An Agent-Based Simulation Model of Swiss Travel: First Results

Bryan Raney, Dept. of Computer Science, ETH Zürich
Kai Nagel, Dept. of Computer Science, ETH Zürich

STRC 03 Conference paper

STRC | **3rd Swiss Transport Research Conference**
Monte Verità / Ascona, March 19-21, 2003

An Agent-Based Simulation Model of Swiss Travel: First Results

Bryan Raney
Department of Computer Science
ETH Zürich
CH-8092 Zürich, Switzerland

Phone: 01-632 08 92
Fax: 01-632 13 74
eMail: raney@inf.ethz.ch

Kai Nagel
Department of Computer Science
ETH Zürich
CH-8092 Zürich, Switzerland

Phone: 01-632 54 27
Fax: 01-632 13 74
eMail: nagel@inf.ethz.ch

Abstract

In a multi-agent transportation simulation, travelers are represented as individual "agents," who make independent decisions about their actions. We are implementing such a simulation for all of Switzerland, which is composed of modules that model those decisions for each agent, such as: (i) **Activities generator**, which generates a complete 24-hour day-plan, with each major activity (sleep, eat, work, shop, drink beer), their times, and their locations. (ii) **Route planner**, which determines the mode of transportation, as well as the actual route plan taken, for each leg of the agent's chosen activity plan. (iii) **Mobility simulation**, which executes all plans simultaneously and in consequence computes the interaction between different travelers, leading e.g. to congestion. (iv) **Feedback and learning**, which resolves the interdependence between the above modules. For example, plans depend on congestion but congestion depends on plans. This is resolved via an iterative method, where an initial plans set is slowly adapted until it is consistent with the resulting travel conditions. This technique has similarities to day-to-day human learning and can also be interpreted that way. – Besides these modules, one also needs input data, such as the road network, or (synthetic) populations. In the future, further modules need to be added, such as for housing and land use, or for freight traffic.

We discuss the operation and interaction of these modules, and our implementation of feedback via an **agent database** that gives each agent a "memory" of its past plans from earlier iterations, plus the performance of those plans. When a new plan-set is generated, each agent chooses a plan from those in its memory, based on their relative performance.

We present results of testing the above set-up, without the activities generator, for Swiss morning peak traffic. Hourly demand matrices were taken from work with static assignment models and converted to our needs. Routes were assigned via feedback learning using the agent database. Resulting flow volumes are compared to the static assignment results, and to field data. We conclude that the above set-up is at least as good as the assignment result.

Keywords

traffic simulation– transportation planning– route planning– learning– multi-agent simulation– 3rd Swiss Transport Research Conference – STRC 03 – Monte Verità

1. Introduction

There is by now some agreement that multi-agent simulation may be a viable technology for traffic forecasting. “Multi-agent” means that each traveler, and potentially each entity of the simulation, such as, say, traffic lights or variable message signs, are represented as individual objects or “agents,” which make independent decisions about their actions.

A multi-agent simulation of a real-world system consists, at least conceptually, of the following two parts: (i) the simulation of the “physical” properties of the system; and (ii) the generation of the agents’ strategies. For traffic applications, the first is the micro-simulation of the traffic system, where, say, travelers walk from home to their cars, drive the car to the park-and-ride parking lot, walk to the train, take the train to the city, and then walk to work. This part of the simulation should take care of the physical constraints of the system, such as acceleration or speed limits, capacity limits, storage limits (such as number of vehicles on a link, or number of passengers in a train), etc. The meaning of “strategies” for multi-agent traffic simulation is that travelers have plans, for example for activities or routes, that they execute during a day. At the same time, they may also modify those plans.

We are implementing such a multi-agent simulation for all of Switzerland, which, with about 7 million inhabitants, also serves as a proxy for a large metropolitan area. The challenges with such an implementation are many: availability and quality of input data, computational implementation and computational performance, conceptual understanding of agent learning, and validation. This paper reports first results which are essentially based on typical transportation planning data: it uses standard origin-destination matrices; it uses the transportation planning network from the corresponding Swiss federal planning authority; and it performs route assignment based on these input data. The main difference to typical planning tools, such as EMME/2 or VISUM, is that our implementation is indeed completely agent-based, that is, the individuality of each traveler is fully maintained throughout the process.

At this stage, our approach is somewhat similar to ITS (Intelligent Transportation System) simulations such as DYNAMIT or DYNASMART.[1] Currently, the most important differences are: (i) Both our target size and the actual feasible scenario size are considerably larger than for existing ITS simulations. (ii) Most, if not all, ITS simulation implementations group travelers by common characteristics; for example, travelers are differentiated by destination, but all travelers to the same destination take the same route or the same set of routes. In contrast, our approach is *completely* agent-based, that is, there is absolutely no internal relation between travelers who are going to the same destination.

This paper will continue with an outline of the simulation structure (Sec. 2), of the mobility simulation (Sec. 3), and of the route/strategy generation module (Sec. 4). This is followed by a longer section on agent-based learning (Sec. 5). The following two sections (Secs. 6 and 7) describe the real-world test scenario and results obtained with it, including a comparison to VISUM results based on the same scenario. The paper is concluded by a discussion/outlook and a summary.

[1] See www.dynamictrafficassignment.org for documentation on both.

2. Simulation Structure

Traffic simulations for transportation planning typically consist of the following modules (Fig. 1):

- **Population generation**. Demographic data is disaggregated so that one obtains individual households and individual household members, with certain characteristics, such as a street address, car ownership, or household income (Beckman *et al.*, 1996). – This module is not used for our current investigations but will be used in the future.

- **Activities generation**. For each individual, a set of activities (home, going shopping, going to work, etc.) and activity locations for a day is generated (Vaughn *et al.*, 1997; Bowman, 1998). – This module is not used in our current investigations but will be used in the future.

- **Modal and route choice**. For each individual, the modes are selected and routes are generated that connect activities at different locations (see Sec. 4). The routing should be dynamic in order to adequately model time-dependent congestion effects.

- **Mobility simulation**. Up to here, all individuals have made *plans* (or *strategies*) about their behavior. The mobility simulation executes all those plans simultaneously (see Sec. 3). In particular, we now obtain the result of *interactions* between the plans – for example congestion.

- **Feedback**. In addition, such an approach needs to make the modules consistent with each other (Sec. 5). For example, plans depend on congestion, but congestion depends on plans. A widely accepted method to resolve this is systematic relaxation (Kaufman *et al.*, 1991; Nagel, 1994/95; Bottom, 2000) – that is, make preliminary plans, run the mobility simulation, adapt the plans, run the simulation again, etc., until consistency between modules is reached. The method is somewhat similar to the Frank-Wolfe-algorithm in static assignment, or in more general terms to a standard relaxation technique in numerical analysis.

This modularization has in fact been used for a long time; the main difference to earlier implementations is that it is now feasible to make all modules completely microscopic, i.e. each traveler is individually represented in all modules.

As of now, not all of the above modules are currently implemented. This paper discusses results obtained with a version of the simulation system that consists of car-only versions of the router, the mobility simulation, and the feedback. These modules will be described in more detail in the following sections. It should be noted that in particular the feedback system is unique in that it explicitly keeps track of many strategies of each individual traveler. Most simulation systems assume either only one strategy per traveler, or they group travelers together according to their characteristics, for example by common destination. Since the activity generation module is currently not used, demand is obtained from traditional origin-destination matrices. This will be further discussed in conjunction with the scenario, in Sec. 6.

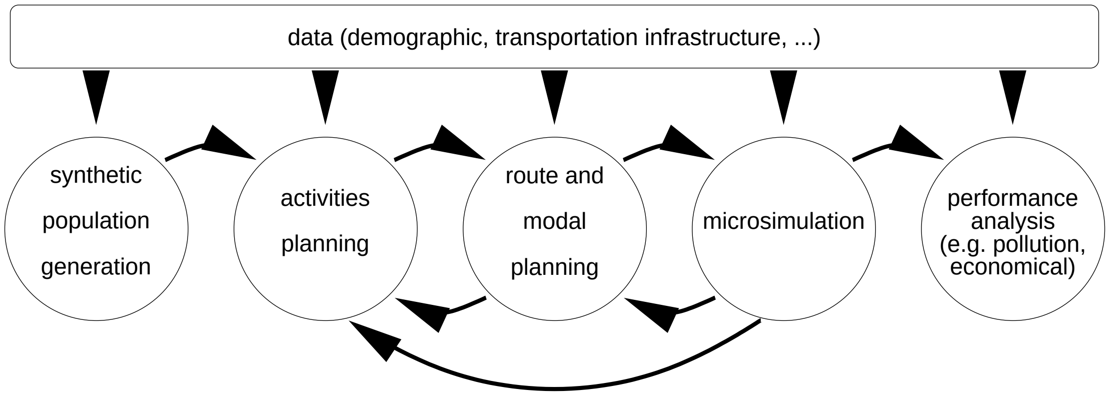

Figure 1: TRANSIMS Modules

3. Mobility Simulation

Our main mobility simulation is the queue simulation (Gawron, 1998; Cetin and Nagel, 2003b,a). The intent with this simulation is to keep it as simple as possible while maintaining a microscopic view of the travelers (agents/vehicles) by following the individualized route plans they generate (see Sec. 4); and to have queue spillback to model the indirect interactions between agents. This is similar in spirit to traffic simulations based on the smooth particle hydrodynamics approach, such as DYNEMO (Schwerdtfeger, 1987), DYNAMIT (see earlier), or DYNASMART (see earlier).

In the queue simulation, streets are essentially represented as FIFO (first-in first-out) queues, with the additional restrictions that (1) vehicles have to remain for a certain time on the link, corresponding to free speed travel time; and that (2) there is a link storage capacity and once that is exhausted, no more vehicles can enter the link.

Keeping the simulation simple also means that we do not perform any data aggregation within the simulation. It simply dumps out raw data, in the form of “events,” which tell the time and location where something interesting happens to an agent. These events can be parsed and aggregated, if necessary, by the other modules to obtain any information they may need about what happened in the simulation. At present the simulation only outputs events for agents entering or exiting a link, but other interesting events might be when agents encounter congestion, change speed, etc.

A major advantage of the queue simulation, besides its simplicity, is that it can run directly off the data typically available for transportation planning purposes. This is no longer true for more realistic simulations, which need, for example, the number of lanes including pocket and weaving lanes, turn connectivities across intersections, or signal schedules.

4. Strategy Generation

The mobility simulation computes the physical aspects of movement, such as limits on capacity, storage, or speed. In particular it computes the aspects of interaction, such as congestion. The mobility simulation needs information about where travelers enter and leave the network, which turns travelers take at intersections, etc. As mentioned in Sec. 2, these aspects can be called plans, or strategies. For the transportation simulation, this means that travelers know where they are going, when they want to be there, and the route they want to take to get there. This kind of strategic knowledge is in stark contrast to, say, the simulation of ants in an ant-hill. It also makes the simulation design considerably more demanding, since the generation and handling of strategies is a whole problem of its own. Our own approach to this problem is to allow a distributed design, that is, mobility simulation and strategy generation should be separated as much as possible, and in fact we also intend to have more than one strategy generation module in the future.

However, the only strategy generation module used for this paper is the route generation module. Travelers/vehicles need to compute the sequence of links (road segments) that they are taking through the network. A typical way to obtain such paths is to use a Dijkstra shortest path algorithm. This algorithm uses as input the network link travel times plus the starting and ending point of a trip, and generates as output the fastest path.

It is relatively straightforward to make the costs (link travel times) time dependent, meaning that the algorithm can include the effect that congestion is time-dependent: Trips starting at one time of the day will encounter different delay patterns than trips starting at another time of the day. Link travel times are aggregated from the events fed back from the mobility simulation into 15-min time bins, and the router finds the fastest route based on these 15-min time bins. Apart from relatively small and essential technical details, the implementation of such an algorithm is straightforward (Jacob *et al.*, 1999). It is possible to include public transportation into the routing (Barrett *et al.*, 2000); in our current work, we look at car traffic only.

As pointed out earlier, the ultimate goal of this work is to have a multi-agent simulation for transportation planning applications. This includes the generation of strategies which are on a higher level than routes, such as activity selection, location selection (for activities), and activity scheduling. Sec. 9 will discuss how it is planned to integrate activity generation as an additional strategy generation module in future versions.

5. Adaptation, Learning and Feedback

As is well known, there is a mutual dependence between strategy generation and mobility simulation. For example, congestion is the result of (the execution of) plans, but plans are based on (the anticipation of) congestion. The traditional approach to this kind of problem, both in transportation and in economics, has been the postulation of a Nash Equilibrium (NE). For route assignment, a NE is reached when no traveler can improve its travel time by selecting a different route. In the traditional static assignment approach, under some circumstances it can be proven that there is only one solution (in terms of the link flows) to this problem (e.g. Sheffi, 1985). In consequence, any computational procedure which finds that solution is a valid one.

An extension is the so-called Stochastic User Equilibrium (SUE; e.g. Sheffi, 1985). Instead of selecting the fastest path, travelers select a path according to a probability

$$p_i \propto e^{-\beta T_i} \,, \tag{1}$$

where T_i is the travel time of path i. β can be seen as a tuning parameter: For $\beta \to \infty$, standard static assignment is obtained again; for $\beta = 0$, all paths are selected with equal probability. The theoretical justification for β stems from the assumption of a certain variability of the travel times – which can stem from many sources, including real variability of the travel times, perceived variability of the travel times, or variability of so-called unobserved attributes. In practice, β is best obtained as part of a multinomial logit model estimation from stated or revealed preference data (e.g. Ben-Akiva and Lerman, 1985).

Static assignment does not possess any dynamics, that is, traffic is represented by time-independent streams. This precludes, for example, the representation of queue spill-back, or the representation of time-dependent traffic management strategies such as those used by ITS. As a result, newer work in this area, in particular when related to ITS, has used a dynamic representation of traffic. The NE or SUE approaches can however be maintained, in the sense that for a given departure time, different paths have different (average) travel times, and the traveler either selects the fastest path (NE) or she/he selects according to Eq. (1).

When moving to an agent-based approach, however, some aspects of the above approaches cannot be translated in a simple way. In particular, the SUE approach assumes that travelers know a set of different *plausible* paths, which is typically not available. In the following, two fully agent-based approaches to the problem will be presented. Both of them were already presented at last year's STRC (Raney and Nagel, 2002). It will be repeated here in order to make the paper self-consistent, and also in order to go again through the most important arguments.

5.1 "Basic" Agent-based Feedback

Both basic static assignment and SUE define a state of the system, but no algorithm to get there. Since, as noted above, the solution is unique (in terms of the link flows), *any* algorithm which solves the problem is valid. Most, if not all, of the algorithms turn out to be iterative.

Similarly, a possible way to solve the consistency problem between mobility simulation and strategy generation is to use systematic relaxation (Kaufman *et al.*, 1991; Nagel, 1994/95; Bottom, 2000). This is a cycle in which all agents select "strategies" (sets of choices, e.g. route plans), execute them in the traffic simulation, then some agents revise their strategies, and they are executed again, etc., until some kind of stopping criterion is fulfilled. A possible version of this:

1. The system begins with an initial set of routes (strategies), one per agent, based on free speed travel times, which represent a network with no congestion.

2. The traffic simulation is executed with the current set of strategies.

3. 10% of the population requests new routes from the router, which bases them on the updated link travel times from the last traffic simulation. The new routes then replace the old routes for the "replanned" agents in the set of current routes.

4. This cycle (i.e. steps 2 through 3) is run for 50 times; earlier investigations have shown that this is more than enough to reach relaxation (Rickert, 1998; Bottom, 2000).

5.2 The Agent Database

One problem with the basic approach is that it gives rise to oscillations from one iteration to the next: If, say, route A is faster in one iteration, then many travelers will switch to it, making it slower, and some other route faster. These oscillations can be suppressed by making the replanning fraction smaller with higher iteration numbers, but this is implausible with respect to human behavior.

As pointed out before, an alternative approach is to use Eq. (1) to chose between different strategies. As also noted before, the problem with this approach is that the set of routes for each origin-destination pair is assumed to be known before the start of the iterations. In the following, we present an approach where additional routes are found by the system as it goes, and are added to the repertoire. In addition, this exploration is done by each agent individually; this has particular advantages when agents are very diverse, as exemplified for example by a high-resolution network (each link is a possible starting and ending point), and a quickly changing temporal dynamics. This approach is called the "agent database".

The agent database gives the agents a *memory* of their past strategies, and the outcome (performance) of those strategies. The specific steps in the relaxation cycle are:

1. The system begins with an initial set of strategies, one per agent. In the case of the present results, a strategy is simply a route, and the initial set of routes is generated based on free speed travel times, which represent a network with no congestion.

2. For each agent, the new strategy is stored in the agent database (Raney and Nagel, 2002, 2003), which represents its memory of previously tried strategies. Since the agents have only one strategy apiece, they automatically select this as their next strategy to execute.

3. The traffic simulation is executed with the set of selected strategies.

4. Each agent measures the performance of his/her route based on the outcome of the simulation. "Performance" at present means the total travel time of the entire trip, with lower travel times meaning better performance. This information is stored for all the agents in the agent database, associated with the strategy that was used.

5. A fraction of the population (10% at present) requests new routes from the router, which bases them on the updated link travel times from the last traffic simulation. The new routes are then stored in the agent database and, being new, are mandatorily selected by the agents.

6. Agents who did not request new routes choose a previously tried route from the agent database, by comparing the performance values for the different routes, without knowing anything else about the routes. Specifically, they use a multinomial logit model

$$p_i \propto e^{-\beta T_i}$$

for the probability p_i to select route i, where T_i is the corresponding memorized travel time and β is an empirical constant. This process is explained in detail in Raney and Nagel (2002, submitted, 2003).

7. This cycle (i.e. steps 3 through 6) is run for 50 times.

This strategy means that more than our original 10% replanning fraction of the agents will change their plans at a given iteration. These changes will be "informed" decisions, though – not random exploration.

An additional advantage of the agent database is that the system is considerably more robust than without (Raney and Nagel, 2002, 2003, submitted). Without the agent database, it is imperative that the router generates paths which are an improvement over the previous solution, or said differently, the router needs to generate different paths with probabilities that reflect actual use of those different paths. This puts very high design requirements on the router that will be very difficult to fulfill. Use of the agent database means that the router can be much more creative in the generation of new routes: Routes which turn out to be bad strategies will just be evaluated once by the agent and then never be touched again.

The agent data-base brings our agent-based simulation closer to a classifier system as known for Complex Adaptive Systems (Stein and others, since 1988). From here, alternative methods of agent learning, such as Machine Learning or Artificial Intelligence, can be explored.

5.3 Illustration of Learning and Feedback

Figure 2 shows an example of how the feedback mechanism works in the so-called "Gotthard" scenario. This is a testing scenario in which 50 000 agents start between 6:00 AM and 7:00 AM from random locations all over Switzerland and drive to the same location in Lugano. By having all agents travel to the same destination, we can check if all traffic jams on all used routes to the destination dissolve in parallel or not. In order for the vehicles to get to their destination, most of them have to cross the Alps. However, there are not many ways to do this, resulting in traffic jams, most notably in the corridor leading toward the Gotthard pass. This scenario has some resemblance to real-world vacation traffic in Switzerland.

The figure shows two "snapshots" of the vehicle locations within the queue-based mobility simulation at 9:00 AM. The first image in the figure is a snapshot of the initial (zeroth) iteration of the simulation, and the second is the simulation after 50 iterations via the agent database feedback system described in Sec. 5.

Initially the travelers choose routes without any knowledge of the demand (caused by the other travelers), so they all use the fastest links, and tend to select very similar routes, which compose a subset of available routes. However, by driving on the same links, they cause congestion and those links become slower than the next-fastest links which were not selected. Thus, alternate routes which were marginally slower than the fastest route become, in hindsight, preferred to the routes taken. By allowing some travelers to select new routes using the new information about the network, and others to choose previously tried routes, we allow them to learn about the demand on the network caused by one another.

After 50 iterations between the route selection and the mobility simulation, the travelers have learned what everyone else is doing, and have chosen routes accordingly. Now a more complete set of the available routes is chosen, and overall the travelers arrive to their destination earlier than in the initial iteration. Comparing the usage of the roads, one can see that in the 49th iteration, the queues are shorter overall, and at the same time in the simulation, travelers are, on average, closer to their destination.

More results from this scenario can be found in Raney and Nagel (submitted, 2003).

6. Scenario (Initial and Boundary Conditions)

The goal of our work is a full 24-hour simulation of all of Switzerland, including transit traffic, freight traffic, and all modes of transportation. This will involve about 7.5 million travelers, and more than 20 million trips (including short pedestrian trips, etc.). A more short-term goal is a full 24-hour simulation of all of car traffic in Switzerland. For this, we will have about 10 million trips.

The present study takes a subset of the data for the full 24-hour, car-only simulation, and uses the demand for the morning rush-hour, from 6:00 AM to 9:00 AM. This subset contains about 1 million trips.

The input data consists of two parts: the street network, and the demand.

6.1 The Street Network

The street network that is used was originally developed for the Swiss regional planning authority (Bundesamt für Raumentwicklung), and covered Switzerland. It was extended with the major European transit corridors for a railway-related study (Vrtic *et al.*, 1999). The network supposedly contains the status for 1999, but contains at least one major error (a high capacity tunnel in Zürich is missing). Our initial simulations resulted in traffic gridlock in Zürich, which was also reflected in the VISUM assignment displaying V/C ratios significantly above 100%. A manual comparison with a higher resolution network of Zürich led to the conclusion that capacity in Zürich was in general significantly underestimated; in consequence, we manually increased the corresponding road capacity for transit corridors through Zürich in our network. We can only speculate what led to these network errors; Sec. 9 discusses our plans of how to improve the situation.

After our modifications, the network has the fairly typical size of 10 564 nodes and 28 624 links. Also fairly typical, the major attributes on these links are type, length, speed, and capacity. From

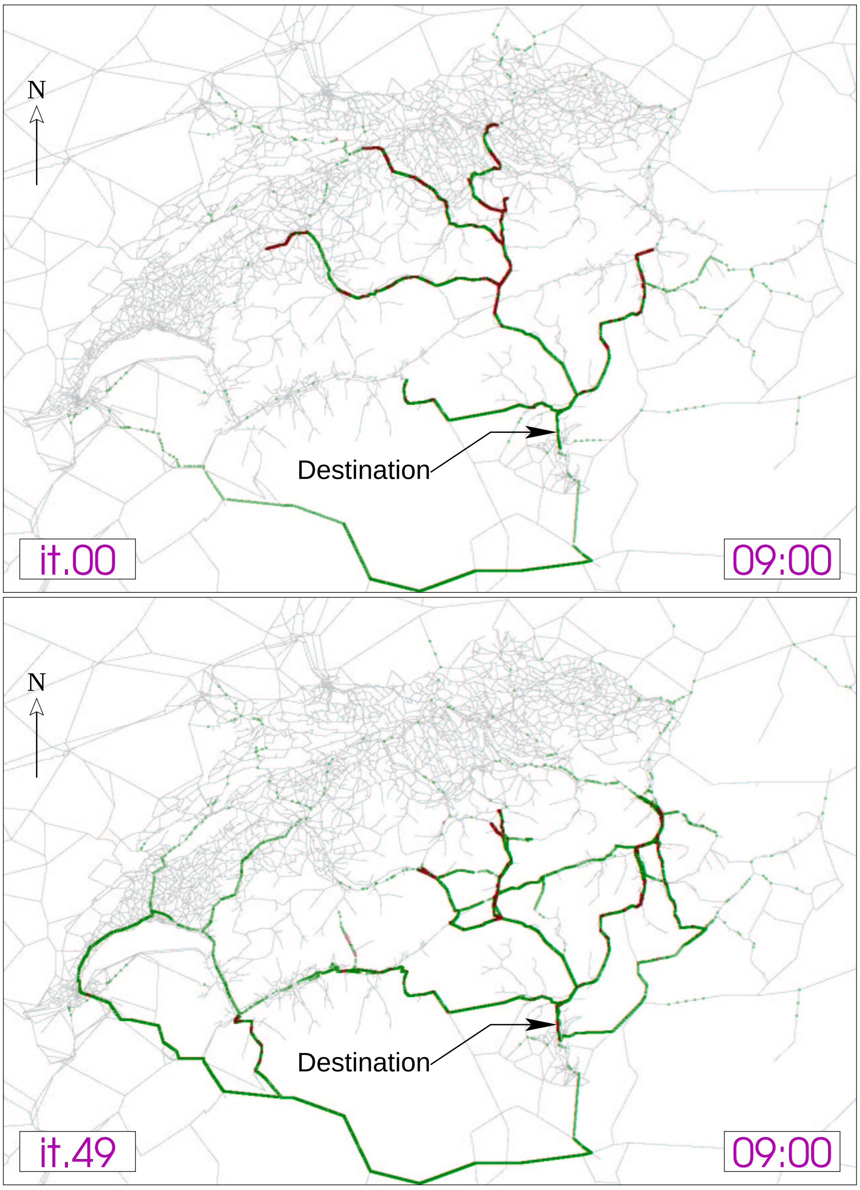

Figure 2: Example of Relaxation Due to Feedback. **TOP:** Iteration 0 at 9:00 – all travelers assume the network is empty. **BOTTOM:** Iteration 49 at 9:00 – travelers take more varied routes to try to avoid one another. Red (dark gray in b/w version) indicates jams, green (dark gray in b/w version) indicates free-flowing traffic, and gray indicates empty roads.

Source: Raney *et al.* (2003)

Sec. 3, one can see that this is enough information for the queue simulation.

6.2 Demand

Our starting point for demand generation for the full Switzerland scenario are 24-hour origin-destination matrices from the Swiss regional planning authority (Bundesamt für Raumentwicklung). Eventually, we intend to move on to activity-based demand generation.

The original 24-hour matrix is converted into 24 one-hour matrices using a three step heuristic (Vrtic and Axhausen, 2002). The first step employs departure time probabilities by population size of origin zone, population size of destination zone and network distance. These are calculated using the 1994 Swiss National Travel Survey (Bundesamt für Statistik und Dienst für Gesamtverkehrsfragen, 1996). The resulting 24 initial matrices are then corrected (calibrated) against available hourly counts using the OD-matrix estimation module of VISUM (www.ptv.de). Hourly counts are available from the counting stations on the national motorway system. Finally, the hourly matrices are rescaled so that the totals over 24 hours match the original 24h matrix.

VISUM assignment of the matrices shows that the patterns of congestion over time are realistic and consistent with the known patterns. The Zürich congestion problem, mentioned above, is contained in the assignment, but did not show up at this higher level view; see Sec. 9 for some discussion of this. A more detailed verification of these results was not possible so far, but is planned.

For the multi-agent simulation, these hourly matrices are then disaggregated into individual trips. That is, we generate individual trips such that summing up the trips would again result in the given OD matrix. The starting time for each trip is randomly selected between the starting and the ending time of the validity of the OD matrix.

The OD matrices assume traffic analysis zones (TAZs) while in our simulations trips start on links. We convert traffic analysis zones to links by the following heuristic:

- The geographic location of the zone is found via the geographical coordinate of its centroid given by the data base.
- A circle with radius 3 km is drawn around the centroid.
- Each link starting within this circle is now a possible starting link for the trips. One of these links is randomly selected and the trip start or end is assigned.

This leads to a list of approximately 5 million trips, or about 1 million trips between 6:00 AM and 9:00 AM. Since the origin-destination matrices are given on an hourly basis, these trips reflect the daily dynamics. Intra-zonal trips are not included in those matrices, as by tradition.

7. Results

The above scenario was fed into two different models: First, into a VISUM (PTV, www.ptv.de) assignment which is a relatively standard assignment (Sheffi, 1985) except that it is dynamic on

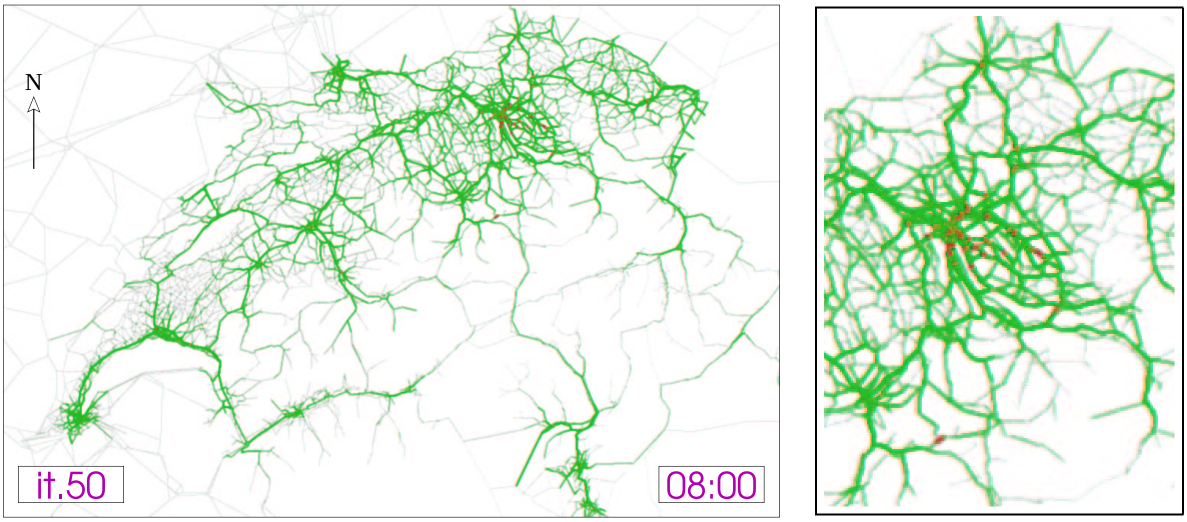

Figure 3: Snapshot of Switzerland at 8:00 AM. From the queue mobility simulation, iteration 50. The right side shows a close-up of the Zürich area.

Source: Raney *et al.* (2003)

an hourly basis (Vrtic and Axhausen, 2002), and second into an agent-based mobility simulation as described above, using the agent database and iterating about 50 times.

Figure 3 shows a result of the Switzerland 6-9 Scenario. This figure is after 50 iterations of the queue mobility simulation, using the agent database. We used as input the origin-destination matrices described in Sec. 6.2, but only the three one-hour matrices between 6:00 AM and 9:00 AM. This means any travelers beginning their trips outside this region of time were not modeled. As one would expect, there is more traffic near the cities than in the country. Jams are nearly exclusively found in or near Zürich (near the top; see close-up). As of now, it is unclear if this is a consequence of a higher imbalance between supply and demand than in other Swiss cities, or a consequence of a special sensitivity of the queue simulation to large congested networks.

Figure 4 shows a comparison between the simulation output of Fig. 3 and field data taken at counting stations throughout Switzerland (see Sec. 6.2 and Bundesamt für Strassen, 2000). The dotted lines, drawn above and below the central diagonal line, outline a region where the simulation data falls within 50% and 200% of the field data. We consider this an acceptable region at this stage since results from traditional assignment models that we are aware of are no better than this (Fig. 4(b); see also Esser and Nagel, 2001).

Figure 4(b) shows a comparison between the traffic volumes obtained by IVT using VISUM assignment against the same field data. Visually one would conclude that the simulation results are at least as good as the VISUM assignment results. Table 1 confirms this quantitatively. Mean absolute bias is $\langle q_{sim} - q_{field} \rangle$, mean absolute error is $\langle |q_{sim} - q_{field}| \rangle$, mean relative bias is $\langle (q_{sim} - q_{field})/q_{field} \rangle$, mean relative error is $\langle |q_{sim} - q_{field}|/q_{field} \rangle$, where $\langle . \rangle$ means that the values are averaged over all links where field results are available.

For example, the "mean relative bias" numbers mean that the simulation underestimates flows

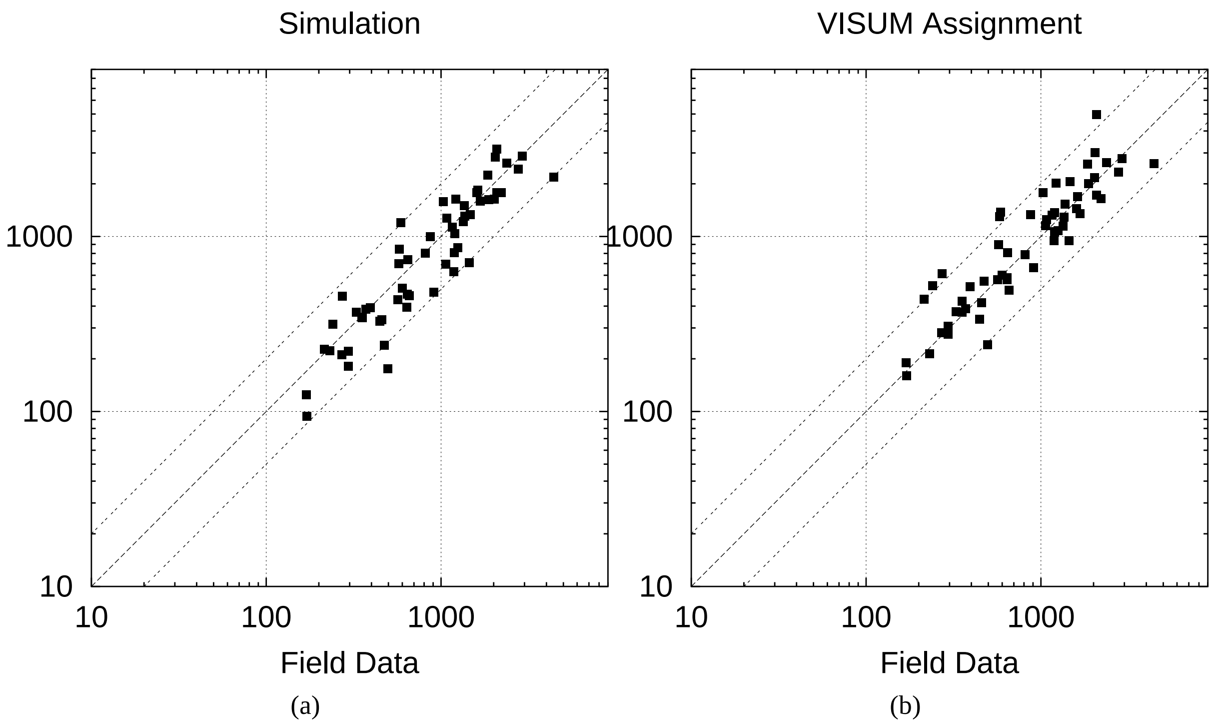

Figure 4: Comparison to Field Data. *(a)* Simulation vs. field data for the 50th iteration. The x-axis shows the hourly counts between 7:00 AM and 8:00 AM from the field data; the y-axis shows throughput on the corresponding link from the simulation. *(b)* VISUM assignment vs. field data. The x-axis is the same as *(a)*; the y-axis shows the volume obtained from the assignment model.

Source: Raney *et al.* (2003)

Table 1: Bias and Error of Simulation and VISUM Results Compared to Field Data

	Simulation	**VISUM**
Mean Abs. Bias:	−64.60	+99.02
Mean Rel. Bias:	−5.26%	+16.26%
Mean Abs. Error:	263.21	308.83
Mean Rel. Error:	25.38%	30.42%

Source: Raney *et al.* (2003)

by about 5%, whereas the VISUM assignment overestimates them by 16%. The average relative error between the field measurement and the simulation is 25%, between the VISUM assignment and reality 30%. These numbers state that the simulation result is better than the VISUM assignment result. Also, the simulation results are better than what we obtained with a recent (somewhat similar) simulation study in Portland/Oregon (Esser and Nagel, 2001); conversely, the assignment values in Portland were better than the ones obtained here.

What makes our result even stronger is the following aspect: The OD matrices were actually modified by a VISUM module to make the assignment result match the counts data as well as possible. These OD matrices were then fed into the simulation, without further adaptation. It is surprising that even under these conditions, which seem very advantageous for the VISUM assignment, the simulation generates a smaller mean error.

8. Computational Aspects

8.1 Computational Performance of the Mobility Simulation

Computational issues for the mobility simulation are discussed in detail elsewhere (Cetin and Nagel, 2003a,b). The main result from those investigations is that, using a Pentium cluster with 64 CPUs and Myrinet communication, the queue simulation can run more than 500 times faster than real time (excluding input and output), meaning that 24 hours of traffic of all of Switzerland can be simulated in less than 3 minutes. The queue simulation used for the results of this paper was still somewhat slower: It needed about 15 min, including input and output, for a simulation of traffic from 6:00 AM until roughly noon. Recall that this paper is concerned with the morning peak only.

8.2 Computational Performance of the Original Agent Database

A possible technology for information exchange between modules is to use temporary files. This is the technology used by TRANSIMS (www.transims.net). Eventually, however, there is the necessity to find certain data items quickly in these files. That problem can be solved by indexing, as TRANSIMS does. On the other hand, all these are typical database operations and thus it should be possible to use a typical database for these problems.

Figure 5 depicts the *cumulative* contributions of the major steps in the feedback system to the total execution time of each iteration. For this figure, the agent database was implemented using the MySQL public domain database (www.mysql.org), and both “strategy selection” and “score update” were database operations. Strategy generation (= route generation) and mobility simulation were done using our own C++ codes. Those operations are:

1. **Strategy Generation** adds a new strategy for 10% of the agents into the database. Once the network data has relaxed, this step takes about the same amount of time in each iteration.

2. **Strategy Selection**, where the other 90% of the agents select a strategy from the database.

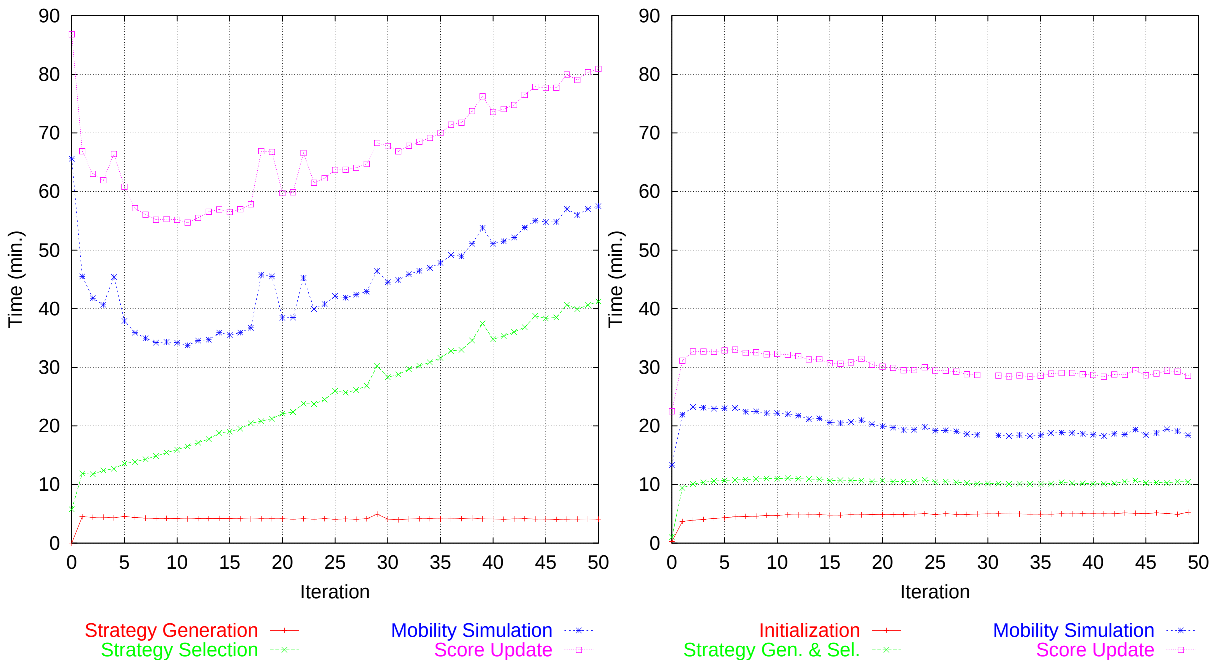

Figure 5: Cumulative Execution Time Contributions of Major Iteration Steps LEFT: File-based database implementation. RIGHT: The new implementation, which keeps all agent information in computer memory, and which also uses a faster mobility simulation.

This execution time for this step scales approximately linearly with the total number of strategies stored in the agent database. Thus, it takes longer to execute with each iteration.

3. **Mobility Simulation**, where the agents interact. This again relaxes to a consistent amount of time, about 15 minutes, though takes longer in earlier iterations when there is more congestion to deal with.

4. **Score Update**, where the agents update the performance scores of their executed strategy from the output of the mobility simulation. The execution time for this operation is fairly constant in each iteration, since it depends on the number of events produced by the simulation (see Sec. 3). This number is proportional to the number of agents and the length of their routes; it does not change very much from one iteration to the next.

More details about the database implementation and execution times can be found in Raney and Nagel (submitted, 2003).

One can see that on average each iteration takes about an hour to execute, with the feedback system, including strategy generation taking about 45 minutes of that time. The overall result is that we can run a metropolitan scenario with 1 million agents, including 50 learning iterations, in about two days.

8.3 Reimplementation of the Agent Database

As one can see from the results above, the file-based database approach is slow when used for large scale scenarios.

We are currently implementing a new approach where all relevant data (i.e. strategies) are permanently kept in computer memory. Rather than using slower scripting languages for each step in the iteration (Raney and Nagel, submitted, 2003), the new implementation is written in C++ and combines all operations into one program.

Preliminary results indicate that the computing time for this new implementation for the Switzerland 6-9 scenario improves, for the feedback and strategy generation components, from more than 60 minutes (in the 50th iteration) to about 20 minutes – see the right side of Fig. 5.

9. Discussion and Future Plans

9.1 Activity Generation

The above results use traditional origin-destination tables for demand generation. We intend to move our investigations to activity-based demand generation. One method will be based on discrete choice theory, one on genetic algorithms.

A fair amount of Swiss traffic is cross-border traffic, either with origin or destination in Switzerland, or completely traversing the country. Also, freight traffic would not be included in a first version of activity-based demand generation, which would concentrate on people. It is planned to include all these effects by conventional origin-destination matrices, i.e. via some "background" traffic that will be able to adjust routes (and maybe starting times) but will not be elastic in terms of number of trips.

9.2 Feedback

The use of the agent database in the feedback mechanism works well, but needs tuning. Both computational speed and the learning behavior of the system are an issue. The computational speed issues are being addressed as described in Sec. 8.3; memory issues are addressed below. The methodological questions will be addressed via an examination of established learning methods (such as best reply or reinforcement learning).

Another shortcoming of the current method is that replanning, and thus learning, can happen only over night. Work is under way to improve this situation via an online coupling between modules, which will allow within-day replanning (Gloor, 2001). This means that agents can also make strategic decisions while they are on the road and not just in between trips (see Fig. 6 for a visual sketch of this). Within-day learning is more realistic since some types of decisions are made on time scales much shorter than a day (Doherty and Axhausen, 1998). We explicitly want to avoid coupling the modules via standard subroutine/library calls, since this both violates the modular approach idea and efficiency considerations for parallel computing.

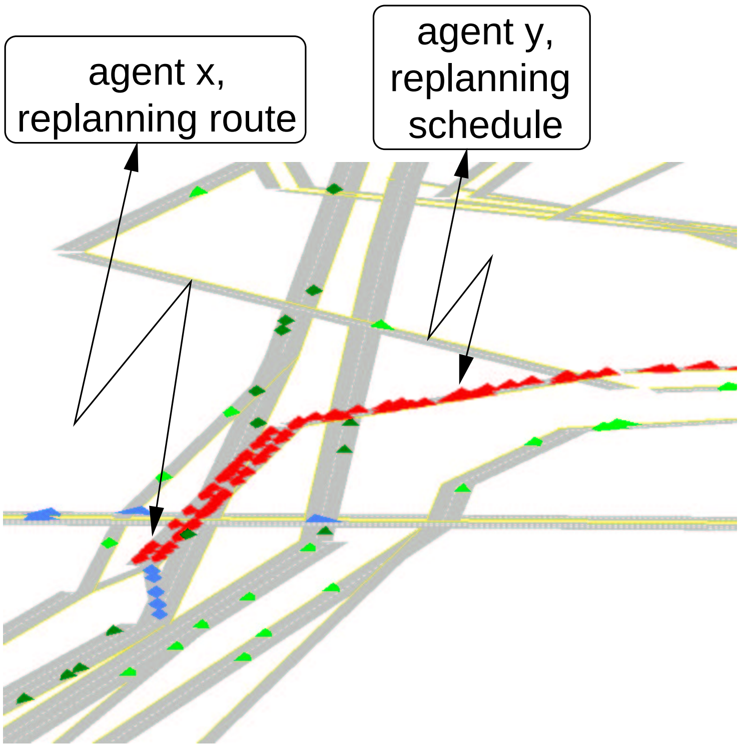

Figure 6: Virtual Reality Representation of Simulated Traffic in Portland/Oregon. Including visualization of within-day replanning.

Even with day-to-day replanning only, many problems remain. It was pointed out in this paper that the use of an agent data-base, i.e. the memorization of more than one strategy for each agent, solves some conceptual problems. However, even if one assumes that one is capable of generating a set of plausible strategies, the question becomes which of those to select. The standard logit approach of $p_i \propto e^{\beta U_i}$, where U_i is the utility of option i, has, as is well known, the so-called IID property ("independence from irrelevant alternatives"). IID essentially means that strategies should not be related. As an extreme example, assume that the agent-database contains three strategies for an agent, two of which are nearly the same. IID says that each strategy will be selected with a probability of 1/3, while it would be plausible that the nearly identical strategies are selected with a probability of 1/4 each, and the third, truly different strategy with a probability of 1/2. Alternatives to standard multinomial logit are C-logit or pathsize logit, which remove some of these problems (Bierlaire, 2002).

On 32-bit systems such as our Pentium systems, the is a limit of approximately 4 GByte of addressable memory. This is both relevant for files (and thus for file-based databases) and for implementations in computer memory. A possible workaround would be to distribute the agent database onto several computers. This is not a problem, since, within the agent database operations, the agents do not interact with one another, and can be processed completely independently.

This would imply the interaction of even more computers, some of them now being responsible for the computation of the strategic/tactical decisions. Another advantage of this approach, besides its improved speed, is that it should work well together with the within-trip replanning as explained above.

9.3 Other

It was mentioned above that there was a serious gridlock problem within the city of Zürich. This was attributed to generally too low network capacities. Unfortunately, this intuition is difficult to check. It is clear that, with the input data that was at our disposal, there was a mismatch between demand and network capacity. Also, the same method worked everywhere else in Switzerland. We can only think of three reasons: (i) there was a demand overestimation in the OD cells for Zürich; (ii) there was a capacity underestimation in the network data; (iii) our queue micro-simulation is overly sensitive to gridlock and this problem shows up only for large congested networks. Unfortunately, there is no other similarly large metropolitan region inside Switzerland; the metropolitan regions of Lugano, Geneva, and Basel extend across the border and therefore cannot be simulated realistically with our available demand data.

It should be noted that simulations with hard capacity and storage constraints are generically much more sensitive to capacity mismatches than static assignment. In static assignment, an overloaded link (with volume higher than capacity) will just be unattractive for the routing, but it will forward the requested steady state flow nevertheless. In a simulation with hard constraints, a queue will form upstream of such a bottleneck, and it will spill back into the rest of the system.

Our plan to solve this problem and to also advance towards more microscopic representation is to include a higher resolution network for the region around Zürich. This network will have considerably more links, possibly leading to a higher network capacity because of the addition of secondary capacity. That network should be a lot more reliable in terms of realism and thus eliminate one of the sources of errors. In addition, adding other choices into the model (mode, destination, activity pattern) should also dampen the adverse effects of demand-capacity mismatch.

10. Summary

The eventual goal of this work is a multi-agent traffic simulation of all of Switzerland. This work should demonstrate the feasibility of this technology on such a large scale, and it should be useful for research questions. A major challenge is to keep the computational performance fast enough so that such problem sizes can be computed over night, rather than over many months as is currently the case with similar packages, e.g. TRANSIMS (www.transims.net).

This paper presents intermediate results of this work. The intent of these intermediate results was to test if the technology, when reduced to a typical route assignment problem, would be useful for real world applications, and how it would compare to a VISUM assignment. The conclusion is that, even without specific tuning, the technology is somewhat better than the VISUM assignment. This means that the technology could be immediately be deployed for field testing. This is in fact intended with a field study around the introduction of the Zurich Glatttalbahn. This will go together with an extension of the technology towards activity-based demand generation. Further details of this will be reported in the years to come.

Acknowledgments

The Swiss regional planning authority (Bundesamt für Raumentwicklung, ARE) provided the original input data, which was then further improved by IVT at ETHZ. Nurhan Cetin provided the queue microsimulation. Milenko Vrtic and Kay Axhausen provided the VISUM results that were used for comparison. ETHZ and the Department of Computer Science made the Beowulf Cluster Xibalba including its Myrinet partition available to us; Marc Schmitt does a great job in maintaining the Linux environment of the cluster. We are deeply indebted to all of these, without whom this work would not be possible. This work was funded by ETHZ core funding and by the ETHZ project "Large scale multi-agent simulation of travel behavior and traffic flow".

References

Barrett, C. L., R. Jacob and M. V. Marathe (2000) Formal-language-constrained path problems, *SIAM J COMPUT*, **30** (3) 809–837.

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

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

Bierlaire, M. (2002) The network GEV model, in *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, CH. See www.strc.ch.

Bottom, J. (2000) Consistent anticipatory route guidance, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Bowman, J. L. (1998) The day activity schedule approach to travel demand analysis, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Bundesamt für Statistik und Dienst für Gesamtverkehrsfragen (1996) Verkehrsverhalten in der Schweiz 1994, Mikrozensus Verkehr 1994, Bern. See also http://www.statistik.admin.ch/news/archiv96/dp96036.htm.

Bundesamt für Strassen (2000) Automatische Strassenverkehrszählung 1999, Bern, Switzerland.

Cetin, N. and K. Nagel (2003a) A large-scale agent-based traffic microsimulation based on queue model, in *Swiss Transport Research Conference*, Monte Verita, Switzerland, March 2003. See www.strc.ch.

Cetin, N. and K. Nagel (2003b) Parallel queue model approach to traffic microsimulations, *Paper*, **03-4272**, Transportation Research Board Annual Meeting, Washington, D.C. Also see sim.inf.ethz.ch/papers.

Doherty, S. T. and K. W. Axhausen (1998) The developement of a unified modelling framework for the household activity-travel scheduling process, in *Verkehr und Mobilität*, no. 66 in Stadt Region Land, Institut für Stadtbauwesen, Technical University, Aachen, Germany.

Esser, J. and K. Nagel (2001) Iterative demand generation for transportation simulations, in D. Hensher and J. King (Eds.), *The Leading Edge of Travel Behavior Research*, 659–681, Pergamon.

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

Gloor, C. (2001) Modelling of autonomous agents in a realistic road network (in German), Diplomarbeit, Swiss Federal Institute of Technology ETH, Zürich, Switzerland.

Jacob, R. R., M. V. Marathe and K. Nagel (1999) A computational study of routing algorithms for realistic transportation networks, *ACM Journal of Experimental Algorithms*, **4** (1999es, Article No. 6).

Kaufman, D. E., K. E. Wunderlich and R. L. Smith (1991) An iterative routing/assignment method for anticipatory real-time route guidance, *Tech. Rep.*, **IVHS Technical Report 91-02**, University of Michigan Department of Industrial and Operations Engineering, Ann Arbor MI 48109, May 1991.

Nagel, K. (1994/95) High-speed microsimulations of traffic flow, Ph.D. thesis, University of Cologne. See www.inf.ethz.ch/˜nagel/papers.

Raney, B., N. Cetin, A. Völlmy, M. Vrtic, K. Axhausen and K. Nagel (2003) An agent-based microsimulation model of Swiss travel: First results, *Networks and Spatial Economics*, **3** (1) 23–41.

Raney, B. and K. Nagel (2002) Iterative route planning for modular transportation simulation, in *Proceedings of the Swiss Transport Research Conference*, Monte Verita, Switzerland, March 2002. See www.strc.ch.

Raney, B. and K. Nagel (2003) Truly agent-based strategy selection for transportation simulations, *Paper*, **03-4258**, Transportation Research Board Annual Meeting, Washington, D.C. Also see sim.inf.ethz.ch/papers.

Raney, B. and K. Nagel (submitted) Iterative route planning for large-scale modular transportation simulations, *Future Generation Computer Systems*. See sim.inf.ethz.ch/papers.

Rickert, M. (1998) Traffic simulation on distributed memory computers, Ph.D. thesis, University of Cologne, Germany. See www.zpr.uni-koeln.de/˜mr/dissertation.

Schwerdtfeger, T. (1987) Makroskopisches Simulationsmodell für Schnellstraßennetze mit Berücksichtigung von Einzelfahrzeugen (DYNEMO), Ph.D. thesis, University of Karsruhe, Germany.

Sheffi, Y. (1985) *Urban transportation networks: Equilibrium analysis with mathematical programming methods*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Stein, D. and others (Eds.) (since 1988) *Lectures in the sciences of complexity*, Santa Fe Institute in the sciences of complexity, Addison-Wesley.

Vaughn, K., P. Speckman and E. Pas (1997) Generating household activity-travel patterns (HATPs) for synthetic populations.

Vrtic, M. and K. Axhausen (2002) Experiment mit einem dynamischen umlegungsverfahren, *Strassenverkehrstechnik*. Also Arbeitsberichte Verkehrs- und Raumplanung No. 138, see www.ivt.baug.ethz.ch.

Vrtic, M., R. Koblo and M. Vödisch (1999) Entwicklung bimodales Personenverkehrsmodell als Grundlage für Bahn2000, 2. Etappe, Auftrag 1, ***Report to the Swiss National Railway and to the Dienst für Gesamtverkehrsfragen***, Prognos AG, Basel. See www.ivt.baug.ethz.ch/vrp/ab115.pdf for a related report.