Networks and Spatial Economics, 3: (2003) 23–41

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

BRYAN RANEY, NURHAN CETIN AND ANDREAS VÖLLMY
Department of Computer Science, ETH Zürich, CH-8092 Zürich, Switzerland
email: raney@inf.ethz.ch; cetin@inf.ethz.ch; res@vis.ethz.ch

MILENKO VRTIC AND KAY AXHAUSEN
Department of Civil, Environmental and Geomatics, ETH Zürich, CH-8092 Zürich, Switzerland
email: vrtic@ivt.baug.ethz.ch; axhausen@ivt.baug.ethz.ch

KAI NAGEL
Department of Computer Science, ETH Zürich, CH-8092 Zürich, Switzerland
email: nagel@inf.ethz.ch

Abstract

In a multi-agent transportation simulation, each traveler is represented individually. Such a simulation consists of at least the following modules: (i) Activity generation. (ii) Modal and route choice. (iii) The traffic simulation itself. (iv) Learning and feedback. In order to find solutions which are consistent between the modules, a relaxation technique is used. This technique has similarities to day-to-day human learning.

Using advanced computational methods, in particular parallel computing, it is now possible to run such a system for large metropolitan areas with 10 million inhabitants or more. This paper reports on such a simulation system for all of Switzerland. Our focus is on a computationally efficient implementation of the agent-based representation, which means that in fact each agent is represented with an individual set of plans as explained above. A database is used to store the agents' strategies, which are loaded into the simulation modules as required; the modules then feed back individual performance measures into the database. This approach allows that additional modules can be coupled easily, and without degrading computational performance.

The set-up was tested for Swiss morning peak traffic. Hourly demand matrices were taken from work with the VISUM assignment package and converted to our needs. Routes were assigned via feedback learning using the agent data base. In other words, the current implementation uses a car-only versions of the modules (ii), (iii), and (iv). Resulting flow volumes are compared to the VISUM assignment results, and to field data.

Keywords: multi-agent simulation, parallel computing, dynamic traffic assignment

1. Introduction

Human transportation has physical, engineering, and socio-economic aspects. This last aspect means that any simulation of human transportation systems has to include elements of adaptation, learning, and individual planning. In terms of computerization, these aspects can be better described by discrete rules which are applied to individual entities than by continuous equations which are applied to aggregated fields. In consequence a rule-based multi-agent simulation is a promising method for transportation simulations (and for socio-economic simulations in general).

By a "multi-agent" simulation we mean a microscopic simulation that models the behavior of each traveler, or *agent*, within the transportation system as an individual, rather than aggregating their behavior in some way. These agents are intelligent, which means that they have strategic, long-term goals. They also have internal representations of the world around them which they use to reach these goals. Such a simulation differs significantly from a microscopic simulation of, say, molecular dynamics, because unlike molecules, two "traveler" particles (agents) in identical situations within a transportation simulation will in general make different decisions.

Such rule-based multi-agent simulations run well on current workstations and they can be distributed on parallel computers of the type "networks of coupled workstations." Since these simulations do not run efficiently on traditional supercomputers (e.g. Cray), the jump in computational capability over the last decade has had a greater impact on the performance of multi-agent simulations than for, say, computational fluid-dynamics, which also worked well on traditional supercomputers. In practical terms, this means that we are now able to run microscopic simulations of large metropolitan regions with more than 10 million travelers. The simulations are even fast enough to run them many times in sequence, which is necessary to emulate the day-to-day dynamics of human learning, for example in reaction to congestion.

In order to demonstrate this capability and also in order to gain practical experience with such a simulation system, we are currently implementing a 24-hour microscopic transportation simulation of all of Switzerland. Switzerland has 7.2 million inhabitants. Assuming the national average of 3.6 trips/day,[1] an average 40% share of car, average 1.6 passengers/car, and a share of 30% of car trips under 3 km one obtains about 4.5 mio interzonal trips within the country. The goal of our study is twofold:

- Investigate the computational challenges and how they can be overcome.
- Investigate what is necessary to make a simulation system realistic enough to be useful for such a scenario, and how difficult this is.

This paper gives a report on the current status. Section 2 describes the simulation modules and how they were used for the purposes of this study. Section 3 describes the input data, i.e. the underlying network and the demand generation. This is followed by Section 4, which describes our main results, including a comparison to field data and to a VISUM assignment result. Section 5 describes issues related to computational performance of the parallel micro-simulation. The paper ends with a discussion and a summary.

2. Simulation modules

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

- **Population generation.** Demographic data is disaggregated so that one obtains individual households and individual household members, with certain characteristics, such as a

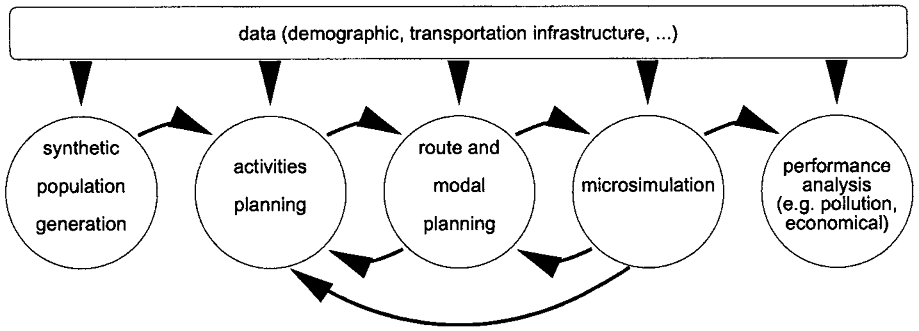
data (demographic, transportation infrastructure, ...)
synthetic population generation
activities planning
route and modal planning
microsimulation
performance analysis (e.g. pollution, economical)

Figure 1. TRANSIMS modules.

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 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 future.
- **Modal and route choice.** For each individual, the modes are selected and routes are generated that connect activities at different locations (see Section 2.1). The routing should be dynamic in order to react adequately to time-dependent congestion effects.
- **Traffic micro-simulation.** Up to here, all individuals have made *plans* about their behavior. The traffic micro-simulation executes all those plans simultaneously (see Section 2.2). 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 (Section 2.3). 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 traffic micro-simulation, adapt the plans, run the traffic micro-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.

Since this paper is a status report, 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 micro-simulation, and the feedback. These modules will be described in more detail in the preceeding sections. It should be noted that in particular the feedback system is unique in that it explicitely 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 Section 3.

2.1. Routing

Travelers/vehicles need to compute the sequence of links that they are taking through the network. A typical way to obtain such paths is to use a shortest path Dijkstra algorithm. This algorithm uses as input the individual 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 fed back from the micro-simulation in 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.

2.2. Micro-simulation

Our main micro-simulation is the queue simulation (Gawron, 1998; Cetin and Nagel, 2003). The intent with this simulation is to keep travelers/vehicles microscopic and to have queue spillback, but apart from this to keep the simulation as simple as possible. This is similar in spirit to traffic simulations based on the smooth particle hydrodynamics approach, such as DYNEMO (Schwerdtfeger, 1987), DYNAMIT (its.mit.edu), or DYNASMART (www.dynasmart.com).

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.

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 micro-simulations, which need, for example, the number of lanes including pocket and weaving lanes, turn connectivities across intersections, or signal schedules.

2.3. Feedback

As mentioned above, plans (such as routes) and congestion need to be made consistent. This is achieved via a relaxation technique (Kaufman et al., 1991; Nagel, 1994/95; Bottom, 2000):

1. Initially, the system generates a set of routes based on free speed travel times.
2. The new routes are stored in a database, called the "agent database" (Raney and Nagel, 2002, 2003), so that the travelers ("agents") may later associate the performance of the route to it, and may choose routes based on performance.
3. The traffic simulation is run with these routes.
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, along with the route that was used.
5. 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 are then stored in the agent database.
6. Travelers who did not request new routes choose a previously tried route from the agent database by comparing performance values for the different 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. β was set heuristically to 1/(360 sec) to obtain a fraction of about 10% non-optimal users.
7. This cycle (i.e. steps (3) through (6)) is run for 50 times; earlier investigations have shown that this is more than enough to reach relaxation (Rickert, 1998).

The result is similar to a Stochastic User Equilibrium (SUE), but it is not the same. The main difference is that in an SUE, the agents use a logit model with an externally specified noise parameter to select between options of different performance, while in our system additional noise comes from the simulation, i.e. from the fluctuations between iterations.

In fact, relaxation itself, in Item [7], is not well defined in a mathematical sense. The intended meaning is that, in the average, a relaxed system should not show any further development or drift. Since the system is stochastic at many levels, the only way in which this could be mathematically achieved is in terms of a steady-state density. If the system were Markovian, then the convergence to a steady-state density would immediately follow; the system is however not Markovian because it can potentially enlarge its phase space at each iteration by finding new routes. Alternatively, one could include all possible routes into the phase space definition. This system would be Markovian, but 50 iterations would be by far too few to explore the phase space, let alone generating a steady state density. Similarly, this enlarged system is ergodic in the sense of Cantarella and Cascetta (1995), but that notion is not useful for the relatively small number of iterations that we use. More precisely: Even systems that are formally ergodic can remain in limited regions of the phase space for very long duration, certainly for much longer than for 50 iterations (e.g. Palmer, 1989).

A related issue is the selection of the replanning fraction. A replanning fraction of 10%, as in Item [5], is a heuristic number that works well in practice. The important limiting considerations are: (i) Adaptation in this system works by travelers finding improved solutions for the current situation. These new solutions will only then work better than previous ones when the system behaves similar from one iteration to the next. This implies that the fraction of the population changing its behavior should not be too large. In fact, for some simpler and deterministic systems one can show that an infinitesimal best reply dynamics leads the system to a Nash Equilibrium (Hofbauer and Sigmund, 1998). One could expect that the situation for our system here is similar, although no formal proof is available. (ii) On the other hand, when the replanning fraction becomes too small, then the relaxation process becomes too slow. For example, with a replanning fraction of 1%, one will need at least 100 iterations until each traveler has obtained a new route once, and that will probably not be enough.

Other methods, in particular the method of successive averages (MSA, see, e.g., Sheffi (1985)) could be tried, although MSA has a reputation, justified from a theoretical perspective, to display rather slow convergence. In addition, some translation of MSA to a stochastic process would be necessary. For example, a traveler's potentially mixed strategy should be an *average* best response against the traffic pattern, and this is not necessarily the same as a best response against the average traffic pattern. It is not immediately clear how to achieve an average best response without a lot of averaging over many iterations (many more than 50).

In practice, however, Rickert (1998) has looked at the sum of all travel times as an indicator for relaxation, and has found that a system which was similar to ours did not display any further drift after about 25 iterations. This observation was confirmed by visual inspection of the traffic patterns. Bottom (2000) has done a much more exhaustive investigation into the same topic, with a similar result.

Note that all the above arguments imply that routes are fixed during the traffic simulation and can only be changed between iterations. Work is under way to improve this situation, i.e. to allow online re-planning (Gloor, 2001). Further investigation of relaxation and learning issues is planned.

3. Input data and scenarios

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

3.1. 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; Section 7 discusses our plans of how to improve the situation.

After our modifications, the network has the fairly typical number of 10 564 nodes and 28 622 links. Also fairly typical, the major attributes on these links are type, length, speed, and capacity. As pointed out above, this is enough information for the queue simulation.

3.2. The "Gotthard" scenario

In order to test our set-up, we generated a set of 50 000 trips going to the same destination. Having all trips going to the same destination allows us to check the plausibility of the feedback since all traffic jams on all used routes to the destination should dissolve in parallel. In this scenario, we simulate the traffic resulting from 50 000 vehicles which start between 6 am and 7 am all over Switzerland and which all go to Lugano, which is in the Ticino, the Italian-speaking part of Switzerland south of the Alps. In order for the vehicles to get there, most of them have to cross the Alps. There are however not many ways to do this, resulting in traffic jams, most notably in the corridor leading towards the Gotthard pass. This scenario has some resemblance with real-world vacation traffic in Switzerland.

3.3. The "Switzerland" scenario

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 matrixes using a three step heuristic. 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 (http://www.statistik.admin.ch/news/archiv96/dp96036.htm). The resulting 24 initial matrices are then corrected (calibrated) against available hourly counts using the OD-matrix estimation module of VISUM (PTV, 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 24 h 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 Section 7 for some discussion of this. A more detailed verification of these results was not possible so far, but is planned.

These hourly matrices are then disaggretated 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 am and 9 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.

4. Results

Figure 2 shows an example of how the feedback mechanism works in the Gotthard scenario. The figure shows two "snapshots" of the vehicle locations within the queue-based

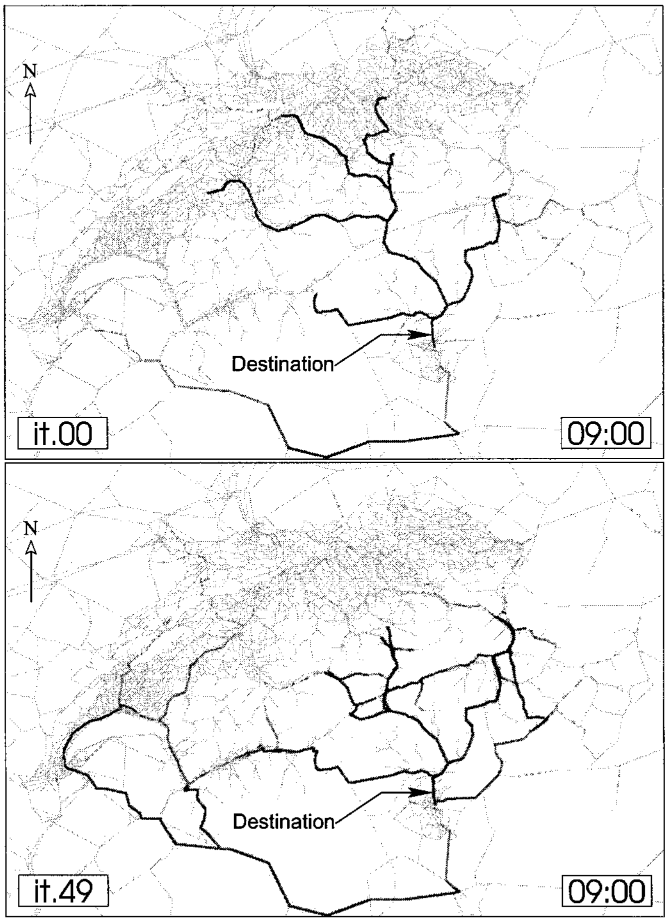

Figure 2. Example of relaxation due to feedback. TOP: Iteration 0 at 9:00—all travelers assume the network is empty. BOTTOM: Iteration 49 at 9:00—travelers take more varied routes to try to avoid one another. Red (dark gray in b/w version) indicates jams, green (dark gray in b/w version) indicates free-flowing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

micro-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 Section 2.3.

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 netwrok caused by one another.

After 50 iterations between the route selection and the micro-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.

Figure 3 shows a result of the Switzerland scenario during morning rush-hour. This figure is after 50 iterations of the queue micro-simulation, using the agent database. We used as input the origin-destination matrices described in Section 3.3, 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 Zurich (near the top). This is barely visible in figure 3, but can be verified by zooming in (possible with the electronic version of this paper, on the TRB CD-ROM or at sim.inf.ethz.ch/papers/ch). As of now, it is unclear if this is a consequence of a higher imbalance between supply and

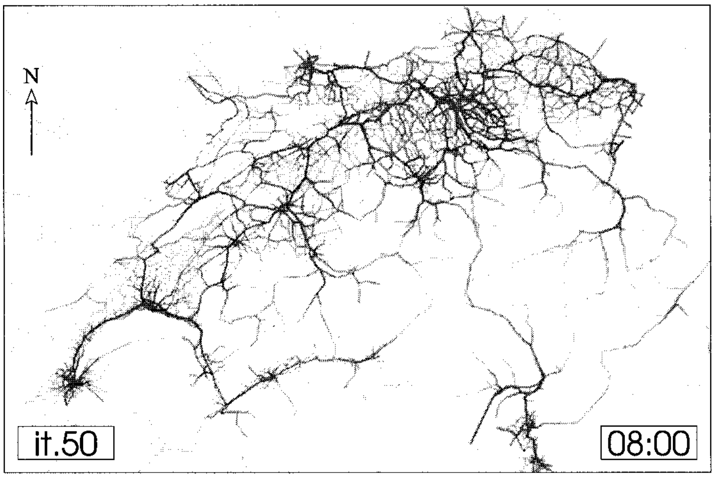

Figure 3. Snapshot of Switzerland at 8:00 am. From the queue micro-simulation, iteration 50.

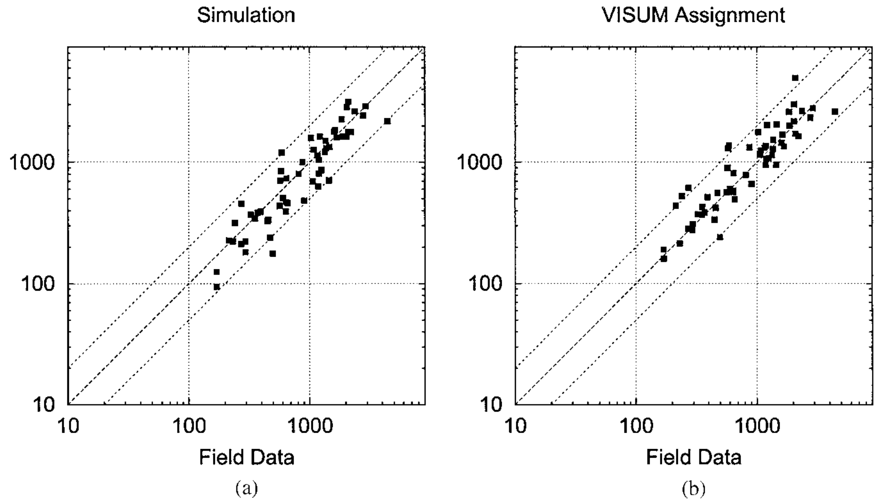

Figure 4. (a) Simulation vs. field data for the 50th iteration. The x-axis shows the hourly counts between 7 am and 8 am from the field data; the y-axis shows throughput on the corresponding link from the simulation. (b) VISUM assignment vs. field data. The x-axis is the same as (a); the y-axis shows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

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 figure 3 and field data taken at counting stations throughout Switzerland (see Section 3.3 and Bundesamt für Strassen (2000)). The dotted lines 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 (figure 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 $\langle q_{\text{sim}}, - q_{\text{field}} \rangle$, mean absolute error is $\langle |q_{\text{sim}} - q_{\text{field}}| \rangle$, mean relative bias is $\langle (q_{\text{sim}} - q_{\text{field}})/q_{\text{field}} \rangle$, mean relative error is $\langle |q_{\text{sim}} - q_{\text{field}}|/q_{\text{field}} \rangle$, where $\langle \cdot \rangle$ means that the values are averaged over all links where field results are available.

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%

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

5. Computational issues

A metropolitan region can consist of 10 million or more inhabitants, the simulation of whom causes considerable demands on computational performance. This demand on computation is made worse by the repeated execution of the relaxation iterations. And in contrast to simulations in the natural sciences, traffic particles (= travelers, vehicles) have internal intelligence. This internal intelligence translates into rule-based code, which does not vectorize and therefore does not run efficiently on traditional vector supercomputers, such as the Cray series. It does however run well on modern workstation architectures, which makes traffic simulations ideally suited for clusters of PCs, also called Beowulf clusters. One uses domain decomposition, that is, each CPU obtains a patch of the geographical region. Information and vehicles are exchanged between the patches via message passing, for example using MPI (Message Passing Interface, MPI (www-unix.mcs.anl.gov/mpi/mpich)).

Two important numbers to judge the performance of a parallel simulation are speed-up and real time ratio. They are defined as follows:

- The **speed-up** gives the ratio between the computational speed on p CPUs to the computational speed on one CPU:

$$S(p) = \frac{T(1)}{T(p)},$$

  where $T(p)$ is the time that the computation needs on p CPUs.
- The **real time ratio** (RTR) gives the ratio between the speed of time in the simulation and the speed of time in reality. For example, an RTR of 10 means that 10 hours of traffic can be simulated during 1 hour of computer time.

The limiting factor for tightly coupled simulations, as is a parallel traffic simulation, is in practice the latency of the communication hardware, which is the time that is needed to initiate a message. For example, 100 Mbit Ethernet has a latency of about 0.5 msec. Since each domain has in the average six neighbors, and typically two messages are sent and received per time step, about 6 msec per time step are spent on communication. This sets an upper limit on parallel simulation speed of 1 sec/6 msec $\approx$ 167 time steps per second for this type of hardware. If the simulation has a time step of one second, this translates into an RTR of 167. Unfortunately, latency of standard communication hardware has not significantly improved over the last decade, so that this limitation will not go away by itself. It is however possible to use Myrinet (www.myri.com), a communications technology specifically developed for cluster computing.

Figure 5 shows measured and predicted computing speeds as a function of the number of CPUs for the queue micro-simulation and the Switzerland scenario. Both figures refer to the simulation of the morning peak, as explained earlier in this paper. The top figure shows the real time ratio (RTR); the bottom figure shows the speed-up. As one can see, the plots are related by a vertical shift of the data, which means a multiplication with a constant

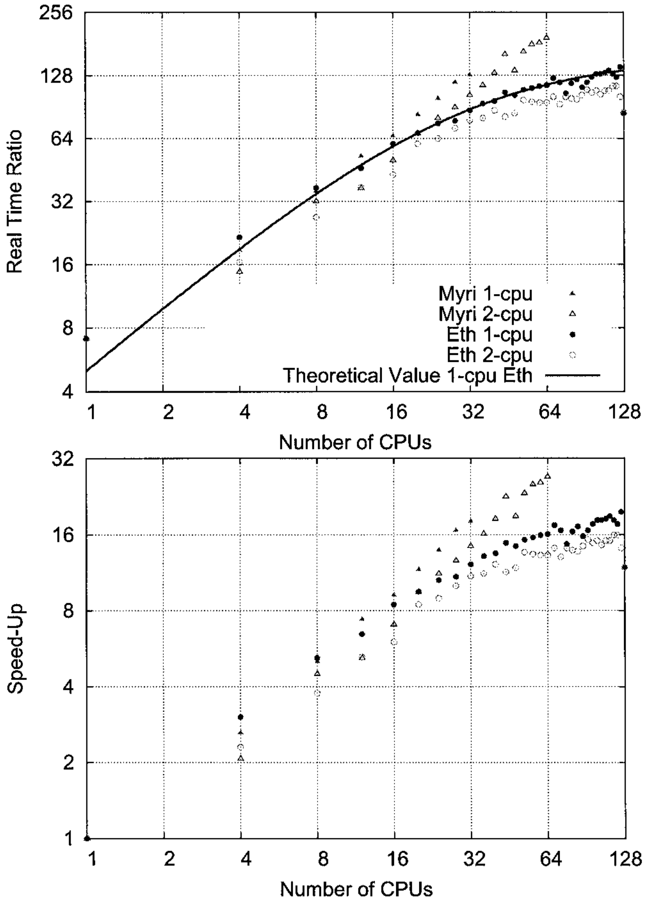

Figure 5. Real time ratio (top) and speed-up (bottom) for the 6–9 scenario on single and dual CPU machines, using Ethernet or Myrinet. The x-axis refers to the number of CPUs. The solid lines give the predictions according to the computing time T_{cmp} and the latency term for Ethernet. As one can see, real time ratio and speed-up are related by a simple vertical shift in the logarithmic plot.

multiplier because of the logarithmic scale. That multiplier is the RTR of the simulation on a single CPU; in our case, the RTR on a single CPU is about eight. The differences between RTR and speed-up would become important if one changed the scenario size: In the RTR plot, the graph would be shifted to the left or right for smaller or larger scenarios, respectively, meaning that the maximally reachable RTR would not change, but would be shifted to a different number of CPUs. In the speed-up plot, the graph would be shifted down or up, respectively, meaning that the maximally reachable speed-up depends on the size of the scenario. That is, for *both* plots the results depend on the scenario size; it is impossible to make parallel performance predictions without knowledge about the scenario size.

High computing speeds matter: For real time applications, one wants the simulation to be considerably faster than reality so that the prediction is finished much before reality catches up. For transportation planning, about 50 iterations between simulation and plans generation are necessary, meaning $50 \times 24 = 1200$ hours of simulated traffic for a 24-hour scenario. With our real time ratio of 200, the computing time for this would still be 6 hours for the micro-simulation alone.

Besides the micro-simulation, also the feedback mechanism consumes computing time; including re-routing and agent database operations, it currently takes roughly 45 min per iteration for the morning peak Switzerland scenario (Raney and Nagel, 2003). This is clearly the bottleneck of the current approach; better implementations are under investigation.

In summary, an iteration using our current implementation takes less than one hour. Running fifty iterations thus takes about two days.

6. Experiences with TRANSIMS (Version 1.0)

Before programming our own modules as explained above, we attempted to use TRANSIMS. The TRANSIMS version that we used is numbered 1.0 and was made available in fall 1999. Our experiences were as follows:

Porting TRANSIMS to our own computational environment was straightforward. Using our own input files was relatively straightforward, but hindered by the fact that errors in input files—such as a forgotten tab—caused the simulation to crash without a meaningful error message. In consequence, one had to find the cause of the problem via manual trial and error.

Computational speed of the microsimulation without tuning was ten times faster than real time for our Swiss network with 28622 links; with tuning it was about 65 times faster than real time. Both values refer to a Beowulf clusters with 32 Pentium CPUs with 800 MHz, and 100 Mbit Ethernet. The latter performance value is about half the theoretical limit, which is given by Ethernet latency (see above).

A major problem was (and is) the availability and conversion of digitally available input data to TRANSIMS needs. As is typical, our input files come from static assignment, and thus contain as link attributes length, free speed, and capacity. The number of lanes can be inferred from the street category and capacity, but information such as intersection prioritization, signal phases, or lane connectivity, were missing. A typical problem is of the type that two one-lane streets, connecting into a two-lane street, will both connect into the same lane, leading to much reduced capacity and thus spurious bottlenecks when compared to reality or to static assignment. According to recent information (Lamba, at www.transmins.net, personal communication), such conversion tools exist for newer versions of TRANSIMS. Also, PTV (www.ptv.de) reports similar conversion tools from VISUM to VISSIM.

Using routing and feedback was essentially straightforward, except for the fact that the results of the Gotthard scenario never were plausible: Contrary to expectation, the different queues leading to the single destination never came close to equilibration (figure 6). It was

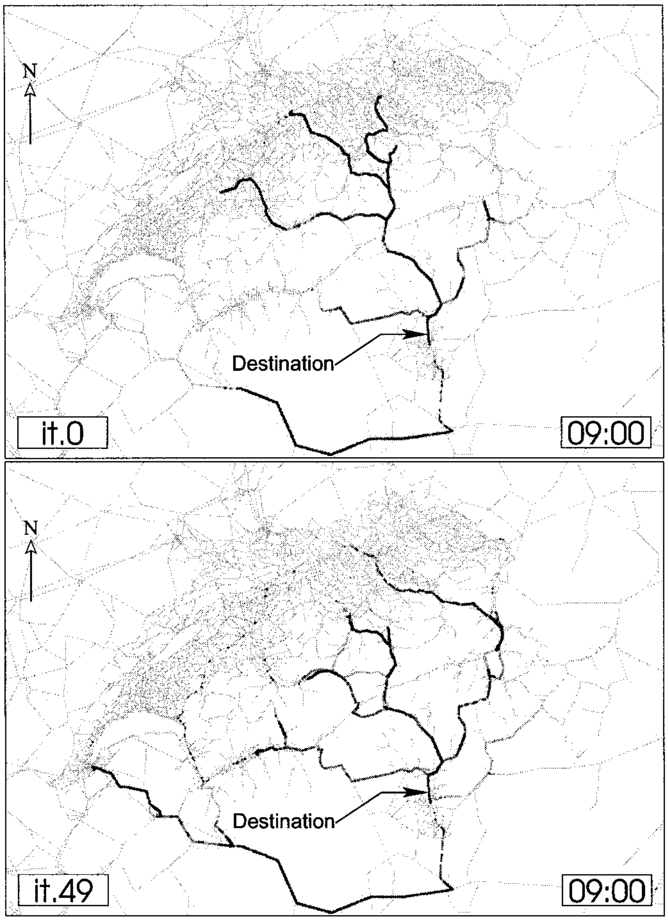

Figure 6. TRANSIMS results for the Gotthard scenario. TOP: Initial iteration. BOTTOM: After 50 iterations. This should be compared to figure 2. The visible differences between the TRANSIMS simulation and our queue simulation in the initial simulation based on the same plans are small (TOP in both figures), indicating that the micro-simulations generate similar traffic patterns. However, the bottom figures are rather different; traffic in figure 2 spreads out much more across the network. Further analysis of the pattern shows that figure 2 contains the better pattern in the sense that travel times between different routes are much more equilibrated. As described in the text, the reason why TRANSIMS (Version 1.0) fails at this scenario is because of a bug in how the router reads link travel times. The different shades of green (color version) or gray are due to the different internal representation of driving dynamics between the TRANSIMS micro-simulation and the queue simulation, and are not important at this level.

finally discovered that there was a bug in the way link travel time feedback was handled: the link travel time reporting allocated the times to the wrong links. More technically, indices were shifted by one in the process, and so travel times for the nth link in the file were assigned to the $n + 1$st link in the router. TRANSIMS version 3.0 now available (www.transims.net) has not been evaluated in the context of this study.

7. Discussion and future plans

7.1. Modules

This paper describes one possible implementation of a large-scale agent-based simulation package for regional planning. As was repeatedly pointed out, the approach is modular and extensible. In order to test the modularity, replacing one or more modules by alternative ones is desirable. In the following, this is discussed on a module-by-module basis.

Traffic micro-simulation. The queue simulation has its limitations, for example with respect to complicated intersections, inhomogeneous vehicle fleets, queue dissolution, interaction between different modes of transportation, etc. These limitations will be difficult or impossible to remove within the method of the queue simulation approach. Therefore it seems desirable to move beyond the queue simulation to a more realistic traffic simulation. Besides being more realistic, this simulation should fulfil the following criteria in order to be consistent with our approach: It should be able to process travelers with individual plans; and it should be computationally fast. There are currently few traffic simulations which fulfill these criteria simultaneously. The TRANSIMS microsimulation is one of them. As discussed above, with the emergence of useful network conversion tools, this may become a viable option. Note that including the micro-simulation into our set-up would still be different from using the full TRANSIMS suite.

Router. Our current router computes car-only fastest paths, without regard for alternative cost functions (such as monetary cost, familiarity, scenic beauty, etc.), and without regard for alternative modes. Again, an option would be to use the multi-modal TRANSIMS router as a single module within our set-up. This will, as discussed above, depend on functionality.

Yet, having the fastest path, even if multi-modal, does not solve all problems. In practice, people often do not use the fastest path, or there are stochastic influences, or the path depends on which part of a network they know (mental map). Maybe somewhat unexpectedly, it is rather difficult to construct non-optimal solutions to the routing problem (e.g. Park and Rilett, 1997).

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.

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 addressed via a combination of database performance tuning and consolidating the current script-based approach into one program. 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 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). 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.

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 to generate 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 irrelvant 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 multinominal logit are C-logit or pathsize logit, which remove some of these problems (Bierlaire, 2002).

7.2. *Other*

It was mentioned above that there was a serious gridlock problem within the city of Zurich. 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 Zurich; (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 Zurich. 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.

Finally, it is necessary to point out the necessity of regression testing and "trusted components." The bug in the TRANSMIS feedback setup was found after rather a lot of manual work, and it was only found because of the specific testing set-up. In many "normal" scenarios, such as our 6–9 scenario, there is a good chance that the problem would have gone unnoticed for a much longer time. The major concern is however that a problem may get fixed, but then, with further changes, some new problem may appear. It is therefore desirable, albeit awkward, to consider systematic regression testing in the community of large scale microscopic simulation. Regression testing means systematic test suites which are run every time the software is changed, and which ensure that previously working functionality is not degraded by later changes in the code. Trusted components means that possibly certain pieces of a software, maybe after a formal proof of their correctness, should be completely removed from further changes—all improvements then need to be done via transparent object-oriented interfaces. It is unclear if one can reconcile such an approach with the desire for flexbility in a research environment.

8. Summary

In terms of travelers and trips, a simulation of all of Switzerland, with more than 10 million trips, is comparable to a simulation of a large metropolitan area, such as London or Los Angeles. It is also comparable in size to a molecular dynamics simulation, except that travelers have considerably more "internal intelligence" than molecules, leading to complicated rule-based instead of relatively simple equation-based code. Such multi-agent simulations do not run well on traditional vectorizing supercomputers (e.g. Cray) but run well on distributed workstations, meaning that the computing capabilities for such simulations have virtually exploded over the last decade.

This paper describes the status of ongoing work of an implementation of all of Switzerland in such a simulation. The whole simulation package consists of many modules, including the micro-simulation itself, the route planner, and the feedback supervisor which models day-to-day learning. A single destination scenario is used to verify the plausibility of the replanning set-up. A result of a morning peak-hour simulation of all of Switzerland is shown, including comparisons to field data from automatic counting stations. These results are shown to be better than VISUM assignment model results when compared to the same field data. This is in fact somewhat surprising, since the OD matrices were adapted by a VISUM module to make the assignment result match the counts data as well as possible.

However, the really big advantage of the multi-agent approach is that it is theoretically justified even under dynamic and congested conditions, and for that reason is extensible even under those conditions. This makes it possible to integrate aspects such as dynamic activity-based demand generation into the framework. Our expectation is that this new technology will allow to introduce many important aspects, such as time-dependent elastic demand or analysis of multi-functional land-use patterns, into the methodology while maintaining or even improving the level of realism.

Acknowledgments

We thank the Swiss regional planning authority (Bundesamt für Raumentwicklung) for the original input data. We also thank ETHZ and the Department of Computer Science for making 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. This work was funded by ETHZ core funding and by the ETHZ project "Large scale multi-agent simulation of travel behavior and traffic flow."

Note

[1] See http://www.are.admin.ch/imperia/md/content/are/are/medienmitteilungen/2001/1.pdf

References

Barrett, C.L., R. Jacob, and M.V. Marathe. (2000). "Formal-Language-Constrained Path Problems." *SLAM 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.

Bierlaire, M. (2002). "The Network GEV Model." In *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, CH. See www.strc.ch.

Bottom, J.A. (2000). "Consistent Anticipatory Route Guidance." PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Bowman, J.L. (1998). "The Day Activity Schedule Approach to Travel Demand Analysis." PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA.

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

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

Cantarella, C. and E. Cascetta. (1995). "Dynamic Process and Equilibrium in Transportation Network: Towards a Unifying Theory." *Transportation Science A* 25(4), 305–329.

Cetin, N. and K. Nagel. (2003). "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.

DYNAMIT. Massachusetts Institute of Technology, Cambridge, Massachusetts. See its.mit.edu. Also see dynamictrafficassignment.org.

DYNASMART. See www.dynasmart.com. Also see dynamictrafficassignment.org.

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*, Pergamon, pp. 659–681.

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, Chr. (2001). “Modelling of Autonomous Agents in a Realistic Road Network (in German).” Diplomarbeit, Swiss Federal Institute of Technology ETH, Zürich, Switzerland.

Hofbauer, J. and K. Sigmund. (1998). *Evolutionary Games and Replicator Dynamics.*” Cambridge University Press.

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, David E., Karl E. Wunderlich, and Robert L. Smith. (1991). “An Iterative Routing/Assignment Method for Anticipatory Real-Time Route Guidance.” Technical Report IVHS Technical Report 91-02, University of Michigan Department of Industrial and Operations Engineering, Ann Arbor MI 48109.

MPI. MPI: Message Passing Interface. See www-unix.mcs.anl.gov/mpi/mpich.

Nagel, K. (1994/95). “High-Speed Microsimulations of Traffic Flow.” PhD Thesis, University of Cologne. See www.inf.ethz.ch/˜nagel/papers.

Palmer, R. (1989). “Broken Ergodicity.” In D.L. Stein (Ed.), *Lectures in the Sciences of Complexity*, volume I of *Santa Fe Institute Studies in the Sciences of Complexity*, Addison-Wesley, pp. 275–300.

Park, D. and L.R. Rilett. (1997). “Identifying Multiple and Reasonable Paths in Transportation Networks: A Heuristic Approach.” *Transportation Research Records* 1607, 31–37.

PTV—Planung Transport Verkehr. See www.ptv.de.

Raney, B. and K. Nagel. (2002). “Iterative Route Planning for Modular Transportation Simulation.” In *Proceedings of the Swiss Transport Research Conference*, Monte Verita, Switzerland. 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., 2003. Also see sim.inf.ethz.ch/papers.

Rickert, M. (1998). “Traffic Simulation on Distributed Memory Computers.” PhD 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).” PhD Thesis, University of Karsruhe, Germany.

Sheffi, Y. (1985). *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods.* Prentice-Hall, Englewood Cliffs, NJ, USA.

Vaughn, K.M., P. Speckman, and E.I. Pas. (1997). “Generating Household Activity-Travel Patterns (HATPs) for Synthetic Populations.” TRANSIMS internal report.

Vrtic, M., R. Koblo, and M. Vödisch. (1999). Entwicklung Bimodales Personenverkehrsmodell als Grundlage fr 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.