Presented at: 12th Conference of the International Association of Travel Behavior Research (IATBR), Jaipur, India, 2009. Also VSP WP 09-11, see www.vsp.tu-berlin.de/publications.

Combined agent-based simulation of private car traffic and transit

Marcel Rieser, Transport Systems Planning and Transport Telematics, Salzufer 17–19, Sekr. SG 12, Berlin Institute of Technology, D-10587 Berlin, *and* Institute for Transport Planning and Systems, ETH Zurich, CH-8093 Zurich;
Kai Nagel, Transport Systems Planning and Transport Telematics, Salzufer 17–19, Sekr. SG 12, Berlin Institute of Technology, D-10587 Berlin

Abstract

Agent-based Transport Simulations look at individual travelers and simulate their movements through a transportation network. Currently, such simulations are either limited to a small region, or only support a single mode (usually private car traffic) for computational reasons. MATSim (Multi-Agent Transport Simulation) is such a tool that can handle large scenarios (e.g. up to whole Switzerland), but currently only simulates cars. This work describes how MATSim was extended to handle other transportation modes besides private car traffic, with a strong emphasis on transit. This includes not only the simulation of such traffic, but also the agents' mode choice decisions. The agents make use of a utility function to score their experiences with different transportation modes, which influences which mode they choose in following iterations of the simulation. This basically means that we no longer rely on existing mode choice models, e.g. known from the 4-step process. Instead, mode choice is iteratively adapted during the simulation ("traffic assignment" in the 4-step process). The combination of mode choice and traffic assignment leads to better results compared to a pre-calculated mode choice, as the results will show.

1 Introduction

It has been shown in the past *(*1, 2*)* that the traditional four-step process *(*3*)* has some shortcomings, especially when it comes to modeling mode choice problems or time-dependent measures to influence traffic. Agent-based Transport Simulations look at individual travelers and simulate their movements through a transportation network. Such simulations are fully time-dynamic, making it possible to research proposed measures in more detail than traditional tools are usually able to. But currently, such simulations are either limited to a small region, or only support a single mode (usually private car traffic) for computational reasons. VISSIM *(*4*)* allows the very detailed simulation of different vehicles, but is mostly limited to small scenarios covering a few roads and intersections only. MATSim (Multi-Agent Transport Simulation, *(*5*)*) is an agent-based simulation tool that can handle large scenarios (e.g. up to whole Switzerland), but currently only simulates cars.

But the need for tools that support large scenarios as well as multiple transportation modes is rising. Not only are the metropolitan areas growing, but also the dimensions in which measures must be handled. The introduction of road pricing or new public transit offerings usually cannot be limited to a small area, but has to take larger regional effects into consideration as well. Similarly, looking only at a single transportation mode is useless as most measures also target a shift in the modal split.

While there exist a few agent-based public transit simulation tools *(*e.g 6*,* 7*,* 8*)*, they are either limited to transit only or are again only applied to smaller scenarios. In other cases, additional modes like transit were artificially added to simulations only supporting cars, e.g. by adding traffic lights that only influence transit vehicles in order to simulate transit stops *(*9*,* 10*,* 11*)*. While this may give visually a good impression, the model behind the simulation is again very limited to react to advanced measures.

This work describes how MATSim was extended to handle other transportation modes besides private car traffic, with a strong emphasis on public transit. This includes not only the simulation of such traffic, but also the agents' mode choice decisions. Starting with a simple extension to support different transportation modes, the agent-based simulation is extended to model public transit in high detail along regular car traffic, creating a multimodal agent-based simulation.

2 Simulation Structure

2.1 Overview

Each traveler of the real system is modeled as an individual agent in our simulation. The overall approach consists of three important pieces:

- Each agent independently generates a so-called *plan*, which encodes its intentions during a certain time period, typically a day.
- All agents' plans are simultaneously executed in the simulation of the physical system. This is also called the *traffic flow simulation* or *mobility simulation*.
- There is a mechanism that allows agents to *learn*. In our implementation, the system iterates between plans generation and traffic

> flow simulation. The system remembers several plans per agent, and scores the performance of each plan. Agents normally choose the plan with the highest score, sometimes re-evaluate plans with bad scores, and sometimes obtain new plans by modifying copies of existing plans.

The simulation approach is the same as in many of our previous papers *(*e.g. 12*)*:

A **plan** contains the itinerary of activities that the agent wants to perform during the day, plus the intervening trip legs the agent must take to travel between activities. An agent's plan details the order, type, location, duration and other time constraints of each activity, and the mode, route and expected departure and travel times of each leg.

A plan can be modified by various **modules**. Typical examples of such modules are the *Time Adaptation* module and the *Router* module. The Time Adaptation module changes the timing of an agent's plan. A very simple approach is used which just applies a random "mutation" to the duration attributes of the agent's activities *(*13*)*. The router is a time-dependent best path algorithm *(*14*)*, normally using as link costs the link travel times from the previous iteration.

One of the plans is marked as "selected". The **traffic flow simulation** executes all agents' selected plans simultaneously on the network, and provides output describing what happened to each individual agent during the execution of its plan. The traffic flow simulation is implemented as a queue simulation, where each street (link) is represented as a first-in first-out queue with two restrictions *(*15, 16*)*. First, each agent has to remain for a certain time on the link, corresponding to the free speed travel time. Second, a link storage capacity is defined which limits the number of agents on the link. If it is filled up, no more agents can enter this link.

The modules base their decisions on the output of the traffic flow simulation (e.g. knowledge of congestion) using **feedback** from the multi-agent simulation structure *(*17, 18*)*. This sets up an **iteration cycle** which runs the traffic flow simulation with specific plans for the agents, then uses the planning modules to update the plans; these changed plans are again fed into the traffic flow simulation, etc, until consistency between modules is reached. The feedback cycle is controlled by the **agent database**, which also keeps track of multiple plans generated by each agent.

20% of the agents generate new plans by taking an existing plan, making a copy of it, and then modifying the copy with either the Time Adaptation or the Router module. The other agents reuse one of their existing plans. The probability to change the selected plan is calculated by a model which in the steady state converges to a logit model:

$$p_j = \frac{e^{\beta \cdot s_j}}{\sum_i e^{\beta \cdot s_i}} \tag{1}$$

where p_j is the probability for plan j to be selected and s_j its current score. β is a sensitivity parameter, set to 2.

The repetition of the iteration cycle coupled with the agent database enables the agents to learn how to improve their plans over many iterations. Due to memory constraints, the number of plans that one agent may have is limited. In that case, the plan with the worst performance is deleted when adding a new plan to a person that already has the maximum number of permitted plans. The iteration cycle continues until the system has reached a relaxed state. At this point, there is no quantitative measure of when the system is "relaxed"; we just allow the cycle to continue until the outcome is stable.

2.2 Scoring Plans

In order to compare plans, it is necessary to assign a quantitative score to the performance of each plan. In this work, in order to be consistent with economic appraisal, a simple utility-based approach is used. The elements of our approach are as follows:

- The total score of a plan is computed as the sum of individual contributions:

  $$U_{total} = \sum_{i=1}^{n} U_{perf,i} + \sum_{i=1}^{n} U_{late,i} + \sum_{i=1}^{n} U_{tr,i} \ , \tag{2}$$

  where U_{total} is the total utility for a given plan; n is the number of activities, which equals the number of trips (the first and the last activity—both "home"—are counted as one); $U_{perf,i}$ is the (positive) utility earned for performing activity i; $U_{late,i}$ is the (negative) utility earned for arriving late to activity i; and $U_{tr,i}$ is the (negative) utility earned for traveling during trip i. In order to work in plausible real-world units, utilities are measured in Euro.

- A logarithmic form is used for the positive utility earned by performing an activity:

  $$U_{perf,i}(t_{perf,i}) = \beta_{perf} \cdot t_{*,i} \cdot \ln\left(\frac{t_{perf,i}}{t_{0,i}}\right) \tag{3}$$

  where t_{perf} is the actual performed duration of the activity, t_* is the "typical" duration of an activity, and β_{perf} is the marginal utility of an activity at its typical duration. β_{perf} is the same for all activities, since in equilibrium all activities at their typical duration need to have the same marginal utility.

$t_{0,i}$ is a scaling parameter. As long as dropping activities from the plan is not allowed, $t_{0,i}$ has essentially no effect.

- The (dis)utility of being late is uniformly assumed as:

$$U_{late,i} = \beta_{late} \cdot t_{late,i} \ , \tag{4}$$

where β_{late} is the marginal utility (in Euro/h) for being late, and $t_{late,i}$ is the number of hours late to activity i. β_{late} is usually negative.

- The (dis)utility of traveling is uniformly assumed as:

$$U_{tr,i} = \beta_{tr} \cdot t_{tr,i} \ , \tag{5}$$

where β_{tr} is the marginal utility (in Euro/h) for travel, and $t_{tr,i}$ is the number of hours spent traveling during trip i. β_{tr} is usually negative.

In principle, arriving early or leaving early could also be punished. There is, however, no immediate need to punish early arrival, since waiting times are already indirectly punished by foregoing the reward that could be accumulated by doing an activity instead (opportunity cost). In consequence, the effective (dis)utility of waiting is already $-\beta_{perf}\, t_{*,i}/t_{perf,i} \approx -\beta_{perf}$. Similarly, that opportunity cost has to be added to the time spent traveling, arriving at an effective (dis)utility of traveling of $-|\beta_{tr}| - \beta_{perf}\, t_{*,i}/t_{perf,i} \approx -|\beta_{tr}| - \beta_{perf}$.

No opportunity cost needs to be added to late arrivals, because the late arrival time is spent somewhere else. In consequence, the effective (dis)utility of arriving late remains at β_{late}. – These approximate values (β_{perf}, $\beta_{perf} + |\beta_{tr}|$, and $|\beta_{late}|$) are the values that would correspond to the consensus values of the parameters of the Vickrey model *(*19*)*.

3 Simple Mode Choice and Non-Car Transportation Mode

This section reports how mode choice and public transit can be integrated in the agent-based simulation in an approximative way *(*1, 20*)*. The approach uses twice the car free speed travel times as door-to-door public transit travel times. The advantage of this approach is that it works completely without any specific knowledge of the public transit availability in an area, and our results indicate that even in such a situation, car traffic results are closer to reality if such a “pseudo-transit” mode is added. The obvious dis-advantage is that it will not be capable of simulating any specific measures, such as adding/removing/accelerating/decelerating a specific line.

3.1 Mode Choice Model

The basic idea behind our mode choice model is that each agent always has at least one "car" plan and one "non-car" plan. Apart from that, plans are treated as described earlier. Since this always keeps both modes in the choice set, a decision between plans according to Eq. 1 is also a choice between modes.

This requires changes in many parts of the simulation framework, namely the transport simulation, the scoring of plans as well as the replanning. These changes are described in the following.

3.2 Generating non-car plans

To generate non-car plans, an initial demand with car plans must exist already. Starting with that initial demand, the leg modes of all legs in each plan are set to "car", and the fastest routes are calculated. Then, each plan is duplicated, changing all leg modes in the duplicated plans to "non-car".

The duration of every non-car trip is assumed to take twice as long as the car mode at free speed, but no exact route is provided. This is based on the (informally stated) goal of the Berlin public transit company to generally achieve door-to-door travel times that are no longer than twice as long as car travel times. This, in turn, is based on the observation that non-captive travelers can be recruited into public transit when it is faster than this benchmark *(*21*)*. For the purposes of this research, it is assumed that all non-car modes very roughly have the shared characteristics that they are slower than the (non-congested) car mode. In the same vein, both for car and for non-car trips there are no separate considerations of access and egress in this simple model.

3.3 Handling non-car plans

At that time, the simulation only supported a road-network, but no walk- or rail-network. Thus, only car legs can be truly simulated. Agents on non-car legs are teleportated from one location to the next. But the teleportation is not instantaneously, but takes some amount of time, which can be stored in the legs as planned travel duration. While this does not impose any public transit vehicles' capacity constraints, it would still allow us to have individual travel times, depending on agents' demographics or chosen non-car mode (e.g. bike, walk, public transit, . . .). The simulation still generates departure and arrival events for non-car legs, which can be used for analyses.

The scoring of non-car plans is very similar to the scoring of car plans as described in Sec. "Simulation Structure: Scoring Plans", only the marginal disutility of traveling changes. This is expressed by using $\beta_{tr,nc}$ for the marginal utility of traveling, instead of $\beta_{tr,car}$. Note once more that $\beta_{tr,car}$ and $\beta_{tr,nc}$ are *not* values of time by themselves, but they are *additional* marginal disutilities caused by traveling, in addition to the opportunity cost of time. This is consistent with econometric approaches *(*22*)*.

During replanning, plans are duplicated and modified (see "iteration cycle" in Sec. "Simulation Structure, Overview"). This also holds for non-car plans. The only difference is that the plans deletion module makes sure that at least one plan of every mode is kept for every agent. This is to make sure that all agents keep their ability to change mode until the end of the iterations.

The above steps integrate mode choice into the replanning process that takes place iteratively with the simulation. Instead of precalculating the mode choice before the traffic assignment, as it is done in the traditional four-step process, mode choice is now treated at the same level as route choice in the traffic assignment.

3.4 Large-scale application

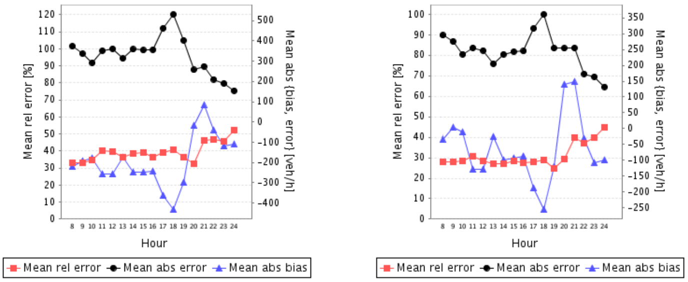
The mode choice model was applied to a large-scale, real-world scenario. We used the area of Zurich, Switzerland, for this application, which has about 1 million inhabitants. The following paragraphs only give a simplified description of the scenario. A full description of the scenario can be found in *(*23*)*.

The network used is a Swiss regional planning network that includes the major European public transit corridors. It consists of 24 180 nodes and 60 492 links.

The simulated demand consists of all travelers within Switzerland that are inside an imaginary boundary around Zurich at least once during their day *(*23, 24*)*. All agents have complete day plans with activities like *home*, *work*, *education*, *shopping*, *leisure*, based on microcensus information *(*25, 26*)*. The time window during which activities could be performed was limited to certain hours of the day: *work* and *leisure* could be performed from 06:00 to 20:00, *shopping* from 08:00 to 20:00, while *home* and *leisure* had no restrictions.

To speed up computations, a random 10% sample was chosen from the synthetic population for simulation, consisting of 181 725 agents. For comparison, the same scenario was run with the pre-calculated mode choice *(*see 23*)*.

Simulated traffic volumes were compared with the hourly traffic volumes from 159 real-world counting stations. Fig. 1 shows, in red, the mean



(a) pre-calculated mode choice

(b) mode choice during simulation ($\beta_{tr,nc} = -3$)

Figure 1: Comparison of simulated traffic volumes with real-world counts. Note different scales on y-axis

relative error between hourly flows in reality and hourly flows from the simulation. The left figure contains the result from the fixed, pre-determined mode choice, the right figure the result of the new adaptive mode choice. One notices a quite distinct reduction in the average error, from about 40% to about 30%. Also the absolute bias, in blue, is reduced.

4 Detailed Public Transit Simulation

Accessibility by public transit often differs from the accessibility by car. Especially rural areas usually only have sparse public transit reachability, whereas there are still many roads available for private cars. In addition, the public transit service quality may change over the time of day much more than the accessibility for private cars. While the actual accessibility by car is mostly dependent on the network congestion—assuming a place is accessible at all—the accessibility by public transit also depends on the actual schedule. For example, many regions may not be served by public transit during the night hours.

To accommodate these facts, the queue-based traffic simulation in MATSim was extended to include public transit traffic as well. This includes the detailed simulation of public transit vehicles and passengers. Agents are able to board and exit vehicles in order to get from one place to another, replacing the teleportation used in the simple model presented before. This section highlights these changes in more detail.

4.1 Required Data

The existing agent-based simulation uses a road-network and so-called *plans* to describe the demand on the network. Agents could travel on all links of the network. In the case of integrated public transit traffic, this may no longer be the case: links could be used to describe bus-lanes or railway tracks, where private cars are not allowed to—or even cannot—drive. Thus, each link has to specify which modes of transports it represents, resulting in a *multimodal network*. In such a network it would also be possible for public transit traffic and private car traffic to coexist on the same link, e.g. buses floating in the regular traffic stream.

The public transit service offerings are described as *public transit schedules*. A public transit schedule contains information about public transit lines, their routes, the travel times between stops, and the time of departure at the start of a route. The route is described as a series of links in a multimodal network. Stop locations contain a coordinate and are assigned to one link in the network, specifying on which link public transit vehicles may approach the stop. In addition, a stop can have additional attributes, e.g. a name or whether a vehicle halting at a stop is blocking other vehicles on the same link or not (useful to model bus bays).

Each departure along a public transit route specifies with which vehicle the route is served. A vehicle, belonging to a specific vehicle type, specifies the number of seats it offers as well as the total passenger capacity. This allows to implement boarding failures during the simulation due to fully occupied vehicles, and also analyze the occupancy of public transit vehicles.

Currently not implemented, but conceptually not challenging, is the inclusion of driver schedules and vehicle schedules. This could be used to research operational aspects of public transit traffic, especially how delays due to congestion interfere with driver and vehicle schedules, maybe making certain combinations more likely to promote delays than others.

Passengers using public transit must specify which public transit line and route they want to board, at which stop they want to enter and at which stop they want to exit the public transit vehicle. This information is stored as the route of a public transit leg, in contrast to the series of nodes or links that build the route of a car leg. If an agent must change lines on a trip from one activity location to another, additional *public transit interaction activities* are inserted between the two (or more) public transit legs. Such public transit interaction activities are also added after the walking leg leading from an activity location to a public transit stop, and before the final walking leg leading from a public transit stop to the actual activity location if the access and egress paths should be included in the simulation. Such route data for passengers can be generated using a public transit router, which is described in a later section.

4.2 Integrated Public Transit Simulation

Given a public transit schedule, the simulation creates a driver-agent for each departure along a route and assigns the driver to the corresponding vehicle. The public transit drivers behave like agents with a plan containing two activities and one leg, the leg being the route of the public transit line. The public transit driver waits at the location of the first activity until its departure time has come. Then, the agent drives its public transit vehicle along the public transit route, until it reaches the end of the route, where it starts its second activity that effectively ends its plan. As noted before, it would be possible to adapt the plans of public transit drivers to include more than one leg, representing a real driver's schedule.

The simulation itself distinguishes private vehicles (i.e. cars) from public transit vehicles. Every time a public transit driver passes a stop along its route, the simulation advises the driver to handle the stop. As the simulation is based on the queue model *(*15, 16*)*, interaction of vehicles with other parts of the simulations is limited to two locations on a link:

- at the very beginning of a link, when an agent is added to a link's queue
- at the very end of a link, when an agent is taken out of a link's queue

Vehicles departing from or arriving at an activity are inserted or removed from the traffic stream at the end of a link in the current model. To be consistent with that model, public transit stops located on a link are handled when the public transit vehicle is at the end of a link.

If agents depart at an activity location, the simulation used to insert these agents at the end of a link so that they could be moved on through the network as vehicles. Now, the simulation first checks whether the agent's next leg is a car trip or a public transit trip. In the first case, the simulation proceeds as usual. In the latter case, the agent is added to a waiting queue at the desired stop location.

To handle a public transit stop, a driver first gives all passengers in its vehicle the chance to exit the vehicle at the specific stop. After that, the driver iterates over the waiting queue at the stop to see if any agent wants to board the driver's vehicle. This is necessary as one stop could be served by more than one line, resulting in some agents waiting for another public transit line. Given there is enough free capacity left in the public transit vehicle, the agents are removed from the waiting queue at the stop and added as passengers to the vehicle. If the capacity is not sufficient, only the front-most passengers can enter the vehicle until the capacity limit is reached.

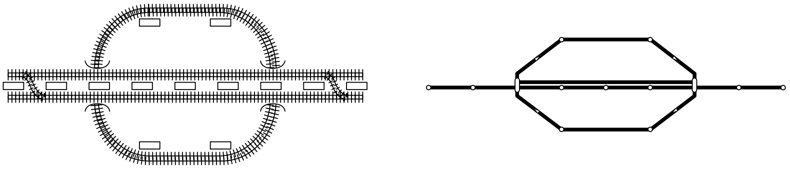
The driver has to wait for a certain amount of time at the stop location, depending on the number of passengers leaving or entering the vehicle. After that time, the driver checks again if additional agents have arrived at the stop location, giving them a chance to enter the vehicle as well. Only after no more passengers enter or leave the vehicle—either because there is no more demand, or because the vehicle is fully loaded—the driver continues along its route.

4.3 Schedule-based Public Transit Router

Based on the public transit schedule, a special routing algorithm searches for the least-cost path from one location to another location using public transit services and short walk legs only. The implemented public transit router is based on Dijkstra's algorithm for finding shortest paths *(*27*)* with some modifications to better accommodate the public transit route search problem.

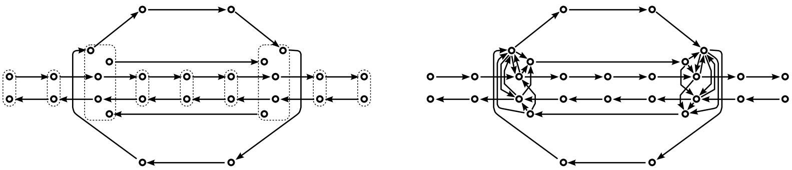
The network the public transit router operates on is generated from the public transit schedule. Each public transit route stop builds a node of the network, and for each public transit route corresponding links are added connecting the stops of the route. After this first step, the network consists of several linear strings of connected nodes, representing the public transit routes from the schedule, but with no connection in-between them yet. In a second step, each node is connected with additional links to other nodes within a configurable distance. These links, that can be seen as transfer or walking links, represent the interchange facilities; not only within one physical facility, but also between nearby stop locations of different lines. Heuristics are applied to reduce the number of transfer links, an important factor in the computational performance of Dijkstra's algorithm. Nodes being the start locations of a public transit route are seen as departure locations only, thus no transfer links being added to them starting at those nodes. Nodes representing the last stop of a public transit route are arrival locations only, and will have no transfer links ending at those nodes. In addition, no transfer links are added between two nodes that belong to the same public transit line and the same stop facility. This comes from the insight that U-turns on public transit lines may never be part of a least-cost path and such transfer links are thus never needed. Fig. 2 shows the single steps in the process of generating the public transit router network.

Once the network is available, it could be used for handling stop-to-stop routing requests by implementing traditional shortest-path calculations. But in reality, people often have more than one stop location from where they can depart or where they can arrive, especially in urban regions. Finding the best route would require to calculate the least-cost route between any combination of start and end locations, and then selecting the best solution from that choice set. This would multiply the computational burden used to find the optimal route. To overcome this problem, Dijkstra's algorithm was modified in order to support multiple start and end nodes. First, the set of nodes to be expanded is not initialized with a single start



(a) Physical infrastructure. Squares represent platforms of stop facilities.

(b) Line map with 3 different lines: A circular line, an express line, and a regular line.



(c) Public Transit router network without transfer links. The dotted areas group the nodes belonging to the same stop facility.

(d) Complete public transit router network

Figure 2: Generation of the Public Transit Router Network

node only, but with all of the possible start nodes together with their respective time and cost to reach that start node by walk from the agent's originating coordinate. Second, the algorithm does not end when the first (and originally the only) end node is to be expanded, but when the last possible end node is to be expanded. Then, to each end node the cost of egressing from that node to the desired location is added, and only after that the least-cost path is selected and stored in the routed leg. This adds only a very small overhead to the computational performance of the original Dijkstra's algorithm.

4.4 Improved Mode Choice Model

In previous projects, plans were typified to differentiate between the different available transportation modes. That approach proves not to be scalable to a lot more of different transportation modes. In addition, it prevents the creation of plans that use different types of modes combined during a day (e.g. park-and-ride). Besides, the typification required quite some changes to the underlying code base of MATSim, indicating that it does not match well with the conceptual structure of the existing simulation framework.

Instead of using pre-determined types to mark plans and ensure that modes always stay available, the mode choice is now implemented as an additional replanning module. This replanning module first selects a random transportation mode from a given list, then changes all legs to use that mode. Afterwards, the routes for all legs need to be re-calculated. By randomly choosing from a list of given modes, agents can explore new, additional transportation modes without having plans of that type in the first place, foregoing the previously necessary pre-processing of the demand.

4.5 Large-Scale Application

For the large-scale application, again the region of Zurich in Switzerland is used. The initial plans and road network is the same as in the aforementioned scenario. A public transit schedule was taken from the official model for public transit in Zurich *(*28*)*, with the data being given in a file for PTV VISUM *(*29*)*. The data was then converted into a format that MATSim could read. Based on that public transit schedule, a network connecting the stops along the public transit routes was generated. This network representing the public transit services was then combined with the existing road network to create a multimodal network. The network obtained this way consists of two sub-graphs; future applications should try to merge the two sub-graphs where possible, e.g. by matching bus routes onto real roads.

The simulation was run with two variants of the public transit schedule: The first one used the unaltered schedule, while in a second variant the S7, one line of the suburban train system, was removed. The S7 connects small towns along the lake of Zurich with an express link to the city center of Zurich. On the same route, two slower lines (S6, S16) with more frequent stops still serve the towns and ensure the accessibility of the region. For the replanning, 10% of the agents adapt departure times, another 10% use the Router module, and an additional 10% of the agents now switch the transportation modes, forcing a new mode to be tested in the next iteration. The remaining 70% still choose one of their existing plans, according to Eq. 1, effectively also doing mode choice.

Comparing the two simulation runs to each other, one can observe the effect of the missing train line nicely (see Tab. 1). The two slower lines are used by a remarkable number of additional passengers in the case of the missing express line. The total number of travelers along this corridor is lower in the modified variant, suggesting that some passengers are likely to change their transportation mode. This assumption can be verified by analyzing the number of people using public transit in the simulation that live in some of the affected towns: In most towns, the number of agents choosing public transit is lower in the modified variant than in the original variant with all lines available, reflecting the lower attractiveness of the public transit offerings.

The computational performance of the detailed public transit simulation was compared to the model presented in the first part, that did not actually

Table 1: Number of legs using one of the specified public transit lines during a day

Public Transit line	with S7	without S7
S6	65194	70510
S16	43182	56324
S7	73681	n/a
sum	182057	126834

simulate public transit passengers but just teleported the passengers from one location to another based on a simple travel time estimation. The tests were run on systems having two Dual-Core AMD Opteron Processors 2222 running at 3 GHz. Memory was connected through a front side bus clocked at 1000 MHz. Tab. 2 lists the findings of the performance comparison. As can be seen, one iteration of the detailed simulation takes more than twice the time in average than the simple, teleporting simulation. On a closer look, one can distinguish between the time needed for replanning (i.e. finding routes) and for the actual traffic flow simulation. There, one can observe that the actual traffic flow simulation is only around 10% slower if public transit is simulated in detail. On the other hand, replanning takes up much more time. This can be related to the following problems:

- The router algorithm used for car legs is heavily optimized (*(*see 14*)*), while the router algorithm used for public transit is a modified variant of the slower Dijkstra's shortest path algorithm.
- In the case of the detailed public transit simulation, new routes must be calculated when an agent changes its leg mode, while the routes are already available when doing mode choice in the simple model. This effectively doubles the amount of agents that need to recalculate their routes in each iteration.
- In the case of the detailed public transit simulation, the router used for car legs operates on a multimodal network and needs to verify in each step if a chosen link can actually be used by the agent. This overhead is not needed in the case of a roads-only network as it is being used in the simple model.

Luckily, the replanning part can easily be sped up using multiple threads, given a machine has enough CPU cores available, so this should not be a problem for future development and larger scenarios.

5 Future Applications of the Public Transit Simulation

The modifications done to simulate public transit open up several possibilities to also simulate other forms of traffic than just private car traffic and

Table 2: Performance comparison between the simple mode choice model and the detailed public transit simulation (times in [min:sec])

	simple model	detailed public transit simulation
avg. time per iteration	03:35	08:44
avg. time for replanning	00:15	04:51
avg. time for traffic flow simulation	03:10	03:36

urban public transit traffic. Most of these possibilities come from the differentiation between vehicles and persons and the extension that vehicles can now transport more people than just the driver. In the following, a few of these additional forms of traffic are presented. It will be shown how the current simulation infrastructure could be used to simulate those kinds of traffic.

5.1 Route-based Para-transit

Paratransit is usually referred to as a flexible passenger transport mode that does not follow fixed schedules or even routes. It is often operated by fleets of small buses or vans, although other vehicles can be used as well. While paratransit is often the only kind of public transportation available in cities of developing countries, it may exist as well in rural areas of developed countries where a fixed-route, fixed-schedule service is not cost-effective. In the latter case, the offered services are often named as *Dial-a-ride* or similar.

One of the biggest differences between paratransit and the currently implemented infrastructure for simulating public transit are the stop locations. Paratransit vehicles usually stop everywhere where it is suitable when passengers want to get in or out the vehicle.

For route-based paratransit, a simple approach would be to define a stop facility on each link that is served by paratransit. More than one stop per link would not be useful as long as the queue model is used in the simulation, as the queue model in its current form does not allow for more than one place of interaction on a link. Passenger agents could then find the best connection using a router that includes the paratransit route. Instead of using a time table for routing, an average expected waiting time—maybe depending on time of day—could be used as time or cost for accessing a new paratransit vehicle in the routing process. In the simulation, a custom paratransit driver could decide by her/himself if she/he should stop at a stop location or not. This decision could be—for example—depending on the number of passenger already in the vehicle. The same way, a paratransit driver could decide to wait a bit longer at a stop in the hope for additional passengers arriving soon.

5.2 Ride-Sharing

Ride Sharing describes two or more people traveling together in a private car. This can be a family driving jointly to some leisure location, or co-workers that live close to each other and share a common trip to/from work. In some regions, agencies for arranged lifts exist, allowing one to easily find other people driving the same way in order to share rides.

Given the code infrastructure that differentiates persons and vehicles, the implementation of ride sharing should be rather simple. Similar to every public transit departure specifying a vehicle to be used, each car-leg or car-route should also specify with which car the leg or route is undertaken. An additional attribute could define which of the persons is the driver. If such an attribute is missing, a simple algorithm could be used (e.g. a random person in the car that has a driving license; this would already clearly define all the cases where parents bring their children to school, as children usually don't have a driving license).

One point that needs to be taken care of is that all persons that are scheduled to drive together indeed travel together. This means mostly that the driver should not depart before all passengers are in the car. Assuming that such combined trips need a special replanning module that tries to put agents into the same car, this could be either solved with activity end times (passengers must be in the car before the driver) or better with additional route attributes. Such attributes, like a list of passengers, could delay the departure of a vehicle until a condition based on the attributes is fulfilled. This would simplify to model actions like parents picking up their child at a kindergarten, or one co-worker picking up another one somewhere along its route.

6 Conclusions

In a first part, a simple extension to include non-car traffic into an existing agent-based simulation was presented. It was shown how the mode choice model works, and what improvements it brings when applied to a large-scale, real-world scenario. In a second part, the simulation was refined to model public transit traffic in more detail. The conceptual changes were highlighted, showing how public transit can be integrated into an existing, agent-based transportation simulation. At least, an outlook was given how the extended simulation could not only be used for simulating public transit, but also other kinds of transportation like paratransit.

Acknowledgments

The simulations were run on our compute cluster, set up and maintained by the Faculty of Mathematics at the Berlin Institute of Technology (TU Berlin), administered by the group of Prof. Schwandt.

References

[1] Marcel Rieser, Dominik Grether, and Kai Nagel. Adding mode choice to a multi-agent transport simulation. Paper 09-2758, Transportation Research Board Annual Meeting, Washington, D.C., 2009.

[2] Peter Vovsha and Mark Bradley. Advanced activity-based models in context of planning decisions. *Transportation Research Record*, 1981:34–41, 2006.

[3] J. de D. Ortúzar and L.G. Willumsen. *Modelling transport*. John Wiley Sons Ltd, Chichester, 3 edition, 2001.

[4] VISSIM www page. www.ptv.de, accessed 2004. Planung Transport und Verkehr (PTV) GmbH.

[5] MATSIM www page. MultiAgent Transport SIMulation. http://matsim.org/, accessed 2009.

[6] Andrew Nash and Daniel Huerlimann. Railroad simulation using OpenTrack. *Computers in Railways IX*, 2004.

[7] Mohamed Wahba and Amer Shalaby. *Schedule-based Modelling of Transportation Networks*, volume 46 of *Operations Research / Computer Science Interfaces Series*, chapter MILATRAS: A new modelling framework for the Transit Assignment Problem, pages 171–194. Springer US, 2009.

[8] Cristián E. Cortés, Laia Pagès, and R. Jayakrishnan. Microsimulation of flexible transit system designs in realistic urban networks. *Transportation Research Record*, 1923:153–163, 2005.

[9] Multisystems, Inc. Bus rapid transit simulation model: Research and development. Report, USDOT/SBIR Phase 1, Oct. 2000.

[10] Inga M. Note. Integration of a center-running guided busway into an arterial street. Presented at VISSIM User Group Meeting, Seattle, 2001.

[11] Amer Shalaby, Baher Abdulhai, and Jinwoo Lee. Assessment of streetcar transit priority options using microsimulation modelling. *Canadian Journal of Civil Engineering*, 30(6):1000–1009, 2003.

[12] B. Raney and K. Nagel. An improved framework for large-scale multi-agent simulations of travel behaviour. In P. Rietveld, B. Jourquin, and K. Westin, editors, *Towards better performing European Transportation Systems*, page 42. Routledge, London, 2006.

[13] M. Balmer, B. Raney, and K. Nagel. Adjustment of activity timing and duration in an agent-based traffic flow simulation. In H.J.P. Timmermans, editor, *Progress in activity-based analysis*, pages 91–114. Elsevier, Oxford, UK, 2005.

[14] N. Lefebvre and M. Balmer. Fast shortest path computation in time-dependent traffic networks. In *Proceedings of Swiss Transport Research Conference (STRC) (*30*)*. See www.strc.ch.

[15] C. Gawron. *Simulation-based traffic assignment*. PhD thesis, University of Cologne, Cologne, Germany, 1998.

[16] N. Cetin, A. Burri, and K. Nagel. A large-scale agent-based traffic microsimulation based on queue model. In *Proceedings of Swiss Transport Research Conference (STRC) (*30*)*. See www.strc.ch. Earlier version, with inferior performance values: Transportation Research Board Annual Meeting 2003 paper number 03-4272.

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

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

[19] R. Arnott, A. de Palma, and R. Lindsey. Economics of a bottleneck. *Journal of Urban Economics*, 27(1):111–130, 1990.

[20] D. Grether, Y. Chen, M. Rieser, and K. Nagel. Effects of a simple mode choice model in a large-scale agent-based transport simulation. In A. Reggiani and P. Nijkamp, editors, *Complexity and Spatial Networks. In Search of Simplicity*, Advances in Spatial Science, chapter 13, pages 167–186. Springer, 2009.

[21] T. Reinhold. Konzept zur integrierten Optimierung des Berliner Nahverkehrs. In *Öffentlicher Personennahverkehr*. Springer Berlin Heidelberg, 2006.

[22] S.R. Jara-Díaz and R. Guerra. Modeling activity duration and travel choice from a common microeconomic framework. In *Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR)*, Lucerne, Switzerland, 2003. See www.ivt.baug.ethz.ch.

[23] Y. Chen, M. Rieser, D. Grether, and K. Nagel. Improving a large-scale agent-based simulation scenario. VSP Working Paper 08-15, VSP, TU Berlin, 2008. See www.vsp.tu-berlin.de/publications.

[24] M. Vrtic, N. Schüssler, A. Erath, K. Meister, and K. Axhausen. Tageszeitliche Fahrtenmatrizen im Personenverkehr an Werktagen im Jahr 2000. Research report, Swiss Federal Department for Environment, Transport, Energy and Communication, Swiss Federal Office for Spatial Development, Swiss Federal Roads Authority and Swiss Federal Office for Transport, IVT, ETH Zürich, Zürich, 2007.

[25] SFSO. Eidgenössische Volkszählung. Swiss Federal Statistical Office, Neuchatel, 2000.

[26] SFSO. Ergebnisse des Mikrozensus 2005 zum Verkehr. Swiss Federal Statistical Office, Neuchatel, 2006.

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

[28] M. Vrtic, P. Fröhlich, N. Schüssler, K.W. Axhausen, P. Kern, F. Perret, S. Pfisterer, C. Schulze, A. Zimmerman, and U. Heidl. Verkehrsmodell für den öffentlichen Verkehr des Kantons Zürich. Report to the Amt für Verkehr, Kanton Zürich, IVT, Zürich, Ernst Basler + Partner AG, Zürich and PTV, Karlsruhe, 2005.

[29] PTV AG. *VISUM 10.0 Benutzerhandbuch*, 2007.

[30] *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, CH. See www.strc.ch.