.

[7] M. J. Lighthill and J. B. Whitham. On kinematic waves. I: Flow movement in long rivers. II: A Theory of traffic flow on long crowded roads. *Proceedings of the Royal Society A*, 229:281–345, 1955.

[8] K. Nagel and P. Nelson. A critical comparison of the kinematic-wave model with observational data. In H.S. Mahmassani, editor, *Transportation and Traffic Theory – Flow, Dynamics and Human Interaction*, pages 145–164. Elsevier, 2005.

[9] E. Brockfeld, R. Kühne, A. Skabardonis, and P. Wagner. Towards a benchmarking of microscopic traffic flow models. *Transportation Research Record*, 1852:124–129, 2003.

[10] F.L. Hall and K. Agyemang-Duah. Freeway capacity drop and the definition of capacity. *Transportation Research Record*, 1320:91–98, 1991.

[11] B. S. Kerner. Experimental features of self-organization in traffic flow. *Physical Review Letters*, 81:3797–3800, 1998.

[12] M. Bando, K. Hasebe, K. Nakanishi, A. Nakayama, A. Shibatan, and Y. Sugiyama. Phenomenological study of dynamical model of traffic flow. *Journal de Physique I France*, 5(11):1389–1399, 1995.

[13] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews of Modern Physics*, 73:1067–1141, 2001.

[14] P. G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research B*, 15:105–111, 1981.

[15] S. Krauß, P. Wagner, and C. Gawron. Continuous limit of the Nagel-Schreckenberg model. *Phys. Rev. E*, 54(4):3707–3712, 1996.

[16] C. Gloor, L. Mauron, and K. Nagel. A pedestrian simulation for hiking in the Alps. In STRC'03 [56].

[17] 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.

[18] M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani. Dynamit: A simulation-based system for traffic prediction. DACCORS Short Term Forecasting Workshop, The Netherlands, February 1998. See also its.mit.edu.

[19] G.L. Chang, H.S. Mahmassani, and R. Herman. A macroparticle traffic simulation model to investigate peak-period commuter decision dynamics. *Transportation Research Record*, 1005:107–120, 1985.

[20] G. Flötteröd and K. Nagel. Some practical extensions to the cell transmission model. Paper 128, IEEE Intelligent Transportation Systems Conference, Vienna, 2005. ISBN 0-7803-9216-7.

[21] C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, 9(3):393–407, 1998.

[22] SUMO www page, accessed 2005.

[23] DYNASMART www page, accessed 2005.

[24] DYNAMIT www page, accessed 2005.

[25] A. de Palma and F. Marchal. Real case applications of the fully dynamic METROPOLIS tool-box: An advocacy for large-scale mesoscopic transportation systems. *Networks and Spatial Economics*, 2(4):347–369, 2002.

[26] TRANSIMS www page. TRansportation ANalysis and SIMulation System. transims.tsasa.lanl.gov, accessed 2005. Los Alamos National Laboratory, Los Alamos, NM.

[27] J. Esser. *Simulation von Stadtverkehr auf der Basis zellularer Automaten.* PhD thesis, University of Duisburg, Germany, 1998. See also www.traffic.uni-duisburg.de.

[28] MATSIM www page. MultiAgent Transportation SIMulation. See www.matsim.org, accessed 2005.

[29] E.J. Miller and M.J. Roorda. A prototype model of household activity/travel scheduling. *Transportation Research Record*, 1831:114–121, 2003.

[30] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269 – 271, 1959.

[31] M. Ben-Akiva and S. R. Lerman. *Discrete choice analysis*. The MIT Press, Cambridge, MA, 1985.

[32] E. Cascetta and A. Papola. An implicit availability/perception random utility model for path choice. In *Proceedings of TRISTAN III*, volume 2, San Juan, Puerto Rico, 1998.

[33] R.M. Pendyala and R. Kitamura. FAMOS: The Florida Activity Mobility Simulator. In H.J.P. Timmermans, editor, *Progress in activity-based analysis*. Elsevier, Oxford, UK, In press.

[34] C.R. Bhat, J.Y. Guo, S. Srinivasan, and A. Sivakumar. A comprehensive econometric microsimulator for daily activity-travel patterns (cemdap). *Transportation Research Record*, forthcoming. Also see www.ce.utexas.edu/prof/bhat/FULL_REPORTS.htm.

[35] J.L. Bowman, M. Bradley, Y. Shiftan, T.K. Lawton, and M. Ben-Akiva. Demonstration of an activity-based model for Portland. In *World Transport Research: Selected Proceedings of the 8th World Conference on Transport Research 1998*, volume 3, pages 171–184. Elsevier, Oxford, 1999.

[36] J. Jonnalagadda, N. Freedman, W.A. Davidson, and J.D. Hunt. Development of microsimulation activity-based model for San Francisco: destination and mode choice models. *Transportation Research Record*, 1777:25–35, 2001.

[37] T.A. Arentze and H.J.P. Timmermans. *Albatross: A Learning-Based Transportation Oriented Simulation System*. EIRASS, Eindhoven, NL, 2000.

[38] Th. Arentze and H.J.P. Timmermans. *ALBATROSS – Version 2.0 – A learning based transportation oriented simulation system*. EIRASS (European Institute of Retailing and Services Studies), TU Eindhoven, NL, 2005.

[39] R. J. Beckman, K. A. Baggerly, and M. D. McKay. Creating synthetic base-line populations. *Transportion Research Part A – Policy and Practice*, 30(6):415–429, 1996.

[40] P.A. Salvini and E.J. Miller. ILUTE: An operational prototype of a comprehensive microsimulation model of urban systems. In *Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR)* [57]. See www.ivt.baum.ethz.ch.

[41] P. Waddell, A. Borning, M. Noth, N. Freier, M. Becke, and G. Ulfarsson. Microsimulation of urban development and location choices: Design and implementation of UrbanSim. *Networks and Spatial Economics*, 3(1):43–67, 2003.

[42] D. Strauch, G. Hertkorn, P. Wagner, and R.D. Kühne. Neue Ansätze zu einer mikroskopisch-dynamischen Verkehrs- und Flächennutzungsplanung im

verbundprojekt ILUMASS. In D. Tavangarian and R. Grützner, editors, *Simulationstechnik*, volume 12 of *Frontiers in Simulation*, pages 511–516. SCS Publishing House, 2002.

[43] R.H.M. Emmerink. *Information and pricing in road transport*. PhD thesis.

[44] G. Flötteröd and K. Nagel. Bayesian modeling and simulation of combined route and activity location choice. Toronto, 2006. Accepted. Also see www.vsp.tu-berlin.de/publications.

[45] S. Jara-Diaz et al. Modelling activity duration and travel choice from a common microeconomic framework. In *Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR)* [57]. See www.ivt.baum.ethz.ch.

[46] J. Hofbauer and K. Sigmund. *Evolutionary games and replicator dynamics*. Cambridge University Press, 1998.

[47] H. G. Schuster. *Deterministic Chaos: An Introduction*. Wiley-VCH Verlag GmbH, 1995.

[48] E. Kutter and H-J. Mikota. Weiterentwicklung des Personenverkehrsmodells Berlin auf der Basis der Verkehrsentstehungsmatrix 1986 (BVG). Technical report, Untersuchung im Auftrag des Senators für Arbeit, Verkehr und Betriebe, Berlin, 1990.

[49] E. Kutter. Integrierte Berechnung städtischen Personenverkehrs – Dokumentation der Entwicklung eines Verkehrsberechnungsmodells für die Verkehrsentwicklungsplanung Berlin (West). Arbeitsberichte zur Integrierten Verkehrsplanung, TU Berlin, Berlin, 1984.

[50] E. Kutter, H-J. Mikota, J. Rümenapp, and I. Steinmeyer. Untersuchung auf der Basis der Haushaltsbefragung 1998 (Berlin und Umland) zur Aktualisierung des Modells "Pers Verk Berlin / RPlan", sowie speziell der Entwicklung der Verhaltensparameter '86–'98 im Westteil Berlins, der Validierung bisheriger Hypothesen zum Verhalten im Ostteil, der Bestimmung von Verhaltensparametern für das Umland. Draft of the final report, Sponsored by the "Senatsverwaltung für Stadtentwicklung Berlin", Berlin/Hamburg, 2002.

[51] J. Rümenapp and I. Steinmeyer. Activity-based demand generation: Anwendung des Berliner Personenverkehrsmodells zur Erzeugung von Aktivitätenketten als Input für Multi-Agenten-Simulationen. Working Paper 06-09, Fachgebiet Verkehrssystemplanung und Verkehrstelematik, TU Berlin, Berlin, Germany, 2006.

[52] K. Meister, M. Balmer, K.W. Axhausen, and K. Nagel. planomat: A comprehensive scheduler for a large-scale multi-agent transportation simulation. In *Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR)*, Kyoto, Japan, 2006. See www.iatbr.org.

[53] N. Cetin, A. Burri, and K. Nagel. A large-scale agent-based traffic microsimulation based on queue model. In STRC'03 [56]. Earlier version, with inferior performance values: Transportation Research Board Annual Meeting 2003 paper number 03-4272.

20   *REFERENCES*

[54] R. Arnott, A. De Palma, and R. Lindsey. A structural model of peak-period congestion: A traffic bottleneck with elastic demand. *The American Economic Review*, 83(1):161, 1993.

[55] D. Charypar and K. Nagel. Generating complete all-day activity plans with genetic algorithms. *Transportation*, 32(4):369–397, 2005.

[56] *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, CH, 2003.

[57] *Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR)*, Lucerne, Switzerland, 2003. See www.ivt.baum.ethz.ch.