Large Scale Multi-Agent Transportation Simulations

Bryan Raney, Nurhan Cetin, Andreas Völlmy, and Kai Nagel
Dept. of Computer Science, ETH Zürich
CH-8092 Zürich, Switzerland
{raney@inf; cetin@inf; res@student; nagel@inf}.ethz.ch

Submitted to the 42^{nd} ERSA Congress, Dortmund, 2002

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. For each traveler in the simulation, a complete 24-hour day-plan is generated, with each major activity (sleep, eat, work, shop, drink beer), their times, and their locations. (ii) Modal and route choice. For each traveler in the simulation, the mode of transportation and the actual routes are computed. (iii) The Traffic simulation itself. In this module, the travelers are moved through the system, via the transportation modes they have chosen. (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 and can also be interpreted that way. – Besides, one needs input data, such data of 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.

Using advanced computational methods, in particular parallel computing, it is now possible to do this for large metropolitan areas with 10 million inhabitants or more. We are currently working on such a simulation of all of Switzerland. Our focus is on a computationally efficient implementation of the agent-based representation, which means that we in fact represent each agent with an individual set of plans as explained above. We use a database to store the agent's strategies, then load them into the simulation modules as required, and feed back individual performance measures into the database. This approach allows that additional modules can be coupled easily, and without destroying computational performance.

1 Introduction

Human transportation has physical, engineering, and socio-economic aspects. This last aspect means that any simulation of human transportation systems will include elements of adaptation, learning, and individual planning. In terms of computerization, these aspects can be much better described by rules which are applied to individual entities than by 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. Adding the term “rule-based” indicates that the behavior of the agents is determined by sets of rules instead of equations. Thus, a rule-based multi-agent simulation of a transportation system will apply to each agent individually. 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. These 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 3 to 3.5 trips per person per day, this will result in about 20–25 million trips. This number includes pedestrian trips (like walking to lunch), trips by public transit, freight traffic, etc. The number of car trips on a typical weekday in Switzerland is currently about 5 million (see Vrtic (2001) for where the data comes from). 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. Besides "normal" demand, we also describe one where 50 000 travelers travel from random starting points within Switzerland to the Ticino, which is the southern part of Switzerland. We use this second scenario as a plausibility test for routing and feedback. This is followed by Sect. 4, which describes some results and Sect. 5, which 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 (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 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.,

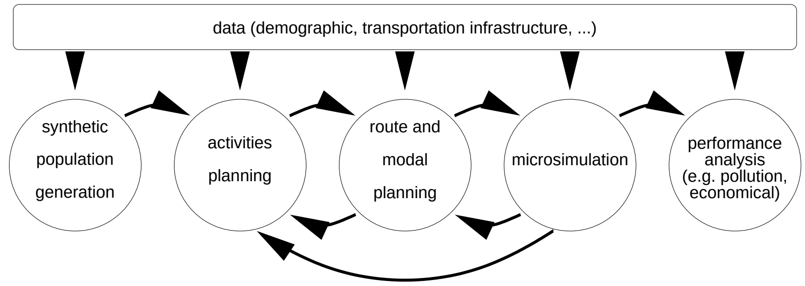

Figure 1: TRANSIMS modules

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 Sec. 2.1).

- **Traffic micro-simulation**. Up to here, all individuals have made *plans* about their behavior. The traffic micro-simulation executes all those plans simultaneously (see Sec. 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 (Sec. 2.3). For example, plans depend on congestion, but congestion depends on plans. A widely accepted method to resolve this is systematic relaxation – 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 is that it is now feasible to make all modules completely microscopic, i.e. each traveler is individually represented in all modules.

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., 1997); 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, in preparation). 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 (Mahmassani et al., 1995).

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-simulation, 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), 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 $1/(360 \text{ sec})$ 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).

Note that this implies 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).

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). It has since been modified by Vrtic at the IVT, and again by us. The network has the fairly typical number of 10 572 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 6am and 7am 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

For a realistic simulation of all of Switzerland, the starting point for demand generation is a 24-hour origin-destination matrix, again from the Swiss regional planning authority (Bundesamt für Raumentwicklung). For this matrix, the region is divided into 3066 zones. Each matrix entry describes the number of trips from one zone to another during a typical 24-hour workday; trips within zones are not included in the data. The original 24-hour matrix was converted into 24 one-hourly matrices using a three step heuristic which uses departure time probabilities and field data volume counts. These matrices are then converted to individual (disaggregated) trips using another heuristic. The final result is that for each entry in the origin-destination matrix we have a trip which starts in the given time slice, with origin and destination links in the correct geographical area. More details can be found in Voellmy et al. (forthcoming).

In the long run, it is intended to move to activity-based demand generation. Then, as explained above one would start from a synthetic population, and for each population member, one would generate the chain of activities for the whole 24-hour period.

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 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 Sect. 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 weren’t 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 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 another view of the network after about 50 iterations with the queue-based micro-simulation for the Gotthard scenario. The figures show the 15-minute aggre-

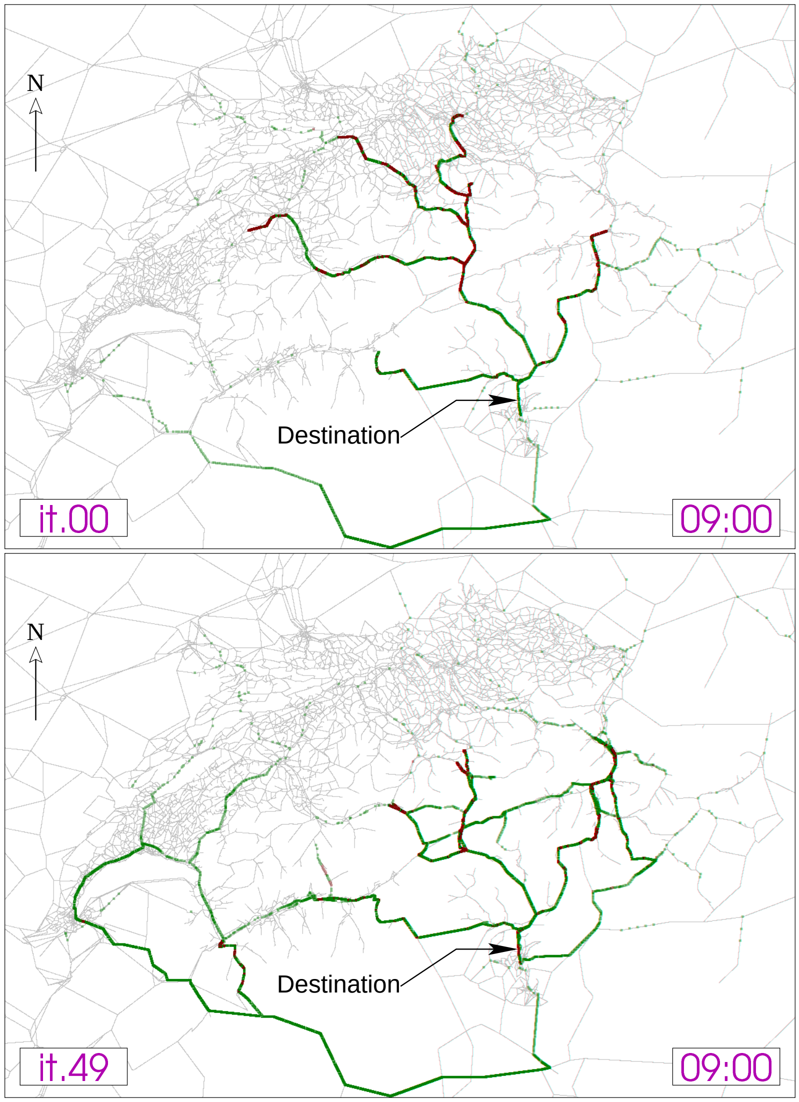

Figure 2: Example of relaxation due to feedback. TOP: Iteration 0 at 9:00 – all travelers assume the network is empty. BOTTOM: Iteration 49 at 9:00 – travelers take more varied routes to try to avoid one another.

gated density of the links in the simulated road network, which is calculated for a given link by dividing the number of vehicles seen on that link in a 15-minute time interval by the length of the link (in meters) and the number of traffic lanes the link contains. In all of the figures, the network is drawn as the set of small, connected line segments, re-creating the roadways as might be seen from an aerial or satellite view of the country. The lane-wise density values are plotted for each link as a 3-dimensional box super-imposed on the 2-dimensional network, with the base of a box lying on top of its corresponding link in the network, and the height above the "ground" set relative to the value of the density. Thus, larger density values are drawn as taller boxes, and smaller values with shorter boxes. Longer links naturally have longer boxes than shorter links. Also, the boxes are shaded, with smaller values having lighter shades of gray, and larger values having darker shades of gray. In short, the higher the density (the taller/darker the boxes), the more vehicles there were on the link during the 15-minute time period being illustrated. Higher densities imply higher vehicular flow, up to a certain point (the dark-gray boxes), but any boxes that are black indicate a congested (jammed) link. All times given in the figures are at the end of the 15-minute measurement interval. The Gotthard tunnel is indicated by a circle; the destination in Lugano is indicated by an arrow.

Figure 4 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 Sect. 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 Fig. 4, but can be verified by zooming in (possible with the electronic version of this paper). 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.

Fig. 5 shows a comparison between the simulation output of Fig. 4 and field data taken at counting stations throughout Switzerland (see Sec. 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 acceptible region at this stage since results from traditional static assignments that we are aware of are no better than this (Esser and Nagel, 2001). Only few simulation results are outside this region; investigation of these points is pending.

5 Computational Issues

A metropolitan region can consist of 10 million or more inhabitants which causes considerable demands on computational performance. This is made worse by the relaxation

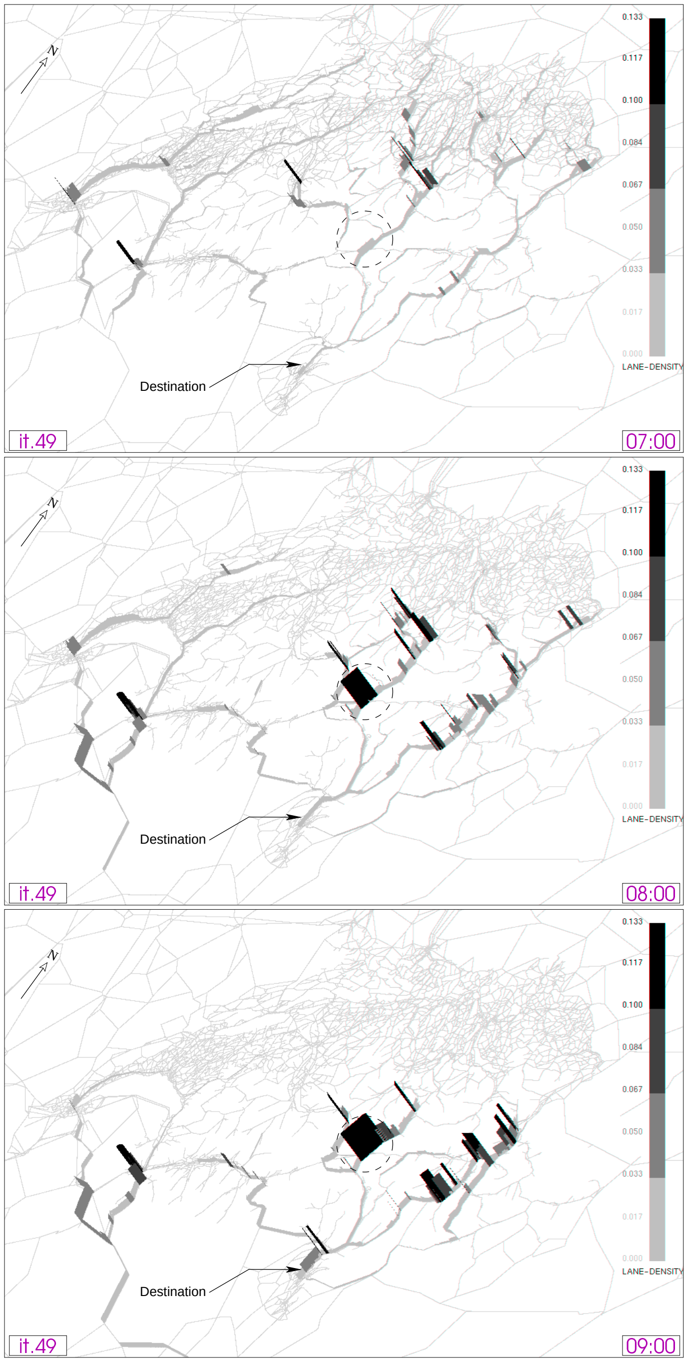

Figure 3: Snapshots at 7:00AM, 8:00AM, and 9:00AM of Gotthard Scenario. The circle shows the traffic jam before the Gotthard tunnel. The arrow indicates the destination of all vehicles.

iterations. And in contrast to simulations in the natural sciences, traffic particles (= travelers, vehicles) have internal intelligence. As pointed out in the introduction, this internal intelligence translates into rule-based code, which does not run well on traditional super-

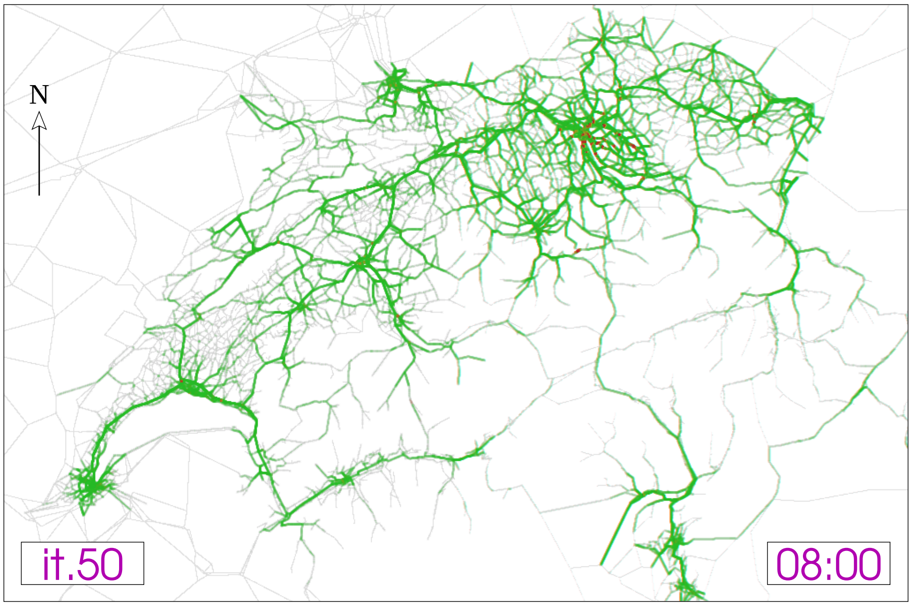

Figure 4: Snapshot of Switzerland at 8:00 AM. From the queue micro-simulation, iteration 50.

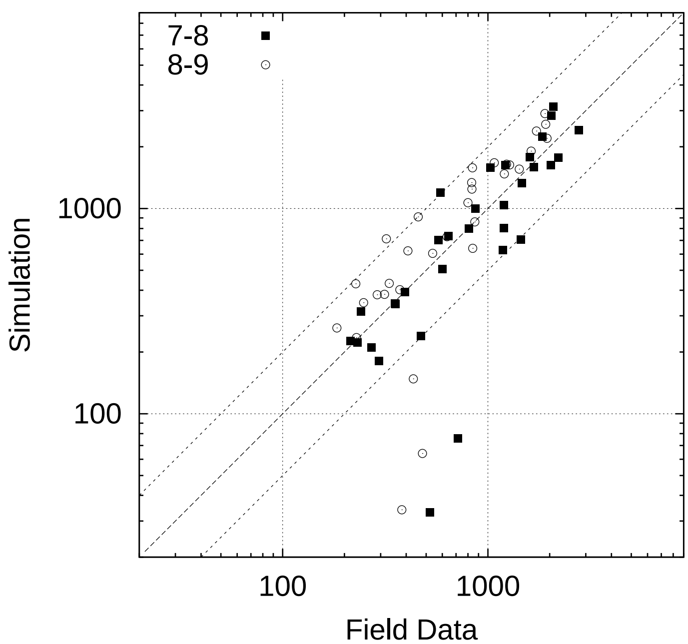

Figure 5: Simulation vs. field data. The x-axis shows the hourly counts from the field data; the y-axis shows throughput on the corresponding link from the simulation. "7-8" and "8-9" refer to the corresonding hours during the morning rush hour.

computers (e.g. Cray) but runs well on modern workstation architectures. This makes traffic simulations ideally suited for clusters of PCs, also called Beowulf clusters. We use domain decomposition, that is, each CPU obtains a patch ("domain") of the geographical region. Information and vehicles between the domains are exchanged via message passing using MPI (Message Passing Interface, www-unix.mcs.anl.gov/mpi).

Table 1 shows computing speed for the queue simulation run on three hours of the Gotthard scenario described in Sect. 3.2. The table lists elapsed time (or wall clock time), real-time ratio, and speedup for the same simulation run on different numbers of CPUs using a standard 100 Mbit Ethernet interface between the computers. The real-time ratio (RTR) is how much faster than reality the simulation is. A RTR of 100 means that one simulates 100 seconds of the traffic scenario in one second of wall clock time. Speedup and RTR are related, in that speedup compares the wall clock time of a multiple-CPU simulation with that of the single-CPU simulation, where as RTR is comparing running time to the simulated time. The simulation scales fairly well for this scenario size and this computing architecture up to about 64 CPUs. Above 80 CPUs, performance does not increase further.

The bottleneck to faster computing speeds is the latency of the Ethernet interface (Rickert and Nagel, 2001; Nagel and Rickert, 2001), which is about 0.5–1 msec per message. Since we have in the average six neighbors per domain meaning six message sends per time step, running 100 times faster than real time means that between 0.5 msec$\times 100 \times 6 = 0.3$ sec and 1 msec $\times\ 100 \times 6 = 0.6$ sec per second corresponding to between 30% and 60% of the computing time is used up by message passing. As usual, one could run larger scenarios at the same computational speed when using more CPUs. However, running the same scenarios faster by adding more CPUs demands a low latency communication network, such as Myrinet, or a supercomputer. Fig. 6 compares the actual experimental RTR between the simulation run over a 100 Mbit Ethernet interface, and a Myrinet interface, with all else being equal. Since Myrinet has a lower latency than Ethernet, the performance is indeed increased as expected. Systematic computational speed predictions for different types of computer architectures can be found in Rickert and Nagel (2001) and Nagel and Rickert (2001).

Number of Processors	Time Elapsed	Real Time Ratio	Speedup
1	552	20	1.00
4	272.7	40	2.02
8	179	60	3.08
16	124	87	4.45
32	82.4	131	6.70
48	74.4	145	7.42
64	65	166	8.49
80	58.4	185	9.45
96	55.6	194	9.93
108	58.2	186	9.48
125	59.2	182	9.32

Table 1: Computational performance of the queue micro-simulation on a Beowulf Pentium cluster. The first column indicates the number of processors used. The second column gives the number of seconds taken to run the first 3 hours of the Gotthard scenario (iteration 49). The third column gives the real time ratio (RTR), which is how much faster than reality the simulation is. A RTR of 100 means that one simulates 100 seconds of the traffic scenario in one second of wall clock time. The fourth column is the speedup, the ratio of the execution time of the simulation to that of a single-processor execution.

6 Discussion and Future Plans

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

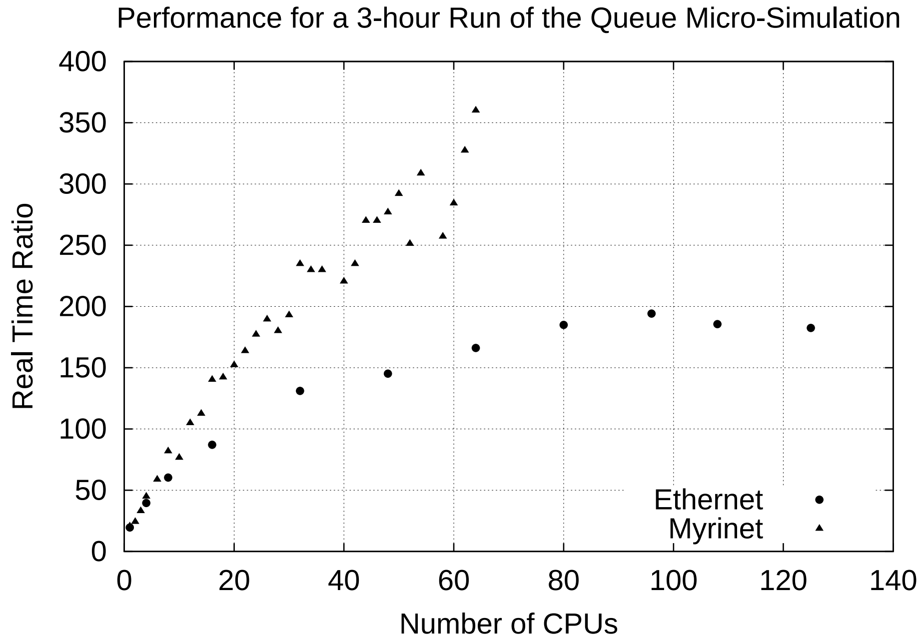

Figure 6: Computational performance of the queue micro-simulation on a Beowulf Pentium cluster using Ethernet and Myrinet network interfaces.

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 (transims.tsasa.lanl.gov; www.transims.net). We attempted to use it in the past years, and indeed some preliminary results were based on it (Raney et al., 2002; Voellmy et al., forthcoming). We however stopped using it because it turned out to be rather difficult to obtain the necessary input data, most importantly lane connectivities across intersections and signal plans. There are automatic generation methods for these attributes from static assignment networks, and we intend to evaluate those. Nevertheless, some aspects will take quite some time and considerable manual work.

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. Tests with the multi-modal TRANSIMS router were unsuccessful, because of at least one serious bug. (This refers to the router of TRANSIMS-

1.0 from fall 1999. Earlier TRANSIMS results were based on a different router. Later versions of TRANSIMS supposedly will have that problem fixed, but are currently not available.) Some of our work investigates how individualized partial knowledge of the road network (mental map) influences route choice.

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.

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 combimation 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).

In addition, a grave shortcoming or 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 explicitely 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.

7 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 preliminary result of a simulation of all of Switzerland is shown, including comparisons to field data from automatic counting stations. Although considerable progress has been made, much work is still to be done.

Acknowledgments

We thank the Swiss regional planning authority (Bundesamt für Raumentwicklung) for the original input data, and Kay Axhausen and Milenko Vrtic at the Institute for Traffic Engineering, Highway- and Railway Engineering (IVT) at ETHZ for adjustments to that input data, and for helpful discussions.

References

C. L. Barrett, R. Jacob, and M. V. Marathe. Formal language constrained path problems. Los Alamos Unclassified Report (LA-UR) 98-1739, Los Alamos National Laboratory, see transims.tsasa.lanl.gov, 1997.

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

J.A. Bottom. *Consistent anticipatory route guidance*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.

J. L. Bowman. *The day activity schedule approach to travel demand analysis*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1998.

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

N. Cetin and K. Nagel, in preparation.

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

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

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

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

David E. Kaufman, Karl E. Wunderlich, and Robert L. Smith. 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, May 1991.

H.S. Mahmassani, T. Hu, and R. Jayakrishnan. Dynamic traffic assignment and simulation for advanced network informatics (DYNASMART). In N.H. Gartner and G. Improta, editors, *Urban traffic networks: Dynamic flow modeling and control*. Springer, Berlin/New York, 1995.

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

K. Nagel and M. Rickert. Parallel implementation of the TRANSIMS micro-simulation. *Parallel Computing*, 27(12):1611–1639, 2001.

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

B. Raney, A. Voellmy, N. Cetin, M. Vrtic, and K. Nagel. Towards a microscopic traffic simulation of all of Switzerland. In P.M.A. Sloot, C.J.K. Tan, J.J. Dongarra, and A.G.

Hoekstra, editors, *Computational Science – ICCS 2002, Part I*, number 2329 in Lecture Notes in Computer Science, pages 371–380. Springer, Heidelberg, Amsterdam, 2002.

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

M. Rickert and K. Nagel. Dynamic traffic assignment on parallel computers in TRANSIMS. *Future generation computer systems*, 17(5):637–648, 2001.

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

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

A. Voellmy, M. Vrtic, B. Raney, K. Axhausen, and K. Nagel. Status of a TRANSIMS implementation for Switzerland. *Networks and Spatial Economics*, forthcoming. See www.inf.ethz.ch/˜nagel/papers.

M. Vrtic. Dynamische Umlegung des Strassenverkehrs. IVT Seminar, ETH Zürich, December 2001. See www.ivt.baug.ethz.ch.