Michael Balmer, Nurhan Cetin, Kai Nagel, Bryan Raney

Towards Truly Agent-Based Traffic and Mobility Simulations

Balmer, M., Cetin, N., Nagel, K., Raney, B. (2004). Towards truly agent-based traffic and mobility simulations. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004. Volume 1. New York: ACM. ISBN: 1-58113-864-4. pp. 60–67.

Towards truly agent-based traffic and mobility simulations

Michael Balmer[a], Nurhan Cetin[b], Kai Nagel[c], and Bryan Raney[b]
[a] Inst. for Transport Planning and Systems, ETH Zurich, Switzerland
[b] Inst. of Computational Science, ETH Zurich, Switzerland
[c] Inst. for Land and Sea Transport Systems, TU Berlin, Germany
{balmermi,cetin}@inf.ethz.ch, nagel@vsp.tu-berlin.de, braney@inf.ethz.ch

Abstract

Traveling is necessary and desirable; yet, it imposes external costs on other people. Quantitative methods help finding a balance. Multi-agent simulations seem an obvious possibility here. A real world traffic simulation consists of many modules, all requiring different expertise. The paper discusses how such modules can be coupled to a complete simulation system, how such a system can be made fast enough to deal with real-world sizes (several millions of travelers), and how agent memory can be introduced. A real-world case study is presented, which says that multi-agent methods for traffic are mature enough to be used alongside existing methods. Finally, some outlook into the near future is given.

1. Introduction

Traffic, transportation, and mobility are important for most societies. Yet, there needs to be a balance between mobility and the external costs this imposes. In order to help with finding this balance, increasingly sophisticated tools are used.

The state of the art is a procedure called the 4-step-process (e.g. [1]). Somewhat loosely speaking, it computes traffic flows in a similar way as flows of electrons are computed in electric networks, with the distinction that there are many classes of travelers, according to their different destinations. A major shortcoming of the standard implementation of the 4-step-process is that it completely severs the connection to the individual travelers. In the 4-step-process, all travelers going to the same destination are assumed to be equal.

Clearly, this could be improved by keeping track of individuals, i.e. by making the approach agent-based. It would consist of two layers [2] (Fig. 1)

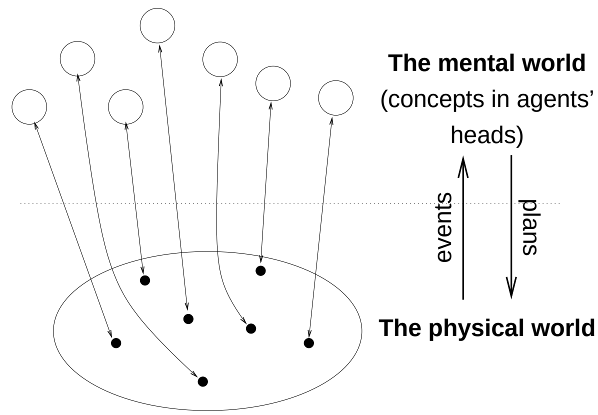

Figure 1. The two layers of a mobility simulation system

- The **physical layer** (Sec. 2), which simulates the physical world where agents move, avoid each other, go around obstacles, generate congestion, etc.
- The **mental layer** (Sec. 3), in which the agents generate strategies, such as routes, mode choice, daily activity plans, etc.

In addition, one needs a **feedback mechanism** (Sec. 4), which mirrors the fact that travelers in the real world learn from one day to the next; and that feedback mechanism needs initial conditions (Sec. 5). The paper will continue with a description of a specific case study, run for the morning traffic of all of Switzerland (1 million agents; Sec. 6), and an outlook on the near future (Sec. 7). A summary concludes the paper.

2. The physical layer: Mobility simulation

The simulation of the physical system, i.e. the real-world traffic system, can be started with relatively standard computational methods, such as molecular dynamics or cellular automata.

An important element is that, at a certain level of abstraction, traffic moves on a **network** rather than on flat 2d (or 3d) space. This implies that a traffic simulation consists of dynamics on the links (roads), and dynamics on the nodes (intersections). All these are far from simple, as can be imagined when thinking of the infamous "traffic jam out of nowhere" (which is hotly debated in the literature), or of complicated adaptive traffic signal schemes.

For a large metropolitan area, several millions of particles (travelers) need to be simulated simultaneously. This results in a significant computational challenge: 10^9 particles are feasible in state-of-the-art large scale molecular dynamics codes, but for traffic one needs to add computer time for the internal intelligence of the particles. A typical traffic simulation can compute about 10^6 vehicle-seconds during one second of a typical 1-GHz CPU. This means that a simulation with 10 million travelers progresses 10 times slower than real time, which is unacceptably slow for real-world applications. Fortunately, with parallel computing, near-linear speed-up can be obtained, and on a cluster with 100 CPUs and Myrinet communication such a simulation runs 10 times *faster* than real time [3].

In addition, the traffic dynamics can be significantly simplified without making the model too unrealistic [4], resulting in another factor of 10. With this, it is for example possible to simulate a whole day of all car traffic of the whole country of Switzerland (approx 7 mio inhabitants) in less than 5 minutes [3].

This simplified approach leaves many aspects unresolved, for example: inclusion of other modes of transportation, in particular by coupling different simulations instead of having to re-implement everything into one common computational code; complicated adaptive signals or message signs; heterogeneous vehicle fleets; complicated intersection/weaving design; etc. Nevertheless, it allows us to move on to the mental or strategy layer within a consistent, agent-based framework, and improvements to the mobility simulation can be achieved over time.

3. Strategy generation

The next layer, quite obviously, concerns the agents' strategies. The currently typical approach to this looks at *daily* plans of travelers, and decomposes the problem into two parts: activity generation and mode/route choice. In the **activity generation** module, for each agent in the simulation a complete 24-hour activity plan is generated, including locations and times of activities. Typical examples of activities are: "home", work, shop, leisure, drop off child at kindergarten, etc. In the **mode/route choice** module, activities at different locations are connected by trips, including the choice of the mode of transportation (e.g. bicycle, walking, car, etc.), and specific routes. An example XML description of the result from these two modules is in Fig. 2.

A plausible solution for route choice is simple Dijkstra shortest path algorithm [5], using travel time as generalized costs on the links. If one makes those link travel times dependent on the time-of-day, then a time-dependent algorithm is capable of routing vehicles around congestion if needed [6]. Such an algorithm, in particular when used together with a stochastic simulation (from which one obtains the link travel times) produces sets of routes which contain approximately 70% of the routes that are used by real-world people, and pieces of most of the remaining routes [7]. Although there is certainly room for improvement, this is not the most pressing issue at this stage.

Public transit could in principle be routed with the algorithms that are available on many public transit www servers these days, which often include even the walking parts to or from a street address. To our knowledge, this approach has not yet been attempted in the area of multi-agent traffic simulation.

Activity generation is a very active field of research, and many methods are tested. The maybe most mature method is discrete choice theory [8], which is based on microeconomic utility maximization, and is also used in marketing research. It couples agents' decisions to attributes of the person and to attributes of the alternatives. For example, the utility of using the bus to work could be $U_{bus} = -\alpha\, T_{bus} - \gamma\, C_{bus} + \mu\, G$, where T_{bus} is the time on the bus, C_{bus} is the monetary cost of taking the bus, G is the gender (e.g. 1 when male and 0 when female), and α, γ, and μ are weight factors that need to be estimated, e.g. by a maximum likelihood procedure. Similarly, the utility of using the car to work could be $U_{car} = -\alpha'\, T_{car} - \gamma'\, C_{car} + \mu'\, G$. Finally, the decision would be made according to probability

$$p_{bus} = \frac{e^{\beta\, U_{bus}}}{e^{\beta\, U_{bus}} + e^{\beta\, U_{car}}} \; .$$

For the complete activity generation problem, actual implementations proceed hierarchically, by first making a decision about the overall pattern, than about locations, and then about times [9]. Certain mathematical requirements ensure that the method is consistent, i.e. that the higher level decision is based on correct assumptions about the lower levels. For example, one should have some ideas about shop locations and travel times before making a decision about a certain shopping trip.

Alternative approaches use Monte Carlo chains [10], generic rules [11], genetic algorithms [12], or mental maps [13]. Those methods are maybe more intuitive, but also more difficult to calibrate. Nevertheless, there are aspects of human strategies that the current versions of discrete choice

```
<person id="241" income="50000">
  <plan score="123">
    <act type="h" end_time="07:00" x="7150"
         y="2790" link="5834" />
    <leg mode="car" dp_time="07:00" trav_time="00:25">
      <route>1932 1933 1934 1947</route>
    </leg>
    <act type="w" dur="09:00" x="0650"
         y="3980" link="5844" />
    <leg mode="car" dp_time="16:25" trav_time="00:14">
      <route>1934 1933</route>
    </leg>
    <act type="h" x="7150" y="2790" link="5834" />
  </plan>
</person>
```

Figure 2. Agent plan, expressed in XML

theory do not include, for example the memory aspects of mental maps. It is unclear which methods for activity generation will be most useful for what types of investigations.

One does not have to stop at activity generation, but can go to strategy levels even further removed from the traffic itself. Examples are residential choice, life-style decisions, etc. [14].

With regard to computational performance, our experience is that, even for large networks, route calculations are relatively fast, and one can generate several hundred routes per second [6], implying a running time of less than three hours for a population of a million. With activity generation, we have less practical experience; treating several hundred agents per second seems to be possible in some cases, but more realistic models need more computer time. Fortunately, basic strategy generation modules are easy to parallelize, since agents do not interact in current implementations. With 100 CPUs, one can afford about 1 second of computer time per agent per iteration for a strategy generation computation.

4. Learning and feedback

4.1. Introduction

Travelers in the real world do similar things over and over again. For a simulation of these processes this implies that a period is selected (e.g. a day, or a full week), and that period is run over and over again, as follows:

- There is some initial condition on which the mobility simulation is run.
- The mental modules are run and compute strategies (plans) based on the result of the mobility simulation.
- The mobility simulation is run again, based on the new plans.
- Etc., until some stopping criterion is fulfilled.

This models **period-to-period replanning**. In reality, people can revise their plans at all points in time, implying **within-period** replanning. In terms of modeling, within-period replanning implies that agents are able to revise their plans while the mobility simulation is running.

4.2. The currently typical approach

The currently typical approach in the traffic community is that it is not the agents that learn, but the system. This is typically achieved by the mobility simulation feeding back *system* performance information rather than agent information. For example, the mobility simulation outputs time-averaged link travel times. From this, the router can compute time-dependent shortest paths (see Sec. 3), and the activity generator can compute point-to-point travel times. The iterations proceed by giving some or all agents new plans. Importantly, those agents *completely forget their previous plans*.

This approach works because the assumption is that the system goes to a Nash Equilibrium (NE). As is well known, at a NE, no agent can improve by unilaterally switching strategies; translated into our system, this means that a NE is reached when all agents always obtain the same strategy as before from the mental modules from one iteration to the next. No agent memory is needed for this approach.

Systems with within-day replanning use a similar approach: If, say, the routes to a certain destination are changed because of congestion, then *all* travelers to that destination from then on need to follow the new routes [15, 16, 17]. In implementations, one uses shortest path trees to each possible destination, which are relatively cheap to compute, and which contain, for each intersection, the next link that needs to be taken to a given destination. Different user classes are possible, but if one introduces as many user classes as there are agents, this approach fails because a full shortest path tree needs considerably more memory than just a route. The same argument implies that destinations need to be spatially aggregated, because again shortest path trees only pay off if many agents travel to the same destination.

In addition, there is a tendency to pack routing and “network loading” (the equivalent of the mobility simulation) into one package, called dynamic traffic assignment (DTA). The input to this is an OD matrix, while the output is link performance information [15, 16, 17]. *This makes truly agent-based approaches impossible since the anonymous OD matrix severs the connection to the agent.*

In consequence, in the literature one finds approaches using mental maps that give OD matrices rather than agent plans to the mobility simulation, and in consequence obtain average link travel times from the mobility simulation rather than individual agent information [14, 18]. This does not seem to be a good approach in the long run.

The reason for this is probably historical: Step 4 of the 4-step-process (mentioned in the introduction) has, under some circumstances, a number of provable mathematical properties, including some uniqueness properties of the solution [19]. This makes different implementations easy to compare, which is an important advantage. However, once traffic assignment is made time-dependent (*Dynamic* Traffic Assignment, DTA), those mathematical properties break down [20], and this is no longer an argument.

4.3. A fully agent-based approach

Two shortcomings of the approach outlined in Sec. 4.2 are: (i) The feedback is based on aggregated system information, not on specific agent information. For example, the spatial aggregation that is typically used with respect to destinations does not permit to, say, recognize the difference between a 10 meter and a 500 meter walk to the bus stop. Nevertheless, this will cause considerable differences in the real world. (ii) If one is not interested in the Nash Equilibrium, be it because one is interested in the transients or because one does not believe that humans indeed go to a Nash Equilibrium, then the approach fails because there is no access to human behavior. For example, the building of an individual mental map is difficult, because the mobility simulation only feeds back aggregate link travel times, not individual agent performance.

An alternative is to revise the whole approach so that it becomes entirely agent-based. This is essentially a straightforward operation, except that it implies the following changes in most existing packages:

- Agent information needs to be maintained *throughout the simulation system*. Instead of having OD matrices at some intermediate step, there need to be individual agents with individual origins and destinations, plus departure time.

- The DTA needs to be separated into its two constituents "routing" and "network loading". The routing needs to read the individual agents that have individual origins and destinations, and add routes. The network loading (now called mobility simulation) needs to read all those agent plans, and execute them simultaneously.

  By doing this, the routing module becomes a strategy module, completely analogous to activity generation or residential choice.

- Finally, the mobility simulation needs to output agent performance information instead of system performance information. This is, in our view, most easily achieved by emitting "events", e.g. "agent arrives at/leaves an activity", "agent enters/exits link", etc. By this, all data aggregation can be done by each individual module. For example, a standard router would aggregate link enter/exit times into the typical aggregated link travel times, while a mental map would concentrate on all information by a specific agent or subset of agents.

One useful module of such an approach is what we call the **agent database**. There, each agent remembers several plans, plus performance information related to those plans. In some fraction of the iterations, the agent receives and executes a new plan ("exploration"), in all other iterations, it choses between the known plans according to their performance ("exploitation"). This is similar to any kind of classifier system or genetic algorithm, with the only caveat that in our simulations not too many agent can simultaneously explore since then the performance information does not reflect what they would encounter later when all agents exploit.

4.4. Computational aspects

In order to execute iterations and feedback, all the different modules need to be coupled. As long as one uses period-to-period replanning only, this is straightforward. Let us assume a simple case with an activity generator, a router, and a mobility simulation. Starting from some initial condition (see below), the mobility simulation runs for the first time, and records agent performance information. Then, the activity generator reads that performance information, and goes through all agents in order to adapt their activity plans. Next, the router reads the same performance information, and then goes through all agents in order to adapt their routes, possibly reacting to different locations generated by the activity generator. The resulting plans are again executed by the mobility simulation, etc.

A big advantage of this approach is that modules can run stand-alone, and can be coupled via files. Albeit old-fashioned, this has the advantage that the expertise of different research groups can be combined, which is close to impossible with all other current approaches (see below). A little bit of better technology can be introduced by using XML files, which lend themselves well to agent plans (Fig. 2). The file-based approach even leaves the parallelization of the individual modules intact.

Once within-day replanning is desired, things get considerably more complicated [21]. The only well-established technology for this seem to be subroutine calls. For such an implementation, first the mobility simulation is started. If an agent wants to re-plan, it interrupts the mobility simulation and generates the new plan, after which the mobility simulation proceeds. A requisite of this approach is that all modules are written in the same programming language. For a parallel implementation, the following additional issues arise: (i) Because of load balancing reasons, agents cannot re-plan whenever they want, but need to be tied to regular intervals. Otherwise, in any given time step *some* CPU will contain a replanning agent, and all other CPUs wait. (ii) Compared to the parallel implementation of the mobility simulation alone, considerably more information needs to be exchanged between the CPUs, leading to bandwidth problems.

A third technique, promising to resolve these issues, is to use messages to exchange information between agents. A possible approach to this is to separate the mobility simulation from the mental simulation, somewhat analogous to robosoccer where the robots have some low-level intelligence of their own, but all more complicated computations are done on external computers which are connected to the robots via wireless. In our situation, the real-world robots would be replaced by another simulation, i.e. the simulation of the physical system. Reports of such an approach are submitted separately for this conference; the overall summary is that (1) such an approach is rather experimental, and (2) at this point it is difficult to use it for large scale simulations, because once more the bandwidth of the computer network is not sufficient. In particular, it is difficult to use the higher performance Myrinet with that message-based approach, because Myrinet does not use TCP/IP.

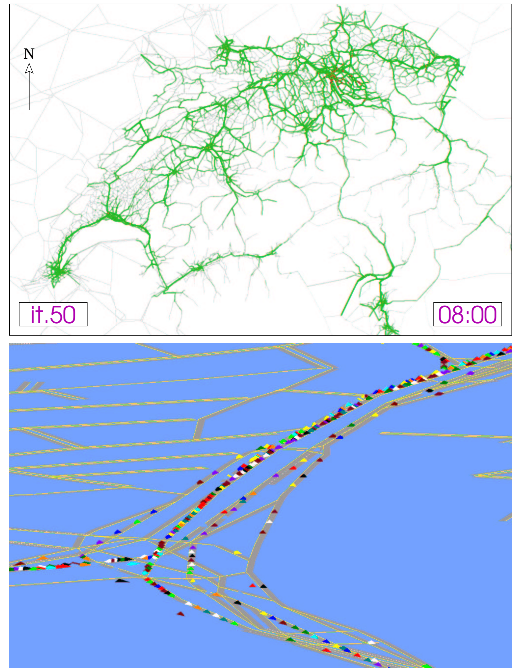

Figure 3. Switzerland at 8:00 AM. TOP: The full network. Each car is represented by a dot, which is not visible at this scale. BOTTOM: Detail.

5. Initial conditions

The iterations somehow need to be started. In particular, the population of agents needs to be generated. This can be achieved via a **synthetic population generation module**, which essentially takes census information as input, and generates synthetic individuals as output. Individuals come with attributes such as age, gender, income, household membership, car ownership, etc.

Similarly, initial daily activity plans can often be obtained from the micro-census or from specific surveys. From such data, activity chains for a small fraction of the population are known, and one can generate plausible activity chains for the whole population by assuming that people with similar characteristics will have similar activity chains.

Initial routes can be generated by just giving everybody the route that would be fastest on an empty network, plus the fastest public transit option as an alternative.

If one is interested in the steady-state behavior (i.e. what happens to the Nash Equilibrium if the system is stochastic and individual agents no longer necessarily find the optimum, but just some “good” solution), then initial conditions do not matter very much, except that bad initial conditions lead to long transients and therefore to long computational running times. If one is interested in the transient learning process itself, the initial conditions matter much more.

Note that even with initial conditions coming from a census-based synthetic population, it is possible to include land use modules (e.g. residential choice) into the computational iterations in order to predict the development of urban areas many years into the future.

6. A real world case study

In this section, it will be shown that the concepts outlined above can indeed be put into practice, and that it is already now possible to use them alongside the existing methods. That study was run for the area of all of Switzerland, represented by a road network with 28 622 links and 10 564 nodes (Fig. 3). The goal was to compute the morning rush hour traffic from standard hourly OD matrices, as they are used for conventional models.

Our starting point for demand generation for the full Switzerland scenario is a 24-hour origin-destination matrix from the Swiss regional planning authority (Bundesamt für Raumentwicklung). That matrix is converted into 24 one-hour matrices using a three step heuristic. The first step employs departure time probabilities by population size of origin zone, population size of destination zone and network distance. These are calculated using the 1994 Swiss National Travel Survey [22]. The resulting 24 initial matrices are then corrected (calibrated) against available hourly counts using the OD-matrix estimation module of VISUM [23], which is a state-of-the-art traditional assignment package. Hourly counts are available from the counting stations on the national motorway system [24]. Finally, the hourly matrices are rescaled so that the totals over 24 hours match the original 24h matrix. Those resulting hourly matrices are then used as input to a VISUM assignment, which is a variant of the traditional assignment models discussed earlier. These assignment results are the base method against which our multi-agent simulation will be compared.

The hourly matrices are also used as input to our multi-agent simulation. For this, they are immediately disaggregated into individual trips, or more correctly into individual agents with one trip each. That is, we generate individual trips such that summing up the trips would again result in the given OD matrix. The starting time for each trip is randomly selected between the starting and the ending time of the validity of the OD matrix.

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

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

This leads to a list of approximately 5 million trips, or about 1 million trips between 6am and 9am. The resulting plans files look like Fig. 2, except that the leg and route information is not yet there. Since the origin-destination matrices are given on an hourly basis, these trips reflect the daily dynamics. Intra-zonal trips are not included in those matrices, as by tradition.

The router then computes routes based on free speed travel times for all agents; the resulting plans files now look exactly like Fig. 2. Those complete plans are fed into the mobility simulation, which executes all the plans simultaneously, and generates events information. The particular simulation used was a queue simulation [4, 3], which essentially moves vehicles forward along a link according to free speed, and adds them to a queue at the end of the link. The queue is served according to the so-called capacity of the link, which comes from the network files. Vehicles can only move if there is space on the following link, which is the main difference to standard queuing theory.

Based on the events, the agent database updates the scores for each individual plan, while the router computes time-dependent link travel times. Then, 10% of the agents obtain new plans (= routes), which are added as plans to each agent's internal memory. The mobility simulation runs again, where those 10% of agents use the new plans, while the other 90% use their previous plans.

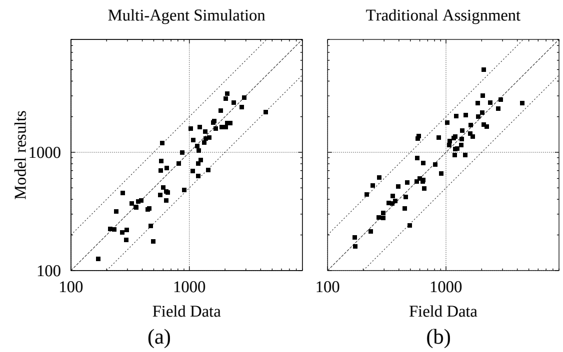

Figure 4. *(a)* Simulation (y-axis) vs. field (x-axis) data for the 50th iteration. *(b)* VISUM assignment vs. field data. After [26].

	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%

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

This is iterated many times. When, in later iterations, agents have a choice between different plans (routes), and they were not selected for the exploration of a completely new route, then they choose between existing routes with a probability of $\exp[\beta\, U_i]$, where U_i is the utility of the ith option, computed as the negative of the travel time. For β, $1/360\, sec$ is used, which is plausible from estimations of discrete choice models [25].

Fig. 4 shows a comparison between the simulation output corresponding to Fig. 3 and field data taken at counting stations throughout Switzerland [24]. The dotted lines outline a region where the simulation data falls within 50% and 200% of the field data. We consider this an acceptable region at this stage since results from traditional assignment models that we are aware of are no better than this (Fig. 4(b); see also [27]).

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

For example, the "mean relative bias" means that our multi-agent simulation underestimates flows by about 5%, whereas the VISUM assignment overestimates them by 16%. The average relative error between the field measurement and the simulation is 25%, between the VISUM assignment and reality 30%. These numbers state that the simulation result is better than the VISUM assignment result.

What makes our result even stronger is the following aspect: As explained earlier, the OD matrices were actually modified by a VISUM module to make the assignment result match the counts data as well as possible. These OD matrices were then fed into the simulation, without further adaptation. It is surprising that even under these conditions, which seem rather advantageous for the VISUM assignment, the simulation generates a smaller mean error. More details in this study can be found in [26].

It is admittedly rather difficult to judge the quality of our multi-agent simulation for traffic based on these results. However, it seems that we consistently obtain results in the same error range: with a simulation study in Portland/Oregon we obtained a mean relative bias of -20% and a mean relative error of 36% (which was slightly worse than the traditional assignment result) [27]; with a study in the Zurich area but now also generating the time structure of the agents' activities self-consistently, we obtained $+9\%$ for the bias and 30% for the error (no comparison to assignment available) [28]. In our view, this allows the tentative conclusion that our multi-agent traffic simulation is already about as good as existing assignment models. In addition, there are still many years of fine-tuning ahead of us; in addition, the multi-agent approach allows the inclusion of sensitivities to many aspects of interest that assignment has difficulties picking up at all (e.g. sensitivity to exact locations of public transit stops; changes in the peak period time structure; reactions to ITS devices; etc.).

7. The near future

As pointed out above, we have already included time replanning in addition to route replanning. That is, agents can change how long they want to stay at home and how long they want to work. We will start with car traffic only, then add public transit, and eventually add pedestrians and bicycles. The prototypes for all this exist already, but need to be integrated into the system.

On the strategy side, as a next step, so-called secondary activities (e.g. shopping, leisure) will be added. This includes finding locations for them, and (once more) generating the daily time schedules. Again, prototypes exist, but need to be integrated.

It would also be interesting to make residences and/or workplaces mobile. For this, other agent-based projects exist, most notably ILUTE [14]. It seems straightforward to couple ILUTE to our own simulation package so that a fully agent-based package for the integrated modeling of transport and land use will become available.

Finally, there is the usual issue of calibration and validation. Calibrating models with emergent properties is a difficult subject, which will need more research. In terms of validation, i.e. comparison of simulation results with field measurements, the main problem is consistent data availability, i.e. data for initial conditions (census, OD matrix), boundary conditions (high resolution road network), and field measurements (e.g. link volumes or link travel times, registered to the same network). This needs a longterm collaboration between academic groups and regional administrations, a feat that is not always easy to achieve.

8. Summary

This paper gives an overview over the state-of-the-art in multi-agent simulation of traffic. A multi-agent simulation of traffic consists of the physical layer (mobility simulation), which simulates the physical system, and the mental layer. The latter consists of several modules, most importantly activity generation, and mode/route choice. A considerable challenge is the size of the problem, consisting of several millions of agents. Within the mobility simulation, they can be solved using relatively standard methods from particle simulations including parallel computing. For the mental modules, parallel computing is straightforward as long as agents do not interact on that level, as is the case for simple versions as are currently used. A result is that about one second of computer time per agent can be afforded to compute complete agent strategies – that implies relatively simple models when compared to some other multi-agent simulations. Agents learn by living through the same period (e.g. day, week) over and over again.

Implementation and interoperability issues get considerably simplified when agents are restricted to *only* be able to revise their strategies between iterations – then simple files, e.g. XML, are sufficient to couple modules, which can in consequence be exchanged as stand-alone executables. When within-day replanning is to be included, the situation becomes considerably more complicated. Adding replanning (i.e. strategy computation) via subroutine calls from the mobility simulation is possible, but (a) restricts one to using the same programming language for all modules, and (b) causes problems with parallel implementations because of load balancing and message bandwidth limitations. An approach completely based on messages promises to remove these problems, but is as of now experimental and also bandwidth-limited.

Finally, a case study is presented, which demonstrates that despite their young age, multi-agent simulations already seem to be at par with the established quantitative methods, called static assignment. In the near future, considerably more modules will be integrated into the system, giving it power far beyond what is currently existing.

Acknowledgments – The Swiss Federal Administration (ARE) made the Switzerland network and demand data available to the Institute of Transport Planning and Systems (IVT), which made it available to us. PTV AG made the Navtech network of the Zurich area available to us.

References

[1] J. de D. Ortúzar and L.G. Willumsen. *Modelling transport*. Wiley, Chichester, 1995.

[2] J. Ferber. *Multi-agent systems. An Introduction to distributed artificial intelligence*, chapter 4. Addison-Wesley, 1999.

[3] N. Cetin and K. Nagel. A large-scale agent-based traffic microsimulation based on queue model. In *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, CH, 2003. See www.strc.ch.

[4] 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.

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

[6] 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.

[7] M. Ben-Akiva. Route choice models. Presented at the Workshop on "Human Behaviour and Traffic Networks", Bonn, December 2001.

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

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

[10] R. Kitamura. Applications of models of activity behavior for activity based demand forecasting. In *TMIP (Travel model improvement program) Activity-based travel forecasting conference*, June 1996. See http://tmip.fhwa.dot.gov/clearinghouse/docs/abtf/.

[11] T. Arentze, F. Hofman, H. van Mourik, and H. Timmermans. ALBATROSS: A multi-agent rule-based model of activity pattern decisions. Paper 22, Transportation Research Board Annual Meeting, Washington, D.C., 2000.

[12] D. Charypar and K. Nagel. Generating complete all-day activity plans with genetic algorithms. In IATBR'03 [29].

[13] T. Arentze and H.J.P. Timmermans. Representing mental maps and cognitive learning in micro-simulation models of activity-travel choice dynamics. In IATBR'03 [29].

[14] P.A. Salvini and E.J. Miller. ILUTE: An operational prototype of a comprehensive microsimulation model of urban systems. In IATBR'03 [29].

[15] DYNAMIT www page. See its.mit.edu and dynamictrafficassignment.org, accessed 2003.

[16] DYNASMART. See www.dynasmart.com and dynamictrafficassignment.org, accessed 2003.

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

[18] D. Ettema, G. Tamminga, H. Timmermans, and T. Arentze. A micro-simulation model system of departure time and route choice under travel time uncertainty. In IATBR'03 [29].

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

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

[21] K. Nagel and F. Marchal. Computational methods for multi-agent simulations of travel behavior. In IATBR'03 [29].

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

[23] PTV www page. Planung Transport Verkehr. See www.ptv.de, accessed 2004.

[24] Bundesamt für Strassen. Automatische Strassenverkehrszählung 1999. Bern, Switzerland, 2000.

[25] K. Axhausen. Personal communication.

[26] B. Raney, N. Cetin, A. Völlmy, M. Vrtic, K. Axhausen, and K. Nagel. An agent-based microsimulation model of Swiss travel: First results. *Networks and Spatial Economics*, 3(1):23–41, 2003. Similar version Transportation Research Board Annual Meeting 2003 paper number 03-4267.

[27] 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 689–709. Pergamon, Oxford, 2001.

[28] M. Balmer, B. Raney, and K. Nagel. Coupling activity-based demand generation to a truly agent-based traffic simulation – activity time allocation. In *Presented at EIRASS workshop on Progress in activity-based analysis*, Maastricht, NL, 2004. Also presented at STRC'04, see www.strc.ch.

[29] *Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR)*, Lucerne, Switzerland, 2003. See http://www.ivt.baum.ethz.ch/allgemein/iatbr2003.html.