Balmer, M.; Nagel, K.; Raney, B.

Large-Scale Multi-Agent Simulations for Transportation Applications

Balmer, M.; Nagel, K.; Raney, B. (2004). Large-Scale Multi-Agent Simulations for Transportation Applications. Journal of Intelligent Transportation Systems, 8(4), 205–221. https://doi.org/10.1080/15472450490523892

Large-Scale Multi-Agent Simulations for Transportation Applications

MICHAEL BALMER
Institute for Transportation Planning and Systems, Zürich, Switzerland

KAI NAGEL
Institute for Land and Sea Transport Systems, TU Berlin, Germany

BRYAN RANEY
Institute for Computational Science, Zürich, Switzerland

In many transportation simulation applications including intelligent transportation systems (ITS), behavioral responses of individual travelers are important. This implies that simulating individual travelers directly may be useful. Such a microscopic simulation, consisting of many intelligent particles (= agents), is an example of a multi-agent simulation. For ITS applications, it would be useful to simulate large metropolitan areas, with ten million travelers or more. Indeed, when using parallel computing and efficient implementations, multi-agent simulations of transportation systems of that size are feasible, with computational speeds of up to 300 times faster than real time. It is also possible to efficiently implement the simulation of day-to-day agent-based learning, and it is possible to make this implementation modular and essentially "plug-and-play." Unfortunately, these techniques are not immediately applicable for within-day replanning, which would be paramount for ITS. Alternative techniques, which allow within-day replanning also for large scenarios, are discussed.

Keywords Multi-agent simulation; Transportation planning; Transportation application; Parallel computation; Route planning; Traffic simulation; Agent learning; Activity planning

INTRODUCTION

The negative impacts of traffic, such as noise and other emissions, accidents, or waste of time in congestion, are recognized as problems. On the other hand, eliminating traffic is not an option: Mobility of goods is necessary for economic performance, and mobility of people corresponds to a desire of most humans. The latter is, for example, visible in the increasing share of leisure traffic. In this situation, two reactions are necessary:

Address correspondence to K. Nagel, Institute for Land and Sea Transport Systems, TU Berlin Sek.SG12, Salzufer 17-19, 10587. E-mail: nagel@vsp.tu-berlin.de

- A search for better technologies.
- A consensus in society about the balance between desire for mobility, acceptance of negative impacts, and willingness to spend on better technologies.

One such technology is ITS.

The impact of any infrastructure change, including ITS, depends critically on people's behavior. Forecasting this impact, in particular in complex situations, is, therefore, a difficult problem. For ITS technology, this forecast is even more challenging since ITS has a dynamic component that brick-and-mortar technologies do not possess: Individual route guidance or public variable message signs can change at any given point in time. In contrast, a new road, once open, will remain there for many years.

It is therefore critical that models of ITS treat human behavior in a meaningful way. One approach to this problem is to write simulation models which treat each traveler as a microscopic entity and to model that entity's reaction to the system directly. This is what is meant by the term "multi-agent" simulation (MASim). Optimally, such a system would model each individual traveler's space-time path through the virtual system; this would generate synthetic traffic sensor data that would be sent to the virtual traffic management centers; those centers would compute strategies (such as individual route guidance) and send them back to the agents; and then each individual agent would react to that information according to its individual behavioral profile. This approach would make the inclusion of behavioral diversity, for example based on attributes of the persons and/or the alternatives, conceptually easy to model.

Several challenges are connected to the successful application of MASim technology. The first one is the computer implementation of a full MASim system for transportation applications. Although this may seem unrelated to transportation, it is probably not: Computer implementation decisions have an impact on what is easy to model and what is not.

Next, human behavior needs to be modeled realistically. This is the topic of several other presentations of this workshop. One aspect of human behavior is human learning, for example, the realistic day-to-day dynamics of how the system adapts after a major infrastructure change. In the past, this problem has been avoided by just considering the "relaxed" state that is reached once all learning has stopped—the assumptions were that this state would be close to a Nash Equilibrium (NE), that the NE would be unique, and that, therefore, the computational and modeling challenge was to reach that NE as quickly as possible, rather than to model human learning. This approach may no longer be possible with ITS technology.

A final challenge is the size of realistic systems: Metropolitan areas often consist of several millions of potential travelers, and all of them need to be simulated directly in order to make the MASim approach work. One might argue that one should start with smaller systems. The counter-argument to this is that the relevant systems are the large ones, and that the behavior of large systems may depend more on large scale collective effects and less on individual behavior.

This paper will discuss in some detail that a multi-agent simulation of large scale real-world scenarios is possible, and the techniques necessary to achieve this. It will also discuss that those techniques do not immediately transfer to en route (within-day) replanning, which is nevertheless important for ITS. A possibility to simultaneously allow within-day replanning and efficient large scale computing is discussed near the end of the paper.

Let us make a remark on the difference between simulation-based ITS *evaluation* and simulation-based ITS *application*. In the first case, the ITS system, as a "black box," is plugged into the simulation system: The simulation system generates synthetic sensor output and communicate it to the ITS system; the ITS "black box" receives this data, computes its response, and send the corresponding measures, such as variable message signs, to the simulation system; the simulation system has the travelers react according the behavioral rules. In this case, the MASim is used as an evaluation tool. An example of such an application, albeit not fully agent-based, is MITSIMLab (2004).

In the second case, the MASim is used as a tool to *generate* the ITS system response in the first place. It could, for example, be used to help with state estimation, or to compute several forecasting scenarios as a function of different possible management strategies. Examples of such applications, albeit once more not fully agent-based, are DYNASMART (2003) and DYNAMIT (2003).

The remainder of this paper concentrates on how MASim could be used for the *evaluation* of ITS systems.

Multi-agent simulations of physical systems generically consist of at least two components (Ferber, 1999, chap. 4; see Figure 1):

- The component(s) which compute(s) the physical aspects of the system (such as excluded volume, limits on acceleration, etc.). This will be called the *mobility simulation* (see below).

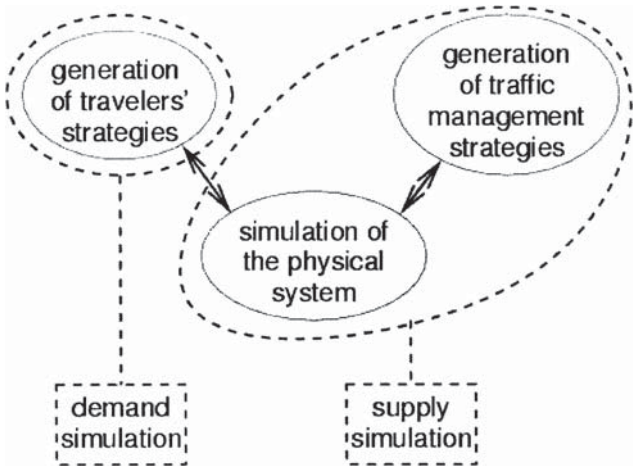

Figure 1 Components of a multi-agent simulation for traffic management.

- The component(s) which compute(s) the strategic decisions of the agents. On the traveler's side, this includes the reaction to ITS technology, but also long-term aspects such as choice of residence or workplace location. Since ITS devices can also be treated as agents, strategies for those ITS devices can be computed in a similar way as strategies for the travelers.
- In general, there will be more than one strategy (plans) generation module. Typical examples of plans generation modules are departure time choice, or route choice.

Strategy generation is shortly discussed below.

The specific distinction between the strategy generation and the mobility simulation is made because they need very different sets of tools, and people with experience in one do not always have experience in the other. For transportation applications, one needs to make both components useful for the real world. This can, in our view, best be achieved by completely separating the strategy generation from the mobility simulation. Then, strategies are submitted to the mobility simulation, which executes them and returns the strategies' performance. The strategy generation can, as discussed below on learning, react to this information.

Such a setup is still no guarantee that the mobility simulation has any relation to the real world. However, it is now possible to construct the mobility simulation with principles from simulation as it is known in the natural and engineering sciences, where there is much more experience with the simulation of realistic systems.

In the transportation community, sometimes the terms "demand simulation" and "supply simulation" are used (see, e.g., http://its.mit.edu). They are not the same as our distinction: As can be seen in the schematic Figure 1, the supply simulation combines the simulation of the physical system and the simulation of the traffic management strategies into one module. The distinction of supply and demand may be appealing from an economics perspective; from a computing perspective it is not: The simulation of the physical system is considerably different from simulations of strategy generation, and should therefore not be combined into a single conceptual module.

Besides the mobility simulation and the strategy generation, there are two more components which are necessary to make the whole simulation work: a method to do learning/feedback, and initial/boundary conditions:

- In general, it is not possible compute *complete* strategies and then run the mobility simulation based on them, since it is computationally infeasible to compute a strategic answer to all possible conditions. Therefore, the simulation system needs to implement some kind of *learning or feedback*, by which the agents modify and improve their strategies. The most prominent example for this is the reaction to congestion: Agents will make tentative plans, the mobility simulation will execute them, agents will revise their plans, these plans will be executed again, etc. Issues related to learning will be treated below.
- *Boundary conditions* refers to all data that remains fixed throughout the simulation, such as the road network. *Initial conditions* are how the simulation is started. Examples of both will be described, with respect to the specific scenario of a simulation of "all of Switzerland," below. This section will also contain some comparison to real world measurements.

Below, we will discuss computational aspects. In particular, it is of critical importance to make the whole iteration cycle computationally fast in order to be able to do systematic computational experiments. The paper is concluded by a longer section on future plans, and a summary.

THE MOBILITY SIMULATION

As stated above, the mobility simulation refers to the simulation of the physical transportation system. It computes what happens to the agents' strategies when they are confronted with (a synthetic version of) the real physical world. That synthetic version of the real world specifically includes the *interaction* between agents, which is responsible for congestion. This implies that a mobility simulation that fits into our framework needs to be capable of executing the strategies (plans) of all agents simultaneously.

As also stated above, the mobility simulation needs to return information about the agents' performances to the strategy generation modules. In our implementation, this is achieved by so-called *events* which are output every time they are triggered. The format of events is "(time, agent ID, event type);" examples of events are "agent XY left from/arrived at activity location" or "agent XY entered/left link." Clearly, it is possible that ITS devices also write event information to file. Our events format has the advantage that it is very easy to implement; it should therefore be possible to implement it into any existing microsimulation. It also has the advantage that the aggregation of the data is completely left to the strategy generation modules. These can aggregate information according to their own preferences; for example, a router will aggregate information according to link IDs, while the agent database

(see below) will aggregate information according to agent IDs.

There are several methods for designing such a mobility simulation. They will be introduced in the following. Only a subset of them is useful for ITS.

The field of traffic operations has a long history of microscopic simulation. Microscopic means that each vehicle is individually resolved, and it is modeled with many aspects of its driving dynamics. There are many different methods for the microscopic simulation of traffic, including coupled differential equations (Herman et al., 1959; Newell, 1961; Bando et al., 1995), coupled maps (= models with continuous space and velocity but coarse-grained discrete time; (Wiedemann, 1994; Krauß, 1997), and the so-called cellular automata approach (Gerlough, 1956; Nagel and Schreckenberg, 1992; Chowdhury, Santen, and Schadschneider, 2000).

Alternatives to microscopic models are fluid-dynamical models. Maybe the easiest way to understand them is to cut the road into segments with, say, a length of 1 km, and then to formulate the mass conservation equations for these segments. First,

$$N_{t+\Delta t}(x) = N_t(x) + \Delta t \left(Q_t\left(x - \frac{\Delta x}{2}\right) - Q_t\left(x + \frac{\Delta x}{2}\right) + S_t(x) \right)$$

which just states that the number of vehicles in segment x is increased by inflow upstream, reduced by outflow downstream, and changed by source/sink terms. The crucial step is to couple the flow terms $Q_t(x)$ to the number of vehicles in the cells. Standard fluid-dynamical partial differential equations are recovered in the limit of $\Delta t \to 0$, $\Delta x \to 0$. A recent example of such an approach to traffic flow is NETCELL (Cayford, Lin, and Daganzo, 1997). Fluid-dynamical models *per se* do not track individual particles/vehicles, and are therefore unsuitable for ITS (but see further down).

Even more aggregated (called macroscopic) approaches are the volume-based cost function of static assignment, or the gravity model often used in trip distribution (e.g., Ortúzar and Willumsen, 1995). Those do not have any temporal dynamics and are therefore even less suitable for ITS.

As said above, in traffic a pure fluid-dynamical method is often not very useful, since it does not allow one to track individual particles/vehicles. In such cases, hybrid methods, corresponding to the smoothed particle hydrodynamics method (Gingold and Monaghan, 1977), are useful: They keep individual particles, but move them according to fluid-dynamical equations. Examples from the traffic simulation area are DYNEMO (Schwerdtfeger, 1987), DYNAMIT (2003) and DYNASMART (2003).

In order to push the limits of computational feasibility, it makes sense to search for very fast microscopic (or mesoscopic) models. A good example of this is the so-called queue model (Gawron, 1998). It has similarities to queuing theory and queuing models (e.g., Simro and Powell, 1992), because it decomposes a street network into queues. Vehicles are discharged from a queue according to the flow capacity. They then enter the next link, along which they travel with free flow speed. Once they have reached the end of the link, they are added to the queue and they are discharged once it is their turn. So far, this is indeed a standard queuing model. The important distinction is that in the queue model of traffic, the number of vehicles on a link (moving plus waiting) is limited. Once the link is full, no vehicle can enter, and this constraint is propagated upstream, so that in addition to the flow capacity constraint of the link itself also the storage constraint of the downstream link can limit outflow from a link. A critical issue here is the allocation of empty spaces. One possibility is to do this proportional to the capacity of the incoming links (Cetin and Nagel, 2003).

This model, by definition, models free speeds and flow capacities realistically. It has some shortcomings in terms of the speed of the backpropagating kinematic wave, and is limited in its representation of dynamics that go beyond the queuing paradigm, such as lane changing, faster cars passing slower cars, complicated intersection dynamics, etc. Nevertheless, rather realistic results can be achieved with this model, as will be shown below.

Sometimes, the term mesoscopic models is used. By definition, they lie between microscopic and macroscopic models. Yet, the definition is not totally clearcut of what is included and what not.

STRATEGY GENERATION

As described above, the mobility simulation reads strategies/plans. These are descriptions of where travelers enter and leave the network, which turns travelers take at intersections, etc. 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 anthill. 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. This is further discussed below.

Important strategy generation modules for the simulation of travelers are route choice, mode choice, activity time choice (which includes departure time choice), activity location choice, activity pattern choice, etc. All of these represent possible dimensions of decision making of the traveler. For ITS applications, all of these may be of importance, since all of them can be changed in reaction to information and guidance measures.

Similarly, as becomes clear by looking at Figure 1, traffic management strategies can (in principle) just be fed into the mobility simulation as well. That is, a group of traffic lights or a group of variable messages signs can follow a plan or strategy in the same way as a traveler can follow a plan.

Our own vision with respect to the simulation of these is a plug-and-play architecture, where strategic modules programmed by different groups can cooperate in one simulation system. The general idea is that the mobility simulation outputs *events*, which is information of the type "(time, agent-id, event)." Examples for events are "traveler entered/left a link," "traveler arrived at/left activity location," or "sensor sensed a vehicle passing over it." Strategy generation modules then read the events, extract those events that they are interested in, and generate plans in response.

An example of a strategic module is route generation. 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 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, for example into fifteen-minute time bins, and the router finds the fastest route based on these time bins. Apart from relatively small and essential technical details, the implementation of such an algorithm is straightforward (Jacob, Marathe, and Nagel, 1999). It is possible to include public transportation into the routing (Barrett, Jacob, and Marathe, 2000); in our current work, we look at car traffic only. Note that such a routing algorithm, with small variations, can both be used to compute a behavioral response of an agent, and routing guidance by a traffic management center.

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 an NE. For route assignment, an NE is reached when no traveler can improve its travel time by selecting a different route.

Static Approaches

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; Cascetta, 2001). 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; Cascetta, 2001). 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. Instead of T_i, some generalized cost C_i can be used. 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 come 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).

Dynamic Approaches

Static assignment does not possess any dynamics; that is, traffic is represented by time-independent streams. This precludes, for example, the representation of queue spillback, 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.

Is *is* possible to maintain the NE or SUE problem statements for the dynamic approaches, 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). The solution is, however, no longer unique (Daganzo, 1998), and therefore the solution process is no longer a purely mathematical exercise but now also has behavioral aspects. Two agent-based approaches, which allow a behavioral interpretation, are described next.

"Basic" Agent-Based Feedback

Both basic static assignment and static 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, Wunderlich, and Smith, 1991; Bottom, 2000). This is a cycle in which all agents select plans (e.g., route plans), execute them in the traffic simulation, then some agents revise their plans, and they are executed again, etc., until some kind of stopping criterion is fulfilled. A possible version of this is:

1. The system begins with an initial set of plans, one per agent, based on some kind of initial guess.
2. The mobility simulation is executed with the current set of plans.
3. 10% of the population requests new plans from the strategy level, which bases the decision on performance information from the mobility simulation. The new plans then replace the old plans for the "replanned" agents in the set of current plans.
4. This cycle (i.e., steps 2 through 3) is run many times, until some kind of stopping criterion is fulfilled.

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 real-world systems, where one would assume that the replanning fraction is given by individual behavioral rules, not by a system-wide requirement.

An alternative approach is to use Eq. (1) to chose between different plans. Where, however, do the different plans come from? In the following, we present an approach where additional plans 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 plans, and the outcome (performance) of those plans. The agent database stores the performance of a plan as a numerical "score" associated with that plan. The score can be, for example, the travel time of the plan, or can be calculated from some utility or fitness function. The specific steps in the relaxation cycle are:

1. The system begins with an initial set of plans, one per agent. In the case of the results presented below in A Practical Scenario a plan 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 plan is stored in the agent database (Raney and Nagel, 2002, 2003), which represents its memory of previously tried plans. Since the agents at this point have only one plan apiece, they automatically select this as their next plan to execute.
3. The mobility simulation is executed with the set of selected plans.
4. Each agent calculates the score of his/her plan based on the outcome of the simulation. In A Practical Scenario below, "performance" will mean 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 plan that was used.
5. A fraction of the population (10% at present) requests new plans from the strategy modules, which base them on information from the last mobility simulation. The new plans are then stored in the agent database and, being new, are mandatorily selected by the agents.
6. Agents who did not request new plans choose a previously tried plan from the agent database, by comparing the performance values for the different plans, without knowing anything else about the plans. Specifically,

they use a multinomial logit model $p_i \propto e^{\beta S_i}$ for the probability p_i to select route i, where S_i is the corresponding memorized score and β is an empirical constant. This process is explained in detail by (Raney and Nagel, 2002, 2003).

7. This cycle (i.e., steps three through six) is run many times, until some kind of stopping criterion is fulfilled.

An advantage of the agent database is that the system is considerably more robust than without (Raney and Nagel, 2002, 2003). Without the agent database, it is imperative that the strategy generation modules generate plans which are an improvement over the previous solution. This means, for example, that 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 strategy generation modules that will be very difficult to fulfill. Use of the agent database means that the strategy generation modules can be much more creative in the generation of new plans: Plans which turn out to be bad plans 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 et al., 1988). From here, alternative methods of agent learning, such as Machine Learning or Artificial Intelligence, can be explored.

Illustration of Learning and Feedback

Figure 2 shows an example of how the feedback mechanism works. In this scenario, many travelers travel from "home" to "work," as indicated in the figure. Travelers have nine different options, all with the same characteristics. Initially, all agents use the central route. Over time, they learn about the other options. Eventually, the system reaches a state that is similar to a Nash (or user) equilibrium: No matter which path the agents take, all have approximately the same travel time. Once also notices in the figure that the situation is not *fully* symmetric between the different paths. This is due to the stochasticity both in the choice behavior and in the traffic simulation. More details about this can be found in Raney and Nagel (2004).

Day-to-Day vs. Within-Day Replanning

The literature (e.g., Cascetta and Cantarella, 1991) discusses the difference between *day-to-day* and *within-day replanning*. The difference is that in the former, the agents modify their plans only "overnight" (i.e., between runs of the mobility simulation), while in the latter they modify their plans while the mobility simulation is running.

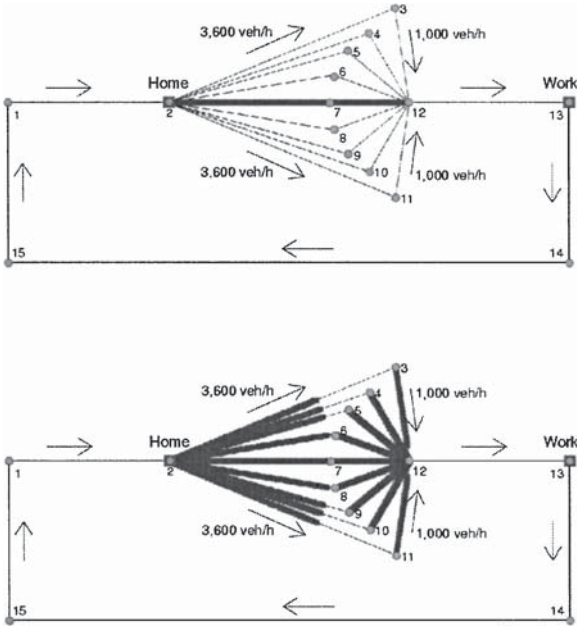

Figure 2 Diagram of the testing network overlaid onto example snapshots of the same time of day (top) before and (bottom) after many iterations of route replanning. After replanning, the agents have spread onto the different available routes between home and work.

It is obvious that for the simulation of ITS systems, some version of within-day replanning is necessary, because otherwise travelers in the simulation simply do not react to ITS measures. There are essentially three ways to implement within-day replanning into a multi-agent system:

- Agents (including the traffic management center [TMC]) precompute *conditional strategies*. For example, a traveler could say that he/she follows a variable message sign if there is one, or the TMC could say that it will switch variable message signs in a certain way under certain conditions. The mobility simulation would then simply execute these conditional strategies.
- Another implementation option is that, at predefined times, say every hour, the mobility simulation is stopped, and all agents (including the TMC) can decide if they *would have changed* their plans during the last hour. If so, the simulation is moved back one hour and run again with the revised plans. This is done over and over again, until at the end of the hour, no agent wants to change any more, at which point the simulation of the next hour will be started. The main advantage of this approach is that it can be implemented with the same tools as day-to-day replanning. Its main disadvantage is that

it is computationally slow, and that it is conceptionally simple only when all modules are deterministic.

- *True within-day replanning*: Finally, both the travelers and the TMC could be enabled to react *while* the mobility simulation is running. At any point in time, any agent (including the TMC) could stop the mobility simulation, replan its own strategy, and then start the mobility simulation again. More realistic implementations would explicitly take care of delays; for example, information from sensors at time t may not be used before a time $t + \tau$ by the TMC.

To make matters worse, the conceptual decision is interwoven with the computational implementation. A straightforward implementation implements within-day replanning as a subroutine of the mobility simulation. This, however, means that is has to be written in the same programming language. In addition, it means that it will use the same memory and CPU as the mobility simulation, which may not be possible for large scale scenario. That is, in our view at this point no robust technology exists for the implementation of within-day replanning into multi-agent (traffic) simulations that is at the same time modular and large-system capable. Future plans relating to this issue are discussed in Future Plans, below.

Toward a Theory of Multi-Agent Learning

Despite some efforts both in computer science (e.g., Weiss, 1999; Ferber, 1999) and in traffic (Cascetta and Cantarella, 1991; Cantarella and Cascetta, 1995; Bottom, 2000), a general theory of multi-agent learning seems to be missing. Some aspects of agent learning are covered by behavioral aspects, which are treated in other parts of this workshop. Here, we will concentrate on how multi-agent learning interacts with the computation. There are two subproblems to solve: single-agent learning, and multi-agent coevolution. We will treat them one by one.

Single-Agent Learning

Here, one should consider the problem of a single learning agent when confronted with a stochastic environment. In order to make this problem better defined, let us assume that this environment is stationary; that is, the random quantities do not "drift." An example, relevant to traffic, would be a traveler who is going through a typical weekday over and over again and needs to find a good or the best solution in terms of activities, mode choice, and route choice.

This problem can be formulated as a *reinforcement learning problem* (RLP). A typical formulation of the RLP is that there are states, s, and that for each state there are several possible actions a. The actions influence the state transition probabilities, that is, $T(s, s', a)$. For each state-action-pair there is a reward, $R(s, a)$. The goal of the agent is to find actions that is, (s) such that the time-averaged reward flow is maximized.[1] For traffic, a state could for example be "being at intersection X at 8 am" or "having worked for seven hours at 4 pm." A transition could be the next network link to take, or a transition to another activity (Charypar, Graf, and Nagel, 2004).

A possible solution method for the RLP is *Q-learning* (e.g., Russel and Norvig, 1995). In Q-learning, the agent samples all possible transitions very often, while back-propagating future rewards when taking that transition. A detailed description is beyond the scope of this paper. Q-learning solves the RLP optimally only when the system is ergodic, that is, when any state can be reached from any other state, via a sequence of transitions. In practice, one also needs that the state space cannot be too large, because otherwise sampling of the complete state space becomes impossible in plausible time, and Q-learning will find a solution that is locally but not necessarily globally optimal. It is this second problem that haunts us in traffic simulation: It will *not* be possible to explore, for each agent in the traffic simulation, the complete state space of all options. In this situation of limited search time, mathematical properties of the system such as said ergodicity, or the fact that a Markovian system goes to a steady-state density, are no longer useful, and building a useful theory becomes considerably more difficult.

Q-learning is a useful general purpose algorithm to illustrate the problem, but it fails when the search space becomes too large. In such cases, it is necessary to look for alternatives. For example, Q-learning for route generation is orders of magnitude slower than the Dijkstra algorithm (above). In other words, when the structure of the problem is known, then better algorithms than Q-learning can be used. All these algorithms fall under the category best reply algorithms, since they find the best reply of the agent to the last iteration.

Often, it is behaviorally not plausible that agents always select the best option. In such cases, discrete choice theory (Ben-Akiva and Lerman, 1985) assigns probabilities to *all* possible options, and the agent makes a random draw according to those probabilities. The main disadvantage of discrete choice theory is that the necessity to

[1] Most RLP formulations maximize some discounted expected reward, with a discount factor β. In the limit $\beta \to 1$, our formulation is obtained, which we find more intuitive.

compute probabilities for *all* options makes it even harder to compute than best reply algorithms.

Since optimal solutions cannot always be computed, there is a growing number of heuristics that find good but non-optimal solutions. Those algorithms are often inspired by evolutionary biology and are therefore called evolutionary or bio-inspired algorithms; typical exponents are Genetic Algorithms (Goldberg, 1989; Charypar, and Nagel, 2003) or Ant Colony Optimization. As long as such algorithms are applied to model human behavior, the fact that they do not find optimal solutions is not a disadvantage. Nevertheless, there are open questions with respect to validation and calibration.

Finally, there are path dependent algorithms, where what the agent does depends on what the agent knows (mental maps). For example, the agent can memorize which parts of the network it has already seen, and prefer in future decisions those parts of the network. Such algorithms are currently implemented at some places (Arentze and Timmermans, 2003; Kistler, 2004), but their effect on large scale transportation planning applications is entirely untested.

Note once more that all of these approaches can also be used for the strategy generation of a traffic management center. The only difference is that a TMC will put more emphasis on exact and good algorithms, and less on behavioral realism.

Coevolution

In traffic, it is not true that a single agent is learning while all other agents have fixed strategies; rather, all agents learn simultaneously. This is typically termed coevolution. It means that a learning agent is no longer faced with a stationary environment, but one which "drifts" because the other agents also learn.

A possible theory for coevolutionary systems is developed by Hofbauer and Sigmund (1998). That book essentially concentrates on biological systems. Different strategies are represented by different species. Encounters between species are rated by different scores ("fitness"), and the replication rate of species depends on these scores. Consider for example the so-called hawk-dove game, where encounters between two hawks result in severe negative scores for both, encounters between two doves result in zero scores, and encounters between a hawk and a dove result in a positive score for the hawk and a negative score for the dove. Clearly, a population of doves can be "invaded" by a hawk, while a population of only hawks cannot survive. A steady state can be reached with a small number of hawks (which ensures that hawks encounter each other only rarely) and a large number of doves. In this theory, the learning system is just an example of a dynamical system. As is well known, a dynamical system can reach fixed point attractors, periodic attractors, or chaotic attractors. As it turns out, for certain learning dynamics the fixed points are Nash Equilibria, meaning that one recovers some of the solutions of static game theory. However, the other attractors do not correspond to anything known from static game theory, which implies that simulations may display a long-term behavior that does not correspond to anything known from static game theory.

Although this shows a possible way to go for a theory for multi-agent learning, it is not possible to use this theory directly. The assumption for the theory is that there are a number of members of a certain species, and their scores decide about increase or decrease of these numbers. Translated to traffic, this would mean that travelers with unsuccessful strategies get weeded out while ones with successful strategies replicate. Although this could work as a metaphor, this is certainly not directly useful for real-world implementations.

An issue with coevolution is that different agents can learn with different learning speeds. This becomes particularly critical once there are entities on different hierarchy levels of the design. For example, there could be travelers that adapt to measures of a traffic management center, and the traffic management center could react to the behavior of the travelers. Depending on "who learns faster," the results can be quite different (e.g., Zuylen and Taale, 2004; Nagel, Strauss, and Shubik in press). The issue is related to "sequential games," "subgame perfection," and "Stackelberg games" in game theory.

As a somewhat extreme illustration, let us assume that there are two routes to a destination, and both are served by greedy road pricing agencies that attempt to maximize profit. If one agency raises prices, revenues will initially go up. But eventually more travelers will use the alternate route and (assuming there are no additional congestion effects) the agency may make *less* money than before. In consequence, the agency is driven back to the lower price. If, however, the agency evaluates the effect of the price change immediately after one day, it will conclude that the price increase was successful, and will stick with it.

A PRACTICAL SCENARIO

One goal of our work is a full twenty-four-hour simulation of all of Switzerland, including transit traffic, freight traffic, and all modes of transportation. The simulation

should include within-day replanning, making it capable of evaluating ITS devices/strategies. This will involve about 7.5 million travelers, and more than twenty million trips (including short pedestrian trips, etc.).

Our implementation has not yet reached this goal, and it will take some more time until we get there. However, in order to make sure that our system will be useful in the real world, systematic studies are performed with intermediate levels of system capabilities. At this point, there is unfortunately no ITS capability that can be tested. Nevertheless, our results allow to make some predictions about the possible usefulness of our system.

In the following, results of a study done in 2002/03 are reported. Further details of this study can be found in (Raney et al., 2003).

That study took a subset of the data for the full twenty-four-hour, car-only simulation, and used the demand for the morning rush-hour, from 6:00 A.M to 9:00 A.M. This subset contained about one million trips.

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

The Street Network

The street network 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, Koblo, and Vödisch, 1999). The network supposedly contained the status for 1999, but contained at least one major error (a high capacity tunnel in Zürich was missing). Our initial simulations resulted in traffic gridlock in Zürich, which was also reflected in the VISUM (PTV, 2003) 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.

After our modifications, the network has the fairly typical size of 10,564 nodes and 28,622 links. Also fairly typical, the major attributes on these links are type, length, speed, and capacity.

Demand

Our starting point for demand generation for the scenario described here were twenty-four-hour origin-destination matrices from the Swiss regional planning authority (Bundesamt für Raumentwicklung).

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

VISUM assignment of the matrices showed 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.

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

The OD matrices assumed traffic analysis zones (TAZs) while in our simulations trips start on links. We converted 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 database.
- 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 led to a list of approximately five million trips, or about one million trips between 6:00 A.M. and 9:00 A.M. Since the origin-destination matrices are given on an hourly basis, these trips reflect the daily dynamics. Intrazonal trips are not included in those matrices, as by tradition.

The Simulations

The above scenario was fed into two different models: First, into a VISUM (PTV, http://www.ptv.de) assignment which is a relatively standard assignment (Sheffi, 1985) except that it is dynamic on an hourly basis (Friedrich et al., 2000; Vrtic and Axhausen, 2002), and second into an agent-based approach. The components of the agent-based approach were as follows:

- For the mobility simulation, the queue simulation was used, as described above.
- For strategy generation, only a time-dependent router was used, as described above.
- The iterations used the agent database, as described above. The score calculated by the agent database is the negative of the travel time of the plan.

Fifty iterations proved to be sufficient to reach relaxation, as was tested by looking at average travel times. The following section describes the results reached after those fifty iterations.

Results

Figure 3 shows a result of the Switzerland 6–9 Scenario. 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 Figure 3 and field data taken at counting stations throughout Switzerland (see above 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 (Figure 4 [right]; see also Esser and Nagel, 2001).

Figure 4 (right) 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 $\langle \cdot \rangle$ means that the values are averaged over all links where field results are available.

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

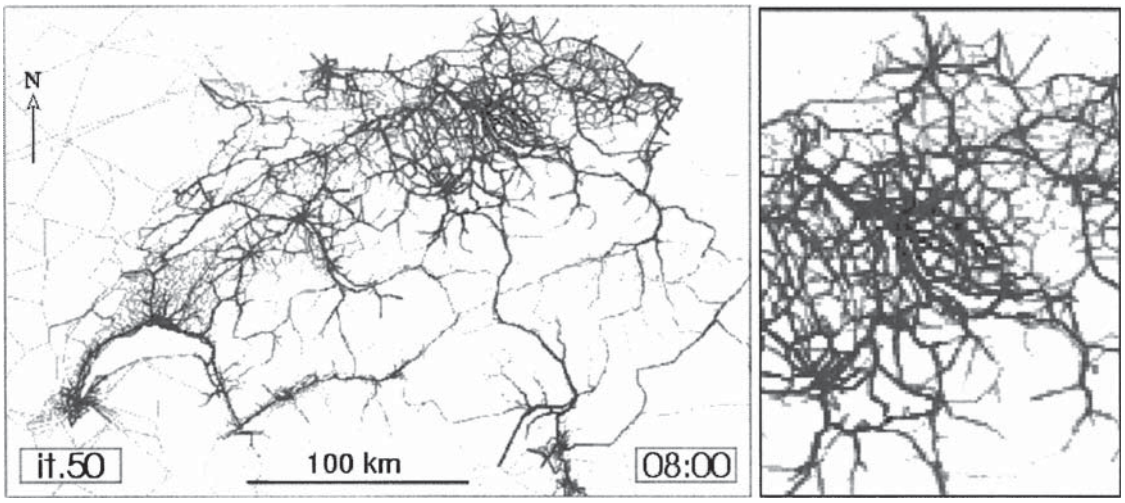

Figure 3 Snapshot of Switzerland at 8:00 A.M. From the queue mobility simulation, iteration fifty. The right side shows a close-up of the Zurich area. Cars are plotted as green/light dots when they are driving in free traffic, and as red/dark dots, when they are stuck in jams. Significant traffic jams are only found in the Zurich area.

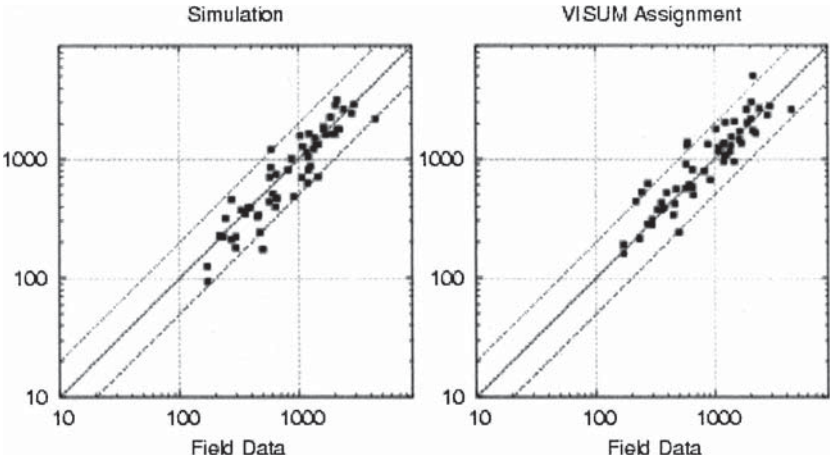

Figure 4 Comparison to Field Data. *(left)* Simulation vs. field data for the 50th iteration. The x-axis shows the hourly counts between 7:00 A.M. and 8:00 A.M. from the field data; the y-axis shows throughput on the corresponding link from the simulation. *(right)* VISUM assignment vs. field data. The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

values in Portland were better than the ones obtained here.

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

Implications for ITS

As stated before, the above study did not contain aspects of ITS. Nevertheless, they allow some outlook towards future possibilities. First, the above study makes clear that it is possible to use MASim for large scenarios: Switzerland, with about seven million inhabitants, is a good proxy for a large metropolitan area. Second, the quality of the results is, at this stage of the technology development, about as good as that of assignment methods. Further improvements are expected with further uses of the MASim technology. Overall, this indicates the general applicability of MASim for transport planning. Since ITS is conceptually much easier to integrate into MASim than into assignment, it is expected that real-world applications of ITS in MASim will soon become available and useful.

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%

COMPUTATIONAL ASPECTS

Computational Performance of the Mobility Simulation

It is possible to make the mobility simulation parallel, that is, to give different pieces of it to different Central Processing Units (CPUs). One good option is to do this via domain decomposition; that is, the geographical area is decomposed into domains, and each CPU computes traffic within that domain. These domains need to exchange information at their borders, which can be achieved by messages. For messages, existing message passing software such as Message Passing Interface (MPI, http://www.mcs.anl.gov/mpi/) or Parallel Virtual Machine (PVM, http://www.epm.ornl.gov/pvm) can be used.

In this situation, simulation time per time step is a sum of the time spent on computation and the time spent on communication,

$$T(p) = T_{cmp}(p) + T_{cmm}(p), \tag{2}$$

where p is the number of CPUs. For the purposes of an intuitive understanding, let us assume that

$$T_{cmp}(p) \approx \frac{T_1}{p} \tag{3}$$

that is, that there is a true distribution of work. T_1 is the time a single CPU needs. For the communication, it turns

out that the main component is the latency t_{lat} of the communication. Latency refers to the amount of time that is necessary to prepare a message before it can be sent away. Most of that time is caused by the hardware, such as Ethernet, and the corresponding Internet protocols, such as TCP/IP. For 100 Mbit Ethernet, this latency time is of the order of 0.5 *msec*. Since in two-dimensional systems each domain has in the average six neighbors, and we need two messages per time step, communication time can be approximated as

> With 100 Mbit Ethernet, the best possible real time ratio of a parallel traffic simulation with a one-second time step is approximately 170.

$$T_{cmm}(p) \approx 12 t_{lat}.$$

Overall, this results in

$$T(p) \approx \frac{T_1}{p} + 12 t_{lat}.$$

Let us define the *real time ratio* as how much faster than reality the simulation is. If we assume that one simulation time step corresponds to one second, then we obtain

$$RTR = \frac{1\ sec}{T(p)} \approx \frac{1\ sec}{T_1/p + 12 t_{lat}} \quad \begin{matrix} p \to \infty \\ \to \end{matrix} \quad \frac{1\ sec}{12 t_{lat}}.$$

For 100 Mbit Ethernet, $t_{lat} \approx 0.5\ msec$, and therefore $RTR \to 167$ for $p \to \infty$. In words:
This statement is independent of the problem size or of the specific model, it only depends on the fact that a simulation time-step corresponds to one second.

The actual performance measurements in Figure 5 (left) from our implementation (dots) demonstrate that these concerns are justified. The line is a fit to those performance measurements, based on the mathematical form of Eq. (3).

The results of Figure 5 refer to runs of the Switzerland 6–9 scenario as explained above. Only with a faster communication hardware, Myrinet (http://www.myri.com), this bottleneck can be overcome and much faster simulations can be achieved. Our best performance currently is an RTR of nearly eight hundred. This means that a whole day of all car traffic in Switzerland, which has about 7 mio inhabitants, can be simulated in less than two minutes. This makes now large scale learning studies possible, although it should be noted that these performance numbers are obtained without taking data input and output into account.

Figure 5 (right) also shows the speedup for the same runs. Speedup is defined as the performance ratio between a multi-CPU and a single-CPU run:

$$Speedup(p) = \frac{T(p)}{T_1}.$$

As one can see in the plots, speedup is obtained by shifting the RTR curve vertically; the amount of shifting is given by the RTR of the single-CPU run. As one sees, our fastest simulation is, on sixty-four CPUs, about two hundred times faster than the single-CPU simulation. This so-called super-linear speedup is caused by the fact that the scenario is too large to fit comfortably into a single-CPU machine and thus causes slow memory paging there. Without further information, speedup curves cannot be used to understand real time limitations on the computation, which is important for ITS applications. More information can be found in Cetin and Nagel (2003).

Computational Performance of a Full Iteration

Figure 6 depicts the *cumulative* contributions of the major steps in the feedback system to the total execution time of each iteration. The left figure shows the implementation of the feedback mechanism with MySQL

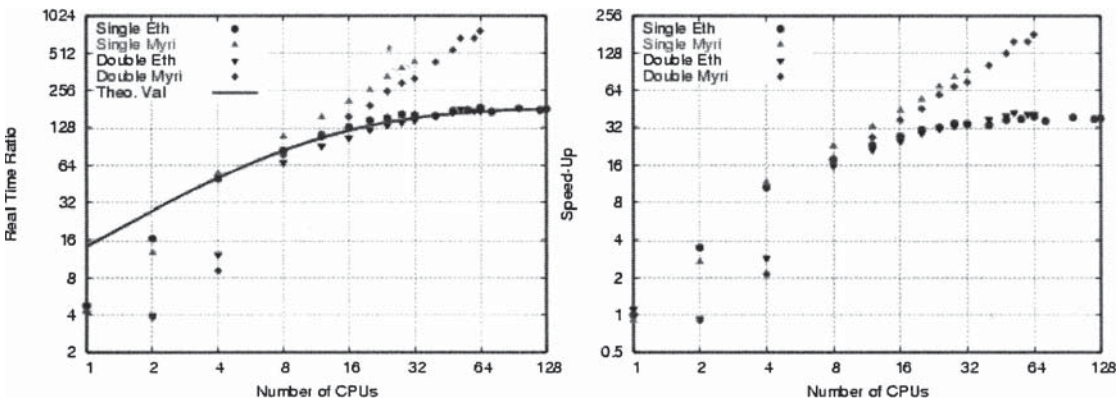

Figure 5 *(left)* RTR and *(right)* Speedup curves of the ch6-9 scenario from February 2003, with the default domain decomposition approach. Dots refer to actual computational benchmarks; the solid line is based on the theory explained in the text.

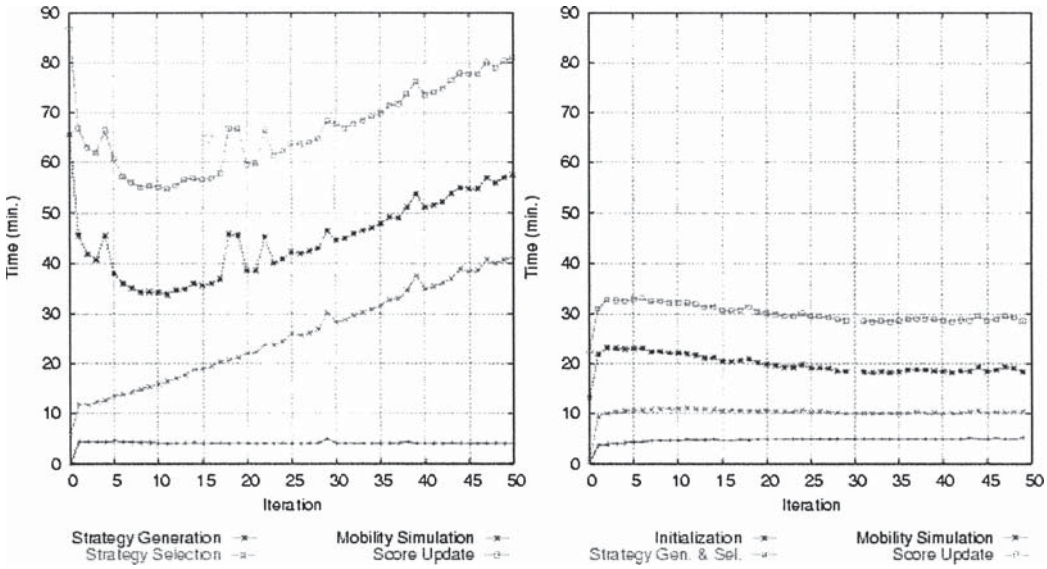

Figure 6 Cumulative Execution Time Contributions of Major Iteration Steps *(left)*: File-based database implementation. *(right)*: The new implementation, which keeps all agent information in computer memory, and which also uses a faster mobility simulation.

(www.mysql.org), a file-based database. As explained above, the agent database keeps track of agent strategies and their scores. The right figure shows an implementation 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, 2003), the new implementation is written in C++ and combines all operations into one program. The figure shows computational performance results for the scenario described above; the contributions shown in the figure 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. In the left figure, the 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. In the right figure it takes essentially the same amount of time in each iteration.
3. *Mobility Simulation*, where the agents interact. This again relaxes to a consistent amount of time, though takes longer in earlier iterations when there is more congestion to deal with. In the left figure, this step ends up taking about fifteen minutes while in the right it ends up taking only about ten minutes. This is due to an improvement in the mobility simuation speed between trials, and has nothing to do with the agent database.
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 above). 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. In the left figure, this operation takes about twenty minutes while in the right figure it takes about ten minutes.

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

One can see that in the left figure, on average each iteration takes about an hour to execute, with the feedback system, with strategy-related steps totaling about forty-five minutes of that time. The overall result is that we can run a metropolitan scenario with one million agents, including fifty learning iterations, in about two days.

For the right figure, the total time is cut in half, with about twenty minutes spent on strategy-related steps.

FUTURE PLANS

Activity Generation

The above results use traditional origin-destination tables for demand generation. We intend to move our

investigations to activity-based demand generation. Our plan for this is to start with a synthetic population, for example generated by Iterative Proportional Fitting (Beckman, Baggerly, and McKay, 1996). Next, by some method, activity patterns and primary activity locations are allocated to that synthetic population. This can, for example, be achieved by randomly drawing from census micro-sample data (Beckman, Baggerly, and McKay, 1996). With this, that is, a synthetic population with activity patterns and primary activity locations given, the iterations are started. One strategy module will generate locations for secondary activities, another strategy module will generate activity timings, yet another strategy module will generate routes, and the agent database will maintain different plans, and evaluate them via the mobility simulation. First results of this, with activity timing and routes inside the feedback loop, were already successful (Raney and Nagel, 2004; Balmer, Raney, and Nagel, 2004), and a prototype for location choice also exists (Marchal and Nagel, 2004).

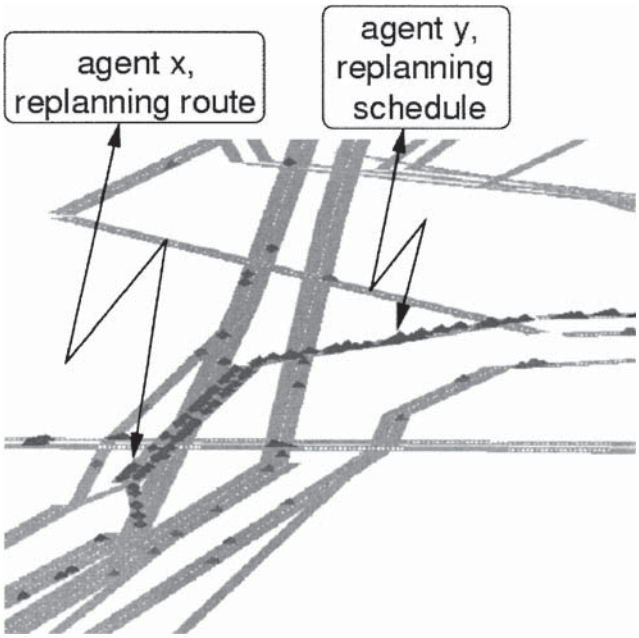

Figure 7 Virtual Reality Representation of Simulated Traffic in Portland/Oregon. Including visualization of message-based within-day replanning.

Message-Based Modules

It was explained earlier that a computational architecture for real-world multi-agent simulations should consist of at least two conceptual parts: the module for the simulation of the physical system, and the strategy generation module(s). As long as one remains within the framework of day-to-day learning, these modules are called sequentially, and it is possible to exchange information between them by a slow technology, for example, via files. For ITS however, it is necessary to include within-day replanning into the simulation system. This implies that the simulation of the physical system needs to remain in permanent contact with the module(s) that compute(s) the strategies.

An implementation that achieves this and also maintains computational efficiency is to keep the strategy modules completely separate from the physical simulation, and to couple them via *messages*. More specifically, the strategy modules would *send* the agent strategies to the mobility simulation, which would attempt to execute them. The mobility simulation would *send* the events back to the strategy modules. In intuitive sketch of this is Figure 7. Further details can be found in (Nagel and Marchal, 2003) and (Gloor et al., 2003).

SUMMARY

This paper describes aspects of large-scale MASim for traffic simulations in general, and the evaluation of ITS systems in particular. A first important starting point is that such simulation systems consist of two components: the simulation of the physical system (mobility simulation), and the simulation of the strategy generation. Both modules are rather different, and thus need rather different techniques in terms of modeling and implementation.

It is important to note that ITS devices can be treated as agents in a similar way as the travelers. A strategy for an ITS device could for example be a message on a variable message sign, or a signal timing plan, and both could be computed by a traffic management center and then sent to that device for execution in the mobility simulation.

Besides the two components "mobility simulation" and "strategy generation," there also needs to be an implementation of agent learning. Single-agent learning can be understood from a behavioral perspective, treated elsewhere in this workshop, or from a computer science perspective, where it touches upon Artificial Intelligence and Machine Learning. Importantly, when several agents learn together, then the whole learning system can be described as a dynamical system. As is well known, dynamical systems go toward attractors, which can be fixed points, periodic, or chaotic. There is nothing in our knowledge that tells us where a simulation of a learning traffic system will go, or where the real system will go. In addition, issues of "learning speed" matter, in particular when there are several entities (such as users and Traffic Management Center) which learn simultaneously.

MASim looks like the perfect technology to evaluate ITS systems, since is is possible to implement individual behavioral rules for each individual traveler, thus allowing for a differentiated and segmented response of the traveler population. Also, traffic management operations can be included into the simulation framework in a straightforward way: each variable entity, such as a traffic signal or a variable message sign, just becomes a "technical" agent by itself, that follows a plan given to it by the traffic management center.

In order to explore large systems, significant computational speed is necessary. Issues of computational speed are discussed, demonstrating that it is possible to study several millions of travelers with computation times on the order of a day.

Unfortunately, the technology that enables such agent-based large scale simulations does not allow within-day replanning, that is, the capability of the agents to change their plan while en-route. This is, however, clearly necessary for the evaluation of ITS. Therefore, future plans include coupling the different simulation modules via messages. This would not only further increase computational speed, but it would also allow the direct implementation of within-day learning.

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

Arentze, T. and Timmermans, H.J.P. (2003). Representing mental maps and cognitive learning in micro-simulation models of activity-travel choice dynamics, Meeting of the International Association for Travel Behavior Research (IATBR). See http://www.ivt.baum.ethz.ch/allgemein/iatbr2003.html (accessed September 20, 2004).

Balmer, M., Raney, B., and Nagel, K. (2003). Agent-based activities planning for an iterative traffic simulation of Switzerland—activity time allocation, Proceedings of Swiss Transport Research Conference (STRC), 2003, Monte Verita, Switzerland. See http://www.strc.ch (accessed September 20, 2004).

Bando, M., Hasebe, K., Nakayama, A., Shibata, A., and Sugiyama, Y. (1995). Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E*, **51**(2), 1035–1042.

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

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

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

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

Bundesamt für Strassen. (2000). Automatische Strassenverkehrszählung 1999. Bern, Switzerland. See http://www.statistik.admin.ch/news/archiv96/dp96036.htm (accessed September 20, 2004).

Bundesamt für Statistik und Dienst für Gesamtverkehrsfragen. (1996). Verkehrsverhalten in der Schweiz 1994. Mikrozensus Verkehr 1994, Bern, Switzerland.

Cantarella, C. and Cascetta, E. (1995). Dynamic process and equilibrium in transportation network: Towards a unifying theory, *Transportation Science A*, **25**(4), 305–329.

Cascetta, E. (2001). *Transportation systems engineering: theory and methods*. Dordrecht, Holland: Kluwer Academic Publishers.

Cascetta, E. and Cantarella, C. (1991). A day-to-day and within day dynamic stochastic assignment model. *Transportation Research A*, **25A**(5), 277–291.

Cayford, R., Lin, W.-H., and Daganzo, C.F. (1997). *The NETCELL simulation package: Technical description. California PATH Research Report UCB–ITS–PRR–97–23*. Berkeley, CA: University of California.

Cetin, N. and Nagel, K. (2003). A large-scale agent-based traffic microsimulation based on queue model, Proceedings of Swiss Transport Research Conference (STRC), Monte Verita, Switzerland. See http://www.strc.ch (accessed September 20, 2004).

Charypar, D. and Nagel, K. (2003). Generating complete all-day activity plans with genetic algorithms, Meeting of the International Association for Travel Behavior Research (IATBR), Lucrene, Switzerland.

Charypar, D., Graf, P., and Nagel, K. (2004). Q-learning for flexible learning of daily activity plans. In *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, Switzerland.

Chowdhury, D., Santen, L., and Schadschneider, A. (2000). Statistical physics of vehicular traffic and some related systems. *Physics Reports*, **329**(4–6), 199–329.

Daganzo, C.F. (1998). Queue spillovers in transportation networks with a route choice. *Transportation Science*, **32**(1), 3–11.

DYNAMIT, (2003). www.its.mit.edu and dynamictrafficassignment.org (accessed September 20, 2004).

DYNASMART, (2003). www.dynasmart.com and dynamictraffiicassignment.org (accessed September 20, 2004).

Esser, J. and Nagel, K. (2001). Iterative demand generation for transportation simulations. In *The Leading Edge of Travel Behavior Research*, Amsterdam: Pergamon.

Ferber, J. (1999). *Multi-agent systems. An Introduction to distributed artificial intelligence*. Harlow, UK: Addison-Wesley.

Friedrich, M., Hofsäß, I., Nökel, K., and Vortisch, P. (2000). A dynamic traffic assignment method for planning and telematic

applications. Proceedings of Seminar K, Cambridge, GB: European Transport Conference.

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.

Gerlough, D.L. (1956). Simulation of freeway traffic by an electronic computer, F. Burggraf and E.M. Ward, editors, Proc. 35th Annual Meeting, Washington, D.C.: Highway Research Board, National Research Council, 543–547.

Gingold, R.A. and Monaghan, J.J. (1989). Smoothed particle hydrodynamics—theory and application to non-spherical stars, Royal Astronomical Society, Monthly Notices, 181, 1977.

Gloor, C., Cavens, D., Lange, E., Nagel, K., and Schmid, W. (2003). A pedestrian simulation for very large scale applications, A. Koch and P. Mandl, editors, Multi-Agenten-Systeme in der Geographie, number 23 in Klagenfurter Geographische Schriften, pages 167–188.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Herman, R., Montroll, E.W., Potts, R.B., and Rothery, R.W. (1959). Traffic dynamics: Analysis of stability in car following, *Operations Research*, **7**, 86–106.

Hofbauer, J. and Sigmund, K. (1998). Evolutionary games and replicator dynamics. Cambridge University Press.

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

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

Kistler, D. (2004). Mental maps for mobility simulations of agents. Master's thesis, ETH Zurich.

Krauß, S. (1997). Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics, PhD thesis, University of Cologne, Germany.

Marchal, F. and Nagel, K. Modelling location choice in activity-based models with cooperative agents, Proceedings of Swiss Transport Research Conference (STRC), Monte Verita, Switzerland. See http://www.strc.ch (accessed September 20, 2004).

MITSIMLab, (2004). See http://www.mit.edu/its/mitsimlab.html (accessed September 20, 2004).

Nagel, K. (1994/95). High-speed microsimulations of traffic flow, PhD thesis, University of Cologne.

Nagel, K. and Marchal, F. (2003). Computational methods for multi-agent simulations of travel behavior. Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR), Lucerne, Switzerland. See http://www.ivt.baum.ethz.ch/allgemein/iatbr2003.html (accessed September 20, 2004).

Nagel, K. and Schreckenberg, M. (1992). A cellular automaton model for freeway traffic, *Journal de Physique I France*, **2**, 2221–2229.

Nagel, K., Strauss, M., and Shubik, M. The importance of timescales: Simple models for economic markets. *Physica A*, submitted.

Newell, G.F. (1961). Nonlinear effects in the dynamics of car following, *Operations Research*, **9**(2), 209–229.

Ortúzar, J.deD. and Willumsen, L.G. (1995). Modelling transport. Chichester, UK: Wiley.

PTV, (2003). www.ptv.de (accessed September 20, 2004).

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

Raney, B. and Nagel, K. (2002). Iterative route planning for modular transportation simulation, Proceedings of Swiss Transport Research Conference (STRC), Monte Verita, CH.

Raney, B. and Nagel, K. (2003). Truly agent-based strategy selection for transportation simulations, Paper 03–4258, Washington, D.C.: Transportation Research Board Annual Meeting.

Raney, B. and Nagel, K. An improved framework for large-scale multi-agent simulations of travel behavior, Proceedings of Swiss Transport Research Conference (STRC), Monte Verita, Switzerland. See http://www.strc.ch (accessed September 20, 2004).

Russel, S.J. and Norvig, P. (1995). Artificial intelligence: A modern approach, Series in Artificial Intelligence. Engelwood Cliffs, NJ: Prentice Hall.

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

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

Simro, H.P. and Powell, W.B. (1992). Numerical methods for simulating transient, stochastic queueing networks, *Transportation Science*, **26**, 296.

Stein, D.L. and Nadel, L., editors. *Lectures in the sciences of complexity, Santa Fe Institute in the sciences of complexity*, Redwood City, CA: Addison-Wesley.

Vrtic, M. and Axhausen, K.W. (2002). Experiment mit einem dynamischen umlegungsverfahren, *Strassenverkehrstechnik*.

Vrtic, M., Koblo, R., and Vödisch, M. (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.

Weiss, G. editor. (1999). Multiagent Systems. A modern approach to distributed artificial intelligence. Cambridge, MA: The MIT Press.

Wiedemann, R. (1994). Simulation des Straßenverkehrsflusses. Schriftenreihe Heft 8, Institute for Transportation Science, Germany: University of Karlsruhe.

Van Zuylen, H.J. and Taale, H. (2004). Urban networks with ring roads: A two-level, three player game. Paper 04-001659, Washington, D.C.: Transportation Research Board Annual Meeting.