1 **A multi-model approach to large-scale multiagent transport simulation**

2 Date of submission: 2012-11-15

3 Pieter J. Fourie
Future Cities Laboratory, Singapore-ETH Centre, #06-01 CREATE Tower, Singapore, 138602
phone: +65 9116 8440
fax:
fourie@ivt.baug.ethz.ch

4 Johannes Illenberger
Transport systems planning and transport telematics, Institute for Land and Sea Transport Systems, TU Berlin Sekr. SG12, Salzufer 17-19, 10587 Berlin
phone: +49 30 314-78 793
fax:
illenberger@vsp.tu-berlin.de

5 Kai Nagel
Transport systems planning and transport telematics, Institute for Land and Sea Transport Systems, TU Berlin Sekr. SG12, Salzufer 17-19, 10587 Berlin
phone: +49 30 314-23 308
fax:
nagel@vsp.tu-berlin.de

6 Words: 6002 words + 5 figures + 1 table = 7502 word equivalents

ABSTRACT

In this paper, we introduce a multi-model approach to a large-scale, activity-based, multi-agent travel demand simulation. The Multi-Agent Transport Simulation toolkit, MATSim, is a full activity-based travel demand model, capable of handling very large urban scenarios in the order of millions of commuters. Its greatest current performance limitation is the network loading simulation, currently a queue simulation ('QSim'). In our application, the multi-model system periodically replaces the current QSim for a number of iterations with a simplified pseudo-simulation ('PSim') that runs approximately two orders of magnitude faster. PSim uses information generated in the preceding QSim iteration to produce an estimate of how well an agent day plan might perform, which allows the existing model framework to select and improve plans before executing them in a full queue simulation.

We test the technique in an extensive scenario for Zurich, Switzerland, incorporating mode choice, road-pricing, secondary activity location choice, activity timing adjustment and dynamic routing. We find that the technique dramatically improves convergence rates for such complex, large-scale simulations, and fully exploits modern multi-core computer architectures. Its simple operational logic promises easy integration with all existing and upcoming MATSim functionality, and opens the door to more sophisticated approaches to large-scale, integrated transportation planning.

INTRODUCTION

It is common knowledge in transportation modeling that transportation is mostly a derived demand; produced by the need for individuals to pursue schedules of activities, separated in time and space. Furthermore, it is known that individuals can adjust their activity schedules to various degrees in response to experienced transport system performance, e.g. by changing the timing and location of activities, changing routes and modes of travel, avoiding toll or sharing rides, etc. Individuals do not have complete freedom in this regard; they are constrained by demographics, material means, household, work and other prior commitments, physical needs and many more *(1)*. No closed-form analytical solution exists that captures the full complexity and dynamics of the transportation system and the individuals that participate and interact within it. Therefore, simulation-based techniques are becoming increasingly prominent as an alternative to aggregate, analytical methods *(2, 3, 4, 5)*.

The Multi-Agent Transport Simulation toolkit, MATSim *(6)*, is a fully activity-based transport demand modeling system. It produces a full description of transport demand in the form of a day plan of activities and connecting trips for every individual ('agent') in a large-scale urban scenario, and the resulting congestion patterns and network performance measures. It simulates the interaction between supply and demand by iteratively executing agent day plans in a mobility simulation (currently a queue simulation, called 'QSim'; also called 'network loading'). After each iteration, executed plans are scored; rewarded for time spent at activities and penalized for time spent traveling or arriving late for activities. Plans are replicated and mutated across a number of choice dimensions and poorly performing plans are discarded. Agent behavior therefore adapts to transport system performance across 'generations' (iterations) through trial-and-error, analogous to evolutionary adaptation *(7, 8)*; the overall approach is thus one of co-evolution *(9, 10, 11)*. The process continues until some user-specified measure of convergence is reached.

Mobility simulations are time-consuming, as the interactions of all agents participating in the transportation network are executed for every second in a 24-hour simulated day. Plan mutators are comparatively fast (if mutation is simple and random), even when mutation occurs across many dimensions. However, as the number of choice dimensions in the scenario increases, the number of iterations and thus the number of mobility simulation runs required to explore the solution space increases. On the other hand, the impact of random changes to day plans on the agents' and thus the transport system's performance rapidly diminishes with increasing iterations; therefore a lot of time gets spent on mobility simulation with diminishing returns in terms of the rate of system evolution.

Compounding the problem is that it is relatively more challenging to gain performance from modern multi-core computer architectures in the case of the mobility simulator design, versus that of the plan mutators. The synchronization required between computer threads in the mobility simulation typically produces diminishing returns in performance with increasing computation cores *(12)*. In comparison, plan mutators operate independently on plans, requiring little or no synchronization; consequently, their performance scales linearly with increasing cores.

There is a growing need to integrate existing and emerging model capabilities such as within-household interaction and coordination *(13, 14, 4)*, ride-sharing *(15)*, social network interaction *(16, 17)*, complex mode-chaining , dynamic multi-modal pricing *(18)*, public transportation, secondary activity location selection *(19, 20)*, spatially distributed parking capacity *(21)*, and multi-day, need-based activity modeling *(22)*; to produce a truly integrated activity-based transport model. As the agent choice dimensions and constraints increase, the model solution space explodes in size and complexity. Consequently, the MATSim solution process is expected to require a dramatic increase in the number of iterations in order to effectively explore the high-dimensional solution space. The efficiency of the solution process therefore needs improvement, while retaining the flexibility designed into the current framework.

In this paper, we introduce a flexible meta-model to the MATSim framework in order to increase the rate of system evolution. Multi-modeling techniques are frequently used in simulation-based optimization, where a simplified model of the system is estimated based on a sample of simulated observations. The simplified surrogate- or meta-model ideally takes a deterministic form that is computationally cheap to evaluate (see *(23)* for a comprehensive review of surrogate-based tehniques).

In our application, the multi-model system periodically replaces the current QSim for a number of iterations with a simplified pseudo-simulation ('PSim') that runs approximately two orders of magnitude faster. PSim uses information generated in the preceding QSim iteration to produce an estimate of how well an agent day plan might perform, which allows the existing model framework to select and improve plans before executing them in a full queue simulation.

The aim of this paper is to investigate the multi-model approach for (a) performance, (b) compatability and (c) solution quality in comparison to the standard approach. To this end, it was applied to a large-scale scenario for Zurich, Switzerland, consisting of 67,239 agents traveling in a network of 60,518 links with a dynamic road-pricing scheme, allowing agents to simultaneously adjust mode choice, discretionary activity location choice, activity timing and travel route.

To date, the application of multi-modeling techniques in transport simulation appears to be sparse. Outside of certain modules developed for MATSim, that can be seen as surrogate models informed by the queue simulation *(24)*, only a single application, developed by Osorio and Bierlaire, serves as a true example of substituting the full simulation with a surrogate model *(25, 26)*. Consequently, the work presented here should be an informative addition to the dynamic traffic assignment literature, and hopefully spark interest in applying a similar approach to other applications.

LITERATURE REVIEW

Performance challenge

Currently, the biggest obstacle to further acceleration of iterated transportation simulations is the network loading simulation. Computational performance improvements mostly come through multiple CPUs or multiple cores (computational nodes, or CPNs), and while the remaining tasks of one iteration are straightforward to distribute across multiple CPNs, this is not true for network loading. The reason is that the physical system is tightly integrated: a vehicle reacting to another vehicle with a typical reaction time of one second means that neighboring simulation items should not go out of synchronization by more than a second. In this situation, spatial decomposition *(27)* minimizes interactions most, and may even allow somewhat longer synchronization delays *(28)* when network links are sufficiently long. However, parallel implementations of the network loading are difficult to maintain stable in terms of software engineering, and making them more stable eats into their performance *(12)*. The standard queue simulation (QSim) used in MATSim is no exception.

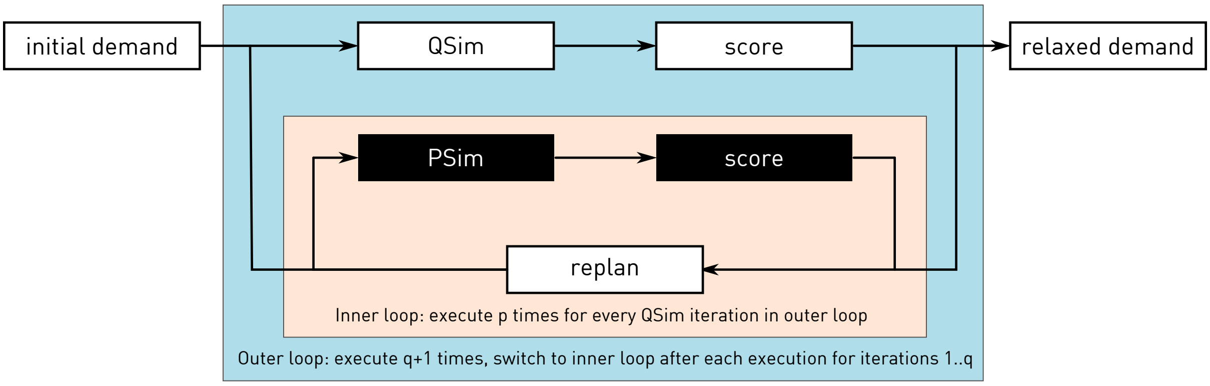


FIGURE 1 Illustration of the operational principle for the multi-model approach in the MATSim framework. The current framework is shown by the white boxes; the logic behind the multi-model approach is to introduce an extra feedback loop (inner loop).

Multi-agent transport simulation

MATSim simulates the traffic produced in a transportation network by agents pursuing daily schedules of activities (plans) separated in time and space. Its principle of operation is shown by the white boxes in Figure 1. The system is fed with an initial demand of agent plans that are repeatedly executed in a QSim network loading. After each QSim run, plan performance is evaluated using a utility-based scoring function. Then, agent plans are mutated along a number of choice dimensions, such as activity start times and durations, route choice, trip transport mode, activity location choice, etc., to produce new plans for execution in the following QSim iteration. With increasing iterations, the number of plans in each agent's memory grows up to a limiting number, following which poorly performing plans are discarded. Consequently, the average score of plans improves with increasing iterations, until a steady state is reached where plan mutations produce only marginal changes in score.

Clearly, this approach is analogous to that of evolution by natural selection, where a genotype (plan) is expressed as a phenotype in the physical environment (agent in traffic) *(7, 10, 29)*. The success of the phenotype determines the longevity of genes in the genotype (combinations of plan elements, such as mode choice, activity timing and location, that become more-or-less stable features across generations).

Mutation approaches

In Figure 1, the 'replan' action represents the mutations producing evolutionary change. Re-planning is done through the chaining of modules into strategies. An example strategy might be:

> Draw 10% of agents, [randomly select a previously executed plan from memory for each agent and make a copy of it], [adjust the start time and duration for each activity in the plan by a random number of seconds less than half an hour], [find the quickest network route between activities based on travel times from the previous iteration], mark these plans as ready for execution.
> For all remaining agents, [select a previously executed plan from memory based on plan score], mark these plans as ready for execution.

In this example, each set of brackets denotes a replanning module. Some modules are merely plan selectors, and do not mutate plans. Other modules can be divided into *random-response* and *best-response* mutators. For the strategy set out above, the start time and duration adjustment module is random-response, while the router is a best-response replanning module, using a Dijkstra algorithm to find the lowest cost route through the network at a given time of day.

Best-response vs. random-response replanning

Best-response modules, though computationally burdensome, reduce total simulation time by exploiting traffic information from the previous iteration, to produce a near-optimal solution to the mutation they are suppose to effect. In the example above, the Dijkstra router produces a truly optimal shortest path for each set of origin and destination points in the agent's plan.

In contrast random-response modules rely on the trial-and-error of the evolutionary algorithm to produce better plans across many iterations, and do not guarantee any improvement in plan fitness.

More complex best-response modules have been developed that explore multiple dimensions of the agent decision space, in order to dramatically reduce the number of iterations until convergence *(*e.g. *24, 30, 31)*.

Such monolithic replanning modules have a number of disadvantages. Firstly, they are purpose-built; if a scenario element is not included in the module, its influence is not considered in the solution. For instance, suppose `modx`, a time-and-mode optimizing module, consistently finds that the best departure time for an agent is 7 am, by car, just when the congestion pricing starts on the highway connecting that agent to work. If `modx` does not consider road-pricing in its design, the resulting plan will be sub-optimal, as the router will, say, find a lower-cost but slower route to work for the given departure time. A more favorable possible alternative, e.g. departing earlier to avoid the road pricing, is unlikely to be found, as `modx` optimizes one sub-problem and the router another.

As the feature set of MATSim grows with time, these modules therefore become obsolete, and require significant re-design to remain relevant. However, due to their design complexity, best-response replanning modules are harder to maintain and integrate with new functionalities than simple random-response modules.

Simulation-based optimization using surrogate models

To date, it appears that only one true multi-model approach has been applied to traffic simulation; where the detailed simulation is used to estimate a simplified surrogate. Osorio and Bierlaire *(26)* combine the output from an AIMSUN dynamic traffic microsimulator with a surrogate model that analytically captures stationary queue distributions. They use this approach to perform simulation-based optimization of signalling plans in a congested network *(25)*.

Their approach differs in two respects from the one presented here. Firstly, their method does not employ an agent-based paradigm. Secondly, they use information from the microsimulation to come up with an analytical description of the network. In our case, we use information from the queue simulation to create a simple lookup table of travel times through the course of the day for every link of the network. The system then uses this information to evaluate and adapt plans before execution in the queue simulation. It therefore relies on the same mechanism of learning through feedback that forms the basis of the MATSim co-evolutionary logic.

Feedback and learning

The idea of predicting the outcome of actions through learning and feedback between the mental and physical domains is not new to transport simulation *(32, 33)*. A multi-level feedback loop, using transport system metrics on one level to inform the location decisions of households and firms, and individual learning on the other as agents respond to resulting changes in demand patterns, has also been the subject of recent investigation *(34)*. Also, UrbanSim *(35)* can use so-called "skims" which means to use a previous output of the assignment model in order to avoid running it – this implies the assumption that travel speeds in the transport system remain the same over a couple of UrbanSim iterations.

DESIGN

Figure 1 illustrates the principle behind the multi-model approach. The system is fed with an initial demand of agent plans, which get executed in QSim. Plans are scored and sent to the replanning modules. An inner loop is then executed for a number of iterations, where new plans are executed in the pseudo-simulation (PSim), scored, and sent for replanning. After, say, p such iterations, plans are selected again for execution in QSim, scored, and the inner loop repeats again for another p iterations. The outer loop repeats q times, then terminates with a final QSim and scoring step, leaving a relaxed demand.

MATSim events

In MATSim, QSim writes out time-stamped, atomic units of information called events, which describe what is happening to each agent at all times. Trawling through these events, it is possible to recontruct every agent's trajectory through the transportation system, and the time they spent at various activity locations.

Consider, for example, an agent traveling from home to work in a small network. Her event stream might look as follows:

```
<event time=21600.0 type="actend" person=1 link=1 actType="home" />
<event time=21600.0 type="departure" person=1 link=1 legMode="car" />
<event time=21609.0 type="wait2link" person=1 link=1 vehicle=1 />
<event time=21610.0 type="left link" person=1 link=1 vehicle=1 />
<event time=21610.0 type="entered link" person=1 link=6 vehicle=1 />
<event time=22057.0 type="left link" person=1 link=6 vehicle=1 />
<event time=22057.0 type="entered link" person=1 link=15 vehicle=1 />
<event time=22487.0 type="left link" person=1 link=15 vehicle=1 />
<event time=22487.0 type="entered link" person=1 link=20 vehicle=1 />
<event time=22846.0 type="arrival" person=1 link=20 legMode="car" />
<event time=22846.0 type="actstart" person=1 link=20 actType="work" />
<event time=61200.0 type="actend" person=1 link=20 actType="work" />
<event time=61200.0 type="departure" person=1 link=20 legMode="car" />
<event time=61200.0 type="wait2link" person=1 link=20 vehicle=1 />
......
```

The XML code shows the simulation time in seconds for each event. This agent (with ID=1), therefore ends activity "home" at six in the morning, departs by car (vehicle ID=1), then enters and leaves a number of links in the network to arrive at work at 06:20:46. The agent departs from work at the scheduled time of 5pm, as specified in her day activity plan, and continues home. Each link traversed is identified explicitly by a link ID. The time taken to traverse a link is generated by the queue simulation dynamics *(*see *36)*, and is therefore a stochastic, emergent property of the simulation.

The default scoring function, derived from Charypar and Nagel *(37)*, in its simplest form, rewards the performing of activities, and penalizes travel and arriving late for activities. During the scoring step in Figure 1, the scoring module evaluates the timing of each agent's actvity start and end events, as well as travel start and end events to derive the total time spent at each activity, time spent traveling, etc. It does not care where the event stream comes from, as long as it is properly formed and chronological for each agent. Consequently, another simulation module than QSim can be used to feed the scoring module with an event stream.

PSim operation

From the QSim event stream, we can deduce the travel time for each agent on each link during the course of the simulated day. We can therefore slice the simulated day up into arbitrary time intervals, say 15 minutes each, calculate the average travel time for each link during every interval, and store these values in a lookup table.

Suppose a replanning module now produces a new plan for the agent above, where she leaves home a little later, or takes a different route to work. The PSim module constructs an event stream that represents her *expected* experience in the transport system, by reading the appropriate times from the lookup table for each link in her route, at each relevant time interval. It passes this event stream to the scoring module, which now produces an expected score for the new plan, and keeps the scored plan in the agent's memory. After repeating the process a number of times, we reach the agent's memory limit, and the poorest performing plan is discarded at the end of each iteration.

The agent is now learning not from the full stochastic queue simulation, but a simplified representation of it; consequently PSim is a surrogate model for QSim. After a number of iterations, we pass the agents back to QSim, to evaluate actual plan performance and produce an updated lookup table of travel times, and the process repeats.

No physical interaction occurs between agents in PSim, so it can fully exploit modern multi-core computer architectures, as no synchronization between threads is required and access to data structures outside a PSim thread is read-only. Load balancing is simple; plans scheduled for execution are simply divided up between threads. Event processing is also completely parallelized, as are re-planning operations.

QSim always requires the full set of agent plans, as travel times emerge from their interaction. As there is no interaction between agents in PSim, it makes sense to only simulate newly generated plans, that do not have a score associated with them yet. This cuts down on the expected computational load even further, as each iteration only generates a small number of new plans, depending on the rate of replanning prescribed by the replanning strategy.

EXPERIMENTAL SETUP

We tested the multi-model approach for compatability, computational performance and solution quality by comparing its results for a large-scale simulation scenario against those produced by a baseline simulation run, that uses the default, QSim-only approach. We are interested to find out if if performance gains from the multi-model approach have any implication on the solution state compared to the standard approach.

Simulation scenario

We used the MATSim development scenario of Swiss car traffic crossing or operating within a 30km radius circle around Bellevue, Zurich, as used in the secondary activity location choice study of Horni *et al. (38)*. The scenario, originally developed by Balmer *et al. (39)*, and updated and further documented in *(40, 41)* is regularly used as a benchmark in MATSim investigations.

We use the same 10% sample from *(38)* study, as well as the same network representation and facility information. The scenario contains 67,239 agents traveling in a network of 60,518 links, and a total of 1,697,196 activity facilities. An arbitrary morning toll was introduced on all links exceeding a capacity of 4,000 vehicles per hour.

The following re-planning modules were used in equal measure, with the total replanning rate (proportion of agents replanned) varied as part of the experimental setup:

1. activity start time and duration adjustment;
2. re-routing using travel times from the previous iteration;
3. subtour mode choice – switches the mode of transport of a randomly selected subtour to car/public transport given that, for this scenario, all agents have access to cars;
4. secondary activity location choice: shopping and leisure activities are switched to a randomly chosen location from a set of qualifying facilities.

Public transport is not explicitly simulated, as this capability would require a full public transport schedule of vehicle departure times, and a full set of public transport lines and routes. Instead, trips using public transport are ‘teleported’ during the simulation from origin to destination with a travel time that is twice that of the free speed shortest path through the network *(42)*.

RESULTS

Characterizing solution state

MATSim employs stochasticity at various points in a simulation run, such as agent selection for different modes of replanning, plan selection for execution, and transition rules at intersections during a queue simulation. In order to make runs repeatable, a seed number is set for the Java random number generator at the beginning of a simulation run.

In our experiments, we used the same random seed for all simulation runs, except a baseline QSim-only run. Then, when comparing the solutions of two QSim-runs with the same parameters except random seed, we have an indication of the minimum deviation we can expect between any two runs of the same scenario.

The baseline against which simulation runs were compared was selected as the simulation state obtained by running the scenario for 101 iterations with QSim only, at an overall replanning rate of 30% per iteration, with a maximum agent memory of 5 plans per agent.

Five measures were used to characterize solution state for comparison against the baseline:

Average executed QSim score

We take the 101^{st} iteration score of 175.4 for the baseline run as a reference value. For all other runs, the first QSim iteration where the score was greater or equal to this value was selected and the rest of the measures were calculated.

Departure profile RMSD

Agent departures are compared at 5 minute intervals for the simulated day. We take the root mean square deviation (RMSD) from the baseline departures as an indication of how similar a simulation state is to the baseline in terms of activity timing.

Mode share

We also compare car mode share (number of car trips / total number of trips) for the large-scale scenario, as mode choice is one of the dimensions included in the replanning strategy.

Daily link volume RMSD

We compare the daily volume of car traffic traversing every link in the network against the volumes produced by the baseline run. We take the root mean square deviation (RMSD) from the baseline link volumes as an indication of how similar a simulation state is to the baseline in terms of car traffic volumes.

Agent total travel time difference

We process the event stream to compare the total travel time experienced by each agent in comparison with those produced by the baseline run. We compare the difference for each agent between the two runs, and count the percentage of agents that experienced a difference below five minutes and one minute, respectively.

We refer to the *reference value* for each measure as the value produced by the *reference case*; i.e. the QSim-only run where only the random number seed differs from the baseline setup.

Varying QSim:PSim ratio

When keeping the replanning rate constant, we found that increasing the number of PSim iterations between QSim iterations increases the rate of convergence, as can be seen from Figure 2. In this figure, we compare the utility vs. number of QSim iterations for two QSim:PSim ratios (red) against the reference case (black).

In general, for a given intermediate utility score, the number of QSim iterations required to achieve that score is approximately inversely proportional to the total number of iterations executed during the simulation, e.g. QSim + PSim iterations.

Performance test

Figure 3 compares the influence of QSim:PSim ratio, number of computational cores and replanning rate on simulation (wall clock) time. Here it is clear that the multi-model strategy is only effective as the number of cores committed to the simulation is increased.

Figure 4 shows the wall clock time it takes, with different set-ups, to reach a certain level of convergence, as described earlier. One notices that the computing (= wall clock) time for replanning scales inversely linear in the number of cores. That is, with an ever growing number of cores, that number will shrink more and more. This is due to the computational (and conceptual) decoupling of the replanning: every agent replans for herself. Second, one notices that replacing most of the regular QSim runs with PSim runs, as discussed in this paper, results in significantly reduced QSim contributions to the overall wall clock time, even if one

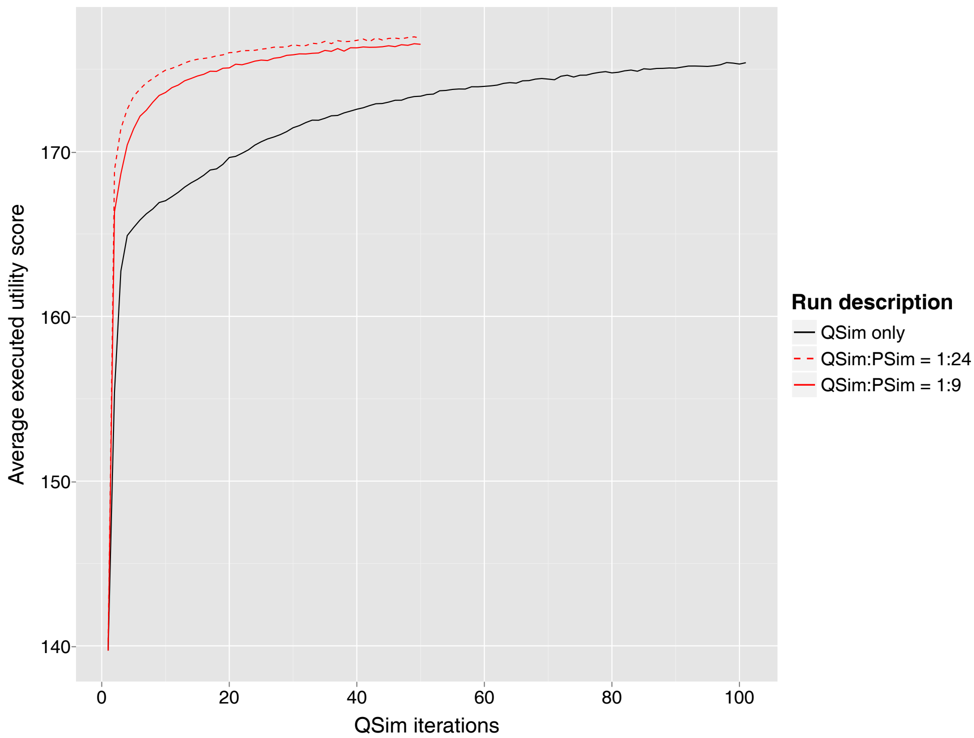


FIGURE 2 Average executed score versus QSim iterations for two ratios of QSim:PSim (red), compared with a reference QSim-only run. Both multi-model runs have a replanning rate of 0.3.

counts in the additional time for the PSim and the additional overhead. At this point, it was possible to reduce the computing time by more than a factor of two, when comparing the 16 core results from the default approach to the fastest version of using the 16 core machine with the multi-model approach.

An interesting result here is that lowering the replanning rate, while increasing the number of PSim iterations in the inner loop gives the best overall performance, with its most significant component being time spent on overhead operations. The reasons for this improved performance in comparison to the other 16 core multi-model run will be explored in the discussion section to follow.

Solution state

Departure profile RMSD

Departure profile RMSD, mode share and daily link volume RMSD for both modes of operation are compared against the reference run in Figure 5. Note from the shape of the RMSD plots that the system has not reached a stable state at the reference score iteration, therefore the system departs from this state in further iterations. This is due to the slow rate of evolution of the random-response replanning modules, and the large number of dimensions being explored in the model. The slope of the RMSD curves only drop off at much higher iterations, especially for departure profile RMSD.

Both the standard QSim-only model and the multi-model approach reach their minimum RMSD value at the iteration where their score equals the reference score of 175.4. However the

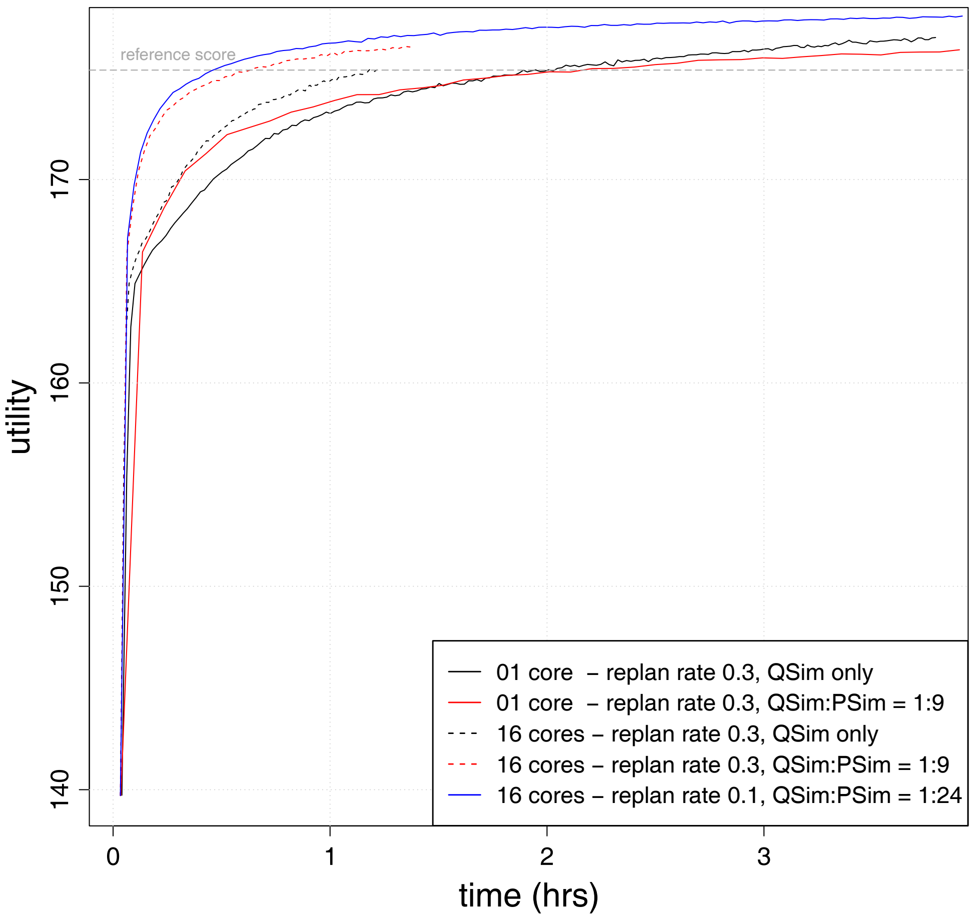


FIGURE 3 Score evolution vs time for large-scale scenario, comparing the influence of QSim:PSim ratio, number of computational cores and replanning rate.

multi-model approach differs from the baseline by a larger margin than the QSim-only reference run at 101 iterations.

Mode share

The multi-model approach produces markedly different car mode shares when compared to the reference run (Figure 5b). The swing towards public transport is much larger for the multi-model runs than for the reference run. The routing and travel time of public transport is independent of network conditions for our simulations, as public transport was not explicitly simulated in order to save simulation time. The meta-model gives many more agents the chance to consider that during the initial iterations, with lots of car congestion, public transit is an attractive alternative. An agent's optimal departure time with public transit is, however, different from the same agent's optimal departure time with car.

This swing to public transport can be minimized by lowering the overall replanning rate, as well as the relative proportion of plans passed to the subtour mode-choice module. A run where this strategy was employed is indicated by the red line in Figure 5(b). For this run, we set the QSim:PSim ratio at 1:24, and the replanning rate at 0.1. The proportion of plans sent for subtour mode-choice mutation was set at half that of other replanning modules.

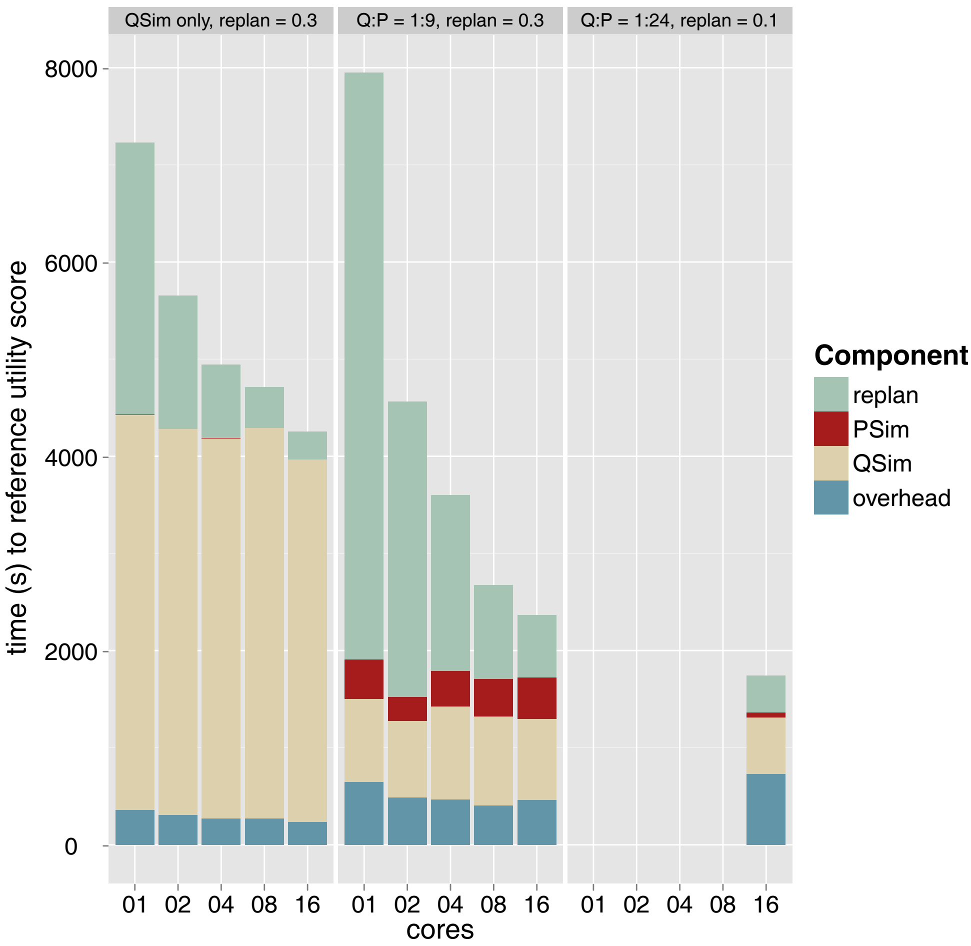


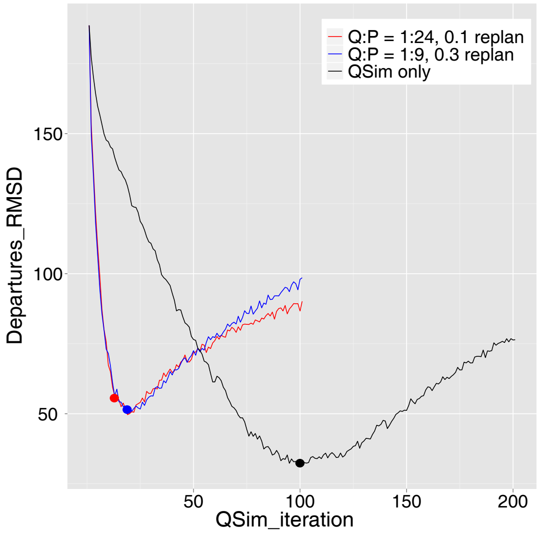
FIGURE 4 Computation time contributions vs number of cores for QSim only (0.3 replanning rate), QSim:PSim = 1:9 (0.3 replanning rate) and QSim:PSim = 1:24 (0.1 replanning rate) at the reference score (grey line in Figure 3).

Daily link volume RMSD

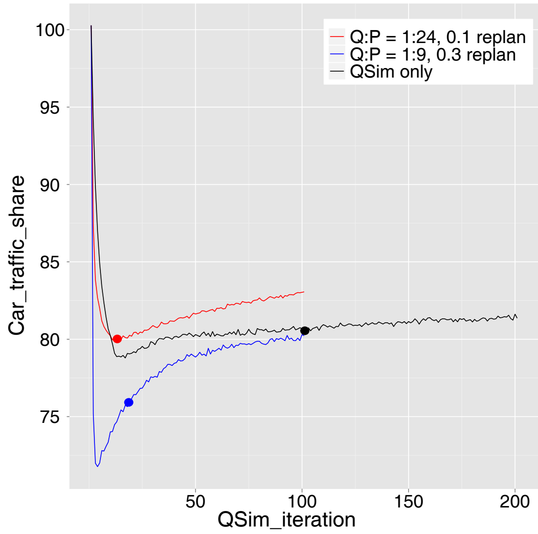
The daily link volume RMSD does not show a minimum at the reference score iteration for any of the runs, and takes longer to reach a minimum. Even though the minimum value is approximately twice that of the reference case, it is still relatively small in absolute value.

Agent total travel time difference

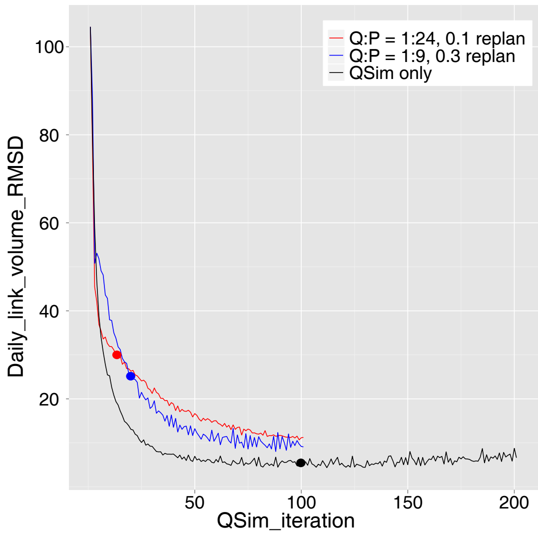
Table 1 compares the agent total travel time difference for the three runs at the reference score iteration, along with the other measures of solution state discussed above. $RMSD_{dep}$ denotes departure profile RMSD; $RMSD_{link}$ is the daily link volume RMSD; $\Delta_{traveltime} \leq 5min.$ and $\Delta_{traveltime} \leq 1min.$ denote the percentage of agents with a total travel time difference (from the baseline) less than 5 minutes and 1 minute, respectively; $share_{car}$ denotes car mode share. We find that the magnitudes for $\Delta_{traveltime}$ between the three cases to be comparable; at least 74% of agents have a total travel time that lies within 5 minutes of that experienced in the baseline run.



(a) Departure profile RMSD



(b) Car mode share



(c) Daily link volume RMSD

FIGURE 5 Departure profile RMSD, car mode share comparison and daily link volume RMSD against the baseline case, for the reference QSim case, and two multi-model runs with varying replanning rate and QSim:PSim ratio. Colored dots indicate the iteration where each run achieved the reference score of175.4.

.

DISCUSSION

The multi-model approach was designed to be consistent with the pre-existing simulation logic of MATSim, and it appears to produce comparable results. In all cases, using the multi-model approach reduces the number of time-consuming QSim iterations required to achieve a given average plan score.

TABLE 1 Summary of solution state measures, compared against the baseline case. Each measure is taken at the point where the average executed score is equal to that of the baseline QSim-only case, at iteration 101.

Run descr.	QSim iter.	$RMSD_{dep}$	$RMSD_{link}$	$\Delta_{traveltime} \leq$ $5min.$ (%)	$\Delta_{traveltime} \leq$ $1min.$ (%)	$share_{car}$ (%)
Reference	101	32.56	5.67	77.1	66.4	80.7
0.3Q:P=1:9	20	50.85	24.53	74.6	65.0	76.2
0.1Q:P=1:24	13	56.42	30.00	76.1	66.7	79.99

Performance

The multi-model approach scales well with an increasing number of cores. Our experiments revealed that the interplay of replanning rate and number of PSim iterations in the inner loop have an important influence on convergence rate. Having a relatively low replanning rate with a higher number of PSim iterations in the inner loop produces the target score in less QSim iterations and less wallclock time.

At first glance, this is a surprising result, because the expected number of plans generated from one QSim iteration to the next is comparable for the two 16-core multi-model runs in Figure 4. The first run has a replanning rate of 0.3 and QSim:PSim ratio of 1:9. Consequently, in 1+9 iterations, the expected number of new plans produced per agent comes to 3, with a standard deviation of 1.44. In comparison, the second run has a replanning rate of 0.1 and QSim:PSim ratio of 1:24, so in 1+24 iterations, it produces only 2.5 new plans per agent on average, with a standard deviation of 1.5.

The reason for the quicker convergence is probably due to the larger number of combinations of replanning modules that can act on any given plan in successive inner loop iterations for the second case. Even if any given combination has only a small chance of occurring; if it is favorable, it will be retained.

The expected value calculation also shows why the total replanning time of the second run is significantly less than the first: In total, it produces 16.7% less plans per outer loop cycle. It suffers, however, from an increased overhead due to a larger total number of iterations.

Solution state

Even though the different measures of solution state depart from those produced by the reference QSim-only run, the departure is not that great for the two measures critical to transport system performance, namely link volume and experienced travel time. The difference in mode share is a cause for concern however. We have come up with a strategy to minimize the overshoot effect, by lowering the replanning rate and relative contribution of subtour mode choice to the replanning strategy. However, further investigation is warranted, in a comparative study with full public transport simulation instead of the teleportation strategy used in this paper.

This study also shows that it is important to consider the relative contribution of each replanning model to the simulation state, because utility on its own is not a complete indication of what is happening in the simulation.

CONCLUSION AND OUTLOOK

The multi-model approach should prove useful in reducing simulation times for most applications of MATSim. Its simple design should make it easy to maintain as MATSim functionality is extended. In this paper, it has been shown to work well with an extensive list of existing MATSim capabilities.

Public transport

In this paper, public transport trips are not explicitly simulated in the QSim iterations, but instead teleported throught the network. Preliminary tests with the multi-model approach have shown promising results for scenarios that explicitly simulate public transport in the presence of private vehicle traffic *(*see *43)*, but further investigation is required.

Social network coordination and ride-sharing

The ultimate purpose of developing the multi-model approach is to explore MATSim's capability to test integrated, complex scenarios. If solution spaces are huge if agents replan independently from each other, they become massively vast when one starts to consider the possibilities that open up when plans are coordinated within households and social networks. A problem of this type stood, in fact, at the beginning of the present investigation: A computational method was needed that would compute utility changes resulting from switching a person's participation from one social group to another *(17)*. If one assumes that this one switch does not influence the network travel times, it is in fact sufficient to recompute the scores of all members of both affected groups. A precurser of the PSim module was used to compute those scores, without running the full network loading.

Parallel simulations

The present paper inserts the multi-model approach so that it stays close to the pre-existing simulation logic. Even though performance gains are the result of the Psim module's capability to fully exploit parallel computation, the simulation logic is still serial.

Currently, the MATSim framework has all agent plans evolving from a single initial condition; the initial demand. The evolutionary logic might preclude certain plans from ever evolving. Consider for instance, an agent whose initial plan is close to a local optimum for being car-only. Assume that the global optimum for this agent is actually a public transport plan, with a markedly different temporal structure to that of the optimal car plan. A random-response switch to public transport for one or more trips produces worse performing plans given the car plan's temporal structure, and are quickly discarded as the agent's memory limit is reached. Consequently, the agent remains stuck at the local optimum.

Once the multi-model capability is fully integrated into MATSim, however, this opens the door to more sophisticated approaches. For example, an agent could optimize a public transit plan over many PSim iterations and only then compare it to an already optimized car plan. Furthermore, such optimizations could run in parallel when computing resources are under-utilized during QSim runs.

ACKNOWLEDGEMENTS

The authors thank the developers of the Zurich scenario for making the data available to us; they include Michael Balmer and Marcel Rieser from Senozon A.G., Zurich, and the team of Kay Axhausen at the Institute of Transport Planning and Systems, ETH Zurich: Francesco Ciari, Andreas Horni, Konrad Meister, Basil Vitins and Rashid Waraich. Gunnar Flötteröd from the Royal Institute of Technology, Stockholm, for insightful feedback, and Dominik Grether from VSP, TU Berlin for his technical assistance during development of the PSim module. This work is funded, in part, by a grant from the National Research Fund of Singapore.

REFERENCES

1. Yoon, S. Y., K. Deutsch, Y. Chen and K. G. Goulias (2012) Feasibility of using time-space prism to represent available opportunities and choice sets for destination choice models in the context of dynamic urban environments, paper presented at the *TRB 2012 Annual Meeting*, GeoTrans Lab., Department of Geography, Santa Barbara, CA 93106, USA.

2. Pendyala, R. (2004) Phased implementation of a multimodal activity-based travel demand modeling system in florida. volume II: FAMOS users guide, *Research Report*, Florida Department of Transportation, Tallahassee. See www.eng.usf.edu/~pendyala/publications.

3. Bhat, C., J. Guo, S. Srinivasan and A. Sivakumar (2004) A comprehensive econometric microsimulator for daily activity-travel patterns, *Transportation Research Record*, **1894**, 57–66.

4. Miller, E. and M. Roorda (2003) A prototype model of household activity/travel scheduling, *Transportation Research Record*, **1831**, 114–121.

5. Arentze, T. and H. Timmermans (eds.) (2005) *ALBATROSS–Version 2.0 – A learning based transportation oriented simulation system*, EIRASS (European Institute of Retailing and Services Studies), TU Eindhoven, NL.

6. Balmer, M., M. Rieser, K. Meister, D. Charypar, N. Lefebvre and K. Nagel (2009) MATSim-T: architecture and simulation times, *Multi-Agent Systems for Traffic and Transportation Engineering*, 57–78.

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

8. Russel, S. and P. Norvig (2010) *Artificial Intelligence – A Modern Approach*, 3. edn., Pearson Education, Inc., 1 Lake Street, Upper Saddle River, New Jersey 07458, USA.

9. Arthur, B. (1994) Inductive reasoning, bounded rationality, and the bar problem, *American Economic Review (Papers and Proceedings)*, **84**, 406–411.

10. Hraber, P., T. Jones and S. Forrest (1994) The ecology of Echo, *Artificial Life*, **3** (3) 165–190.

11. Yang, F. (2005) An evolutionary game theory approach to the day-to-day traffic dynamics, Ph.D. Thesis, University of Wisconsin – Madison.

12. Waraich, R., D. Charypar, M. Balmer and K. Axhausen (2009) Performance improvements for large scale traffic simulation in MATSim, *Technical Report*, ETH Zürich, IVT.

13. Bradley, M. and P. Vovsha (2005) A model for joint choice of daily activity pattern types of household members, *Annual Meeting Preprint*, **05-0954**, Transportation Research Board, Washington D.C.

14. Meister, K., M. Frick and K. Axhausen (2005) A GA-based household scheduler, *Transportation*, **32** (5) 473–494.

15. Ciari, F., N. Schüssler and K. Axhausen (2011) Estimation of car-sharing demand using an activity-based microsimulation approach, *Annual Meeting Preprint*, **11-2077**, Transportation Research Board, Washington D.C.

16. Carrasco Montagna, J. (2006) Social activity-travel behavior: a personal networks approach, Ph.D. Thesis, University of Toronto.

17. Illenberger, J. (2012) Cooperative location choice for leisure activities, *Working Paper*, **11-13**, TU Berlin, Transport Systems Planning and Transport Telematics.

18. Tirachini, A. and D. A. Hensher (2012) Multimodal transport pricing: first best, second best and extensions to non-motorized transport, *Transport Reviews*, **32**, 181–202.

19. Arentze, T. and H. Timmermans (2007) A multi-agent activity-based model of facility location choice and use, *disP*, **170**, 33–44.

20. Horni, A., K. Nagel and K. Axhausen (2011) High-Resolution Destination Choice in Agent-Based Demand Models, *IVT Working paper*, **682**, Institute for Transport Planning and Systems, ETH Zurich, Zurich, Switzerland.

21. Waraich, R. and K. Axhausen (2011) An agent-based parking choice model, *Arbeitsberichte Verkehrs- und Raumplanung*, **696**, IVT, ETH Zurich.

22. Märki, F., D. Charypar and K. Axhausen (2011) Continuous activity planning for continuous traffic simulation, *Transportation Research Record: Journal of the Transportation Research Board*, **2230** (-1) 29–37.

23. Queipo, N. V., R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan and P. Kevin Tucker (2005) Surrogate-based analysis and optimization, *Progress in Aerospace Sciences*, **41** (1) 1–28, January 2005, ISSN 0376-0421.

24. Meister, K., M. Balmer, K. Axhausen and K. Nagel (2006) planomat: A comprehensive scheduler for a large-scale multi-agent transportation simulation, paper presented at the *6th Swiss Transport Research Conference*, Monte Verita, Ascona.

25. Osorio, C. and M. Bierlaire (2008) A multiple model approach for traffic signal optimization in the city of Lausanne, paper presented at the *8th Swiss Transport Research Conference*, Monte Verita, Ascona.

26. Osorio, C. and M. Bierlaire (2009) A multi-model algorithm for the optimization of congested networks, paper presented at the *European Transport Conference, 2009*.

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

28. Charypar, D., K. Axhausen and K. Nagel (2007) An event-driven parallel queue-based microsimulation for large scale traffic scenarios, paper presented at the *Proceedings of the World Conference on Transport Research*, Berkeley, CA.

29. Balmer, M. (2007) Travel demand modeling for multi-agent transport simulations: Algorithms and systems, Ph.D. Thesis, Swiss Federal Institute of Technology (ETH) Zürich, Switzerland.

30. Horni, A., D. M. Scott, M. Balmer and K. W. Axhausen (2009) Location choice modeling for shopping and leisure activities with MATSim, *Transportation Research Record: Journal of the Transportation Research Board*, **2135** (-1) 87–95.

31. Dubernet, T. and K. Axhausen (2012) Including joint trips in a multi-agent transport simulation, paper presented at the *12th Swiss Transport Research Conference*, Monte Verita, Ascona.

32. Arentze, T. and H. Timmermans (2001) Inductive learning approach to evolutionary decision processes in activity-scheduling behavior: theory and numerical experiments, *Transportation Research Record: Journal of the Transportation Research Board*, **1752** (-1) 1–7.

33. Rieser, M., K. Nagel, U. Beuck, M. Balmer and J. Rümenapp (2007) Truly agent-oriented coupling of an activity-based demand generation with a multi-agent traffic simulation, *Transportation Research Record*, **2021**, 10–17.

34. Nicolai, T., L. Wang, K. Nagel and P. Waddell (2011) Coupling an urban simulation model with a travel model – A first sensitivity test, paper presented at the *Computers in Urban Planning and Urban Management (CUPUM)*, Lake Louise, Canada. Also VSP WP 11-07, see www.vsp.tu-berlin.de/publications.

35. Waddell, P., A. Borning, M. Noth, N. Freier, M. Becke and G. Ulfarsson (2003) Microsimulation of urban development and location choices: Design and implementation of UrbanSim, *Networks and Spatial Economics*, **3** (1) 43–67.

36. Dobler, C. and K. W. Axhausen (2011) Design and Implementation of a Parallel Queue-Based Traffic Flow Simulation, *Technical Report*, Zürich.

37. Charypar, D. and K. Nagel (2005) Generating complete all-day activity plans with genetic algorithms, *Transportation*, **32** (4) 369–397, ISSN 0049-4488.

38. Horni, A., K. Nagel and K. W. Axhausen (2012) High-Resolution Destination Choice in Agent-Based Demand Models.

39. Balmer, M., K. Meister, M. Rieser, K. Nagel and K. Axhausen (2008) Agent-based simulation of travel demand: Structure and computational performance of MATSim-T, paper presented at the *Innovations in Travel Modeling (ITM) '08*, Portland, Oregon, June 2008. Also VSP WP 08-07, see www.vsp.tu-berlin.de/publications.

40. Balmer, M., K. Meister, R. A. Waraich, A. Horni, F. Ciari and K. W. Axhausen (2010) Agenten-basierte Simulation für location based services: Schlussbericht KTI 8443.1 ESPP-ES, *Technical Report*, Zürich.

41. Horni, A., B. Vitins and K. Axhausen (2011) The Zurich Scenario: A Technical Overview, *Technical Report*, Zürich.

42. Rieser, M., D. Grether and K. Nagel (2009) Adding mode choice to a multi-agent transport simulation, *Transportation Research Record: Travel Demand Forecasting 2009*, **2132**, 50–58.

43. Rieser, M. and K. Nagel (2009) Combined agent-based simulation of private car traffic and transit, paper presented at the *Proceedings of The 12th Conference of the International Association for Travel Behaviour Research (IATBR)*, Jaipur, India. Also VSP WP 09-11, see www.vsp.tu-berlin.de/publications.