

Using mental simulation to improve the agent learning rate of large-scale multiagent transport simulations

Conference Paper**Author(s):**

Fourie, Pieter Jacobus; Illenberger, Johannes; Nagel, Kai

Publication date:

2013

Permanent link:

<https://doi.org/10.3929/ethz-a-007340021>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

Arbeitsberichte Verkehrs- und Raumplanung 793

1 **Using mental simulation to improve the agent learning rate of large-scale multiagent**
2 **transport simulations**

3 Date of submission: 2012-07-15

4 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

5 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

6 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

7 Words: 5971 words + 6 figures = 7471 word equivalents

ABSTRACT

1 The Multi-Agent Transport Simulation toolkit, MATSim, employs a trial-and-error evolution-
2 inspired approach, by executing and scoring agent day plans in a time-consuming traffic sim-
3 ulation. This paper introduces an information feedback loop to the MATSim framework that
4 evaluates and improves plans before they are executed. We test the technique in an extensive
5 scenario for Zurich, Switzerland, incorporating mode choice, road-pricing, secondary activity
6 location choice, activity timing adjustment and dynamic routing. We find that the technique
7 dramatically improves convergence rates for such complex, large-scale simulations, and fully
8 exploits modern multi-core computer architectures. Its simple operational logic promises easy
9 integration with all existing and upcoming MATSim functionality, and opens the door to more
10 sophisticated approaches to large-scale, integrated transportation planning.

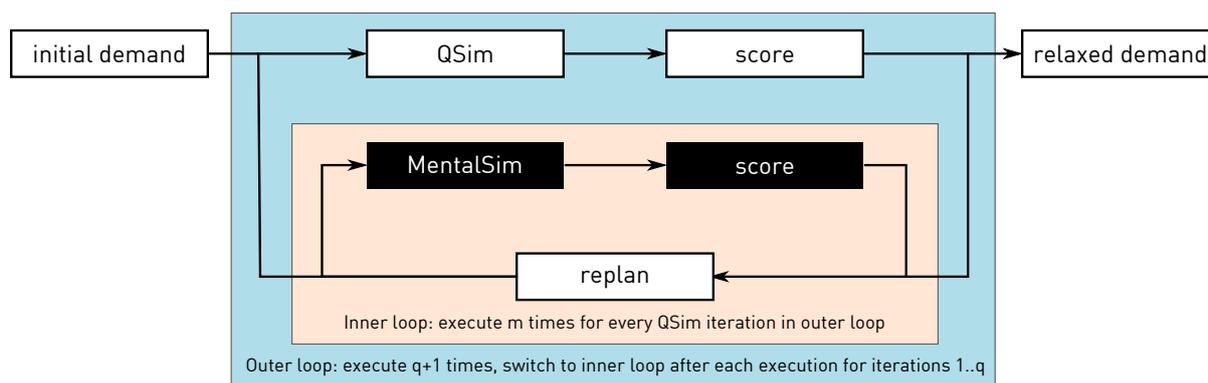


FIGURE 1 Illustration of the operational principle for mental simulation in the MATSim framework. The current framework is shown by the white boxes; the logic behind mental simulation is to introduce an extra feedback loop (inner loop).

INTRODUCTION

1 Humanity's dominance over nature is largely due to our ability to constantly adapt to challenges
 2 in our environment within a single lifetime, rather than across evolutionary time. Based on
 3 knowledge about what's happening in the world, we can evaluate our intended actions in
 4 our minds before executing them in the real world, thus avoiding the wasteful trial-and-error
 5 approach that characterizes evolutionary adaptation. In this paper, an analogy of this concept is
 6 applied to a co-evolutionary, multi-agent transport simulation system to design a flexible strategy
 7 for improving simulation times.

8 The Multi-Agent Transport Simulation toolkit, MATSim (1), simulates agent learning by
 9 iteratively executing agent day plans of activities and connecting trips in a mobility simulation
 10 (currently a queue simulation, or QSim). After each iteration plans are mutated across a number
 11 of choice dimensions and poorly performing plans are discarded. Agent behavior therefore
 12 improves only across 'generations' through trial-and-error, analogous to evolutionary adaptation.

13 Mobility simulations are time-consuming, as the interactions of all agents participating in
 14 the transportation network are executed. Their performance also does not scale well with the
 15 parallel-processing capabilities offered by modern computer architectures. This is due to the
 16 synchronization required between computational threads when transferring agents between
 17 portions of the network handled by different computational cores. In fact, the complexity
 18 introduced by this synchronization makes the single-threaded version of QSim much more
 19 reliable and easy to maintain and debug.

20 However, despite being time-consuming, QSim matches the dynamics of the real transport
 21 system quite well, and produces various indexes of transport system performance, such as
 22 network link travel times and volumes through the course of the day, and waiting times at public
 23 transport stops. The method proposed in this paper uses this information to evaluate and adapt
 24 plans before execution, resulting in improved agent learning rates and system convergence time.

25 Multi-agent transport simulation

26 MATSim simulates the traffic produced in a transportation network by agents pursuing daily
 27 schedules of activities (plans) separated in time and space. Its principle of operation is shown by

1 the white boxes in Figure 1. The system is fed with an initial demand of agent plans that are
2 repeatedly executed in a traffic simulation (QSim). After each QSim run, plan performance is
3 evaluated using a utility-based scoring function. Then, agent plans are mutated along a number
4 of choice dimensions, such as activity start times and durations, route choice, trip transport
5 mode, activity location choice, etc., to produce new plans for execution in the following traffic
6 simulation. With increasing iterations, the number of plans in each agent's memory grows up to
7 a limiting number, following which poorly performing plans are discarded. Consequently, the
8 average score of plans improves with increasing iterations, until a steady state is reached where
9 plan mutations produce only marginal changes in score.

10 Clearly, this approach is analogous to that of evolution by natural selection, where a genotype
11 (plan) is expressed as a phenotype in the physical environment (agent in traffic) (2, 3, 4). The
12 success of the phenotype determines the longevity of genes in the genotype (combinations of
13 plan elements, such as mode choice, activity timing and location, that become more-or-less
14 stable features across generations).

15 **Mutation approaches**

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

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

24 For all remaining agents, [select a previously executed plan from memory based on
25 plan score], mark these plans as ready for execution.

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

31 **Best-response vs. random-response replanning**

32 Best-response modules, though computationally burdensome, reduce total simulation time
33 by exploiting traffic information from the previous iteration, while random-response modules
34 rely on the trial-and-error of the evolutionary algorithm to produce better plans across many
35 iterations.

36 More complex best-response modules have been developed that explore multiple dimensions
37 of the agent decision space, in order to dramatically reduce the number of iterations until
38 convergence (e.g. 5, 6, 7). In general, these modules apply a metaheuristic approach to explore
39 the solution space, ideally evaluating solution quality using the same utility-based scoring
40 function selected for the scoring step of the MATSim framework.

41 Such monolithic replanning modules have a number of disadvantages. Firstly, they are
42 purpose-built; if a scenario element is not included in the module, its influence is not considered

1 in the solution. For instance, suppose `modx`, a time-and-mode optimizing module, consistently
2 finds that the best departure time for an agent is 7 am, by car, just when the congestion pricing
3 starts on the highway connecting that agent to work. If `modx` does not consider road-pricing
4 in its design, the resulting plan will be sub-optimal, as the router will, say, find a lower-cost
5 but slower route to work for the given departure time. A more favorable possible alternative,
6 e.g. departing earlier to avoid the road pricing, is unlikely to be found, as `modx` optimizes one
7 sub-problem and the router another.

8 As the feature set of MATSim grows with time, these modules therefore become obsolete,
9 and require significant re-design to remain relevant. Which brings us to the second problem;
10 due to their design complexity, best-response replanning modules are harder to maintain and
11 integrate with new functionalities than simple random-response modules.

12 MATSim is an open-source project, developed and maintained by an ever-changing, interna-
13 tional group of volunteers, mainly researchers, who generally rely on public funding for projects.
14 That funding tends to be channeled toward the development of new functionality. Maintenance
15 and integration is therefore focused on core functionality relevant to a project; if the design of a
16 module is not readily understandable, it is likely to remain obsolete or useable in only limited
17 cases for extended periods of time.

18 A final, finer point of contention on the use of best-response replanning modules, is that
19 the plans they produce do not reflect the lack of perfect knowledge and compromise that
20 their real-world counterparts exhibit. They race toward the optimum in each iteration, quickly
21 converging to a highly coordinated system, impervious to the churn from minor changes in
22 behavior occurring day-to-day in the real transport system.

23 **Research idea**

24 To summarize; from a systems engineering point of view, simpler, random response modules
25 are more favorable as they are easier to maintain and integrate. From a transport planning point
26 of view, simpler modules can be chained together to solve more complex problems, albeit at
27 the cost of longer simulation runs due to the expanding search space. Best-response modules
28 use information feedback from the last simulation run to produce plans that are more likely to
29 perform well in the mobility simulation. Plans produced by random-response modules are as
30 likely to perform poorly as otherwise, and it's left to the evolutionary selection process to decide
31 their fate.

32 The purpose of this investigation is simply to introduce an information feedback mechanism
33 into the MATSim framework, where the quality of plans produced by the random response
34 modules are evaluated before execution. Plan quality is evaluated by feeding expected travel
35 times (calculated from system metrics produced in the previous simulation) and activity per-
36 formance times into the same scoring function as is used for the mobility simulation. This
37 process is similar to the 'mental simulation' humans perform when imagining a daily schedule
38 being acted out, given their knowledge of the transport system and the demands of their activity
39 commitments. Therefore we shall refer to this mechanism as mental simulation, and to the set of
40 operational modules as MentalSim.

41 The expectation is not only that mental simulation should produce improved convergence
42 rates for any combination of random-response modules, but that the relative simplicity of the
43 operational principle should make it easy to maintain and integrate with new functionalities
44 introduced to the MATSim system.

1 **Related work**

2 The idea of predicting the outcome of actions through learning and feedback between the mental
3 and physical domains is not new to transport simulation (8, 9). A multi-level feedback loop,
4 using transport system metrics on one level to inform the location decisions of households and
5 firms, and individual learning on the other as agents respond to resulting changes in demand
6 patterns, has also been the subject of recent investigation (10). Also, UrbanSim (11) can use
7 so-called “skims” which means to use a previous output of the assignment model in order to
8 avoid running it – this implies the assumption that travel speeds in the transport system remain
9 the same over a couple of UrbanSim iterations. Indeed, using transport system metrics to inform
10 agent learning is the basis of operation of the best-response replanning modules discussed above.

11 What makes the contribution unique in this case, however, is that the simplicity of the
12 mental simulation approach should produce *extensible* accelerated agent learning. It should
13 allow arbitrary combinations of current and future scenario elements and replanning modules to
14 investigate complex transportation questions, such as ‘what would the influence of a morning
15 toll be on public transport ridership and household ride-sharing in a large, demographically
16 diverse metropolitan area?’; and produce answers in short enough time spans as to prove useful
17 to decision-makers. In addition, the present paper performs systematic investigations concerning
18 the convergence gains with the new approach with respect to wall clock time.

19 The following section provides details on the design of the mechanism, given the framework
20 of MATSim. This is followed by some initial results where the system was tested for speed and
21 solution quality on a scenario for Zurich, Switzerland. Concluding remarks summarize what has
22 been learned, and highlight further possible improvements.

METHOD

23 Figure 1 illustrates the principle behind the mental simulation idea. The system is fed with
24 an initial demand of agent plans, which get executed in a queue simulation (QSim). Plans are
25 scored and sent to the replanning modules. An inner loop is then executed for a number of
26 iterations, where new plans are executed in MentalSim, scored, and sent for replanning. After,
27 say, m such iterations, plans are selected again for execution in QSim, scored, and the inner loop
28 repeats again for another m iterations. The outer loop repeats q times, then terminates with a
29 final QSim and scoring step, leaving a relaxed demand.

30 **MATSim events**

31 In MATSim, the queue simulation produces time-stamped, atomic units of information called
32 events, which describe what is happening to each agent at all times. Trawling through these
33 events, it is possible to reconstruct every agent’s trajectory through the transportation system.

34 As the simulation proceeds, these events are sent to a number of event listeners. These are
35 software modules that run in parallel to the simulation. Their purpose is two-fold: to aggregate
36 and interpret events to calculate agent plan scores on the fly; and to build a dynamic record
37 of transport system performance metrics, e.g. link travel times and volumes, public transport
38 ridership, toll paid, emissions produced, etc.

39 In the default mode of operation, these metrics are used to route new plans through the
40 network, then they are either written out to disk and reset before starting a new iteration, or kept
41 for a number of iterations to perform averaging. This persistence of information capability is
42 exploited by the MentalSim module.

1 **MentalSim design**

2 The MentalSim module is a bare-bone mobility simulator that reads through an agent plan and
3 fires appropriate time-stamped events. What kind of events it generates is determined by the
4 scoring function used for the simulation. The default scoring function, derived from Charypar
5 and Nagel (12), in its simplest form, rewards the performing of activities, and penalizes travel
6 and arriving late for activities. It listens for activity start and end events, as well as travel start
7 and end events.

8 To illustrate its operation, suppose a simple home-work-home agent plan. MentalSim reads
9 the home activity departure time from the plan and fires an activity end event with the departure
10 time as its timestamp. It reads the mode of travel and fires a travel start event for that mode
11 of travel, timestamped a second later. It then asks for the travel time between the origin and
12 destination based on the route given in the plan from the appropriate data provider; suitably
13 called a travel time calculator. This travel time is calculated from the recorded travel times
14 during that time of day from the last QSim iteration. MentalSim adds this time to the last
15 timestamp, and fires a travel end event and work activity start event. The process repeats for all
16 activities and trips in the plan.

17 If a more complex scoring function is used, say one that considers road pricing, MentalSim
18 will also fire link enter and leave events with the appropriate timestamps for all links in the
19 agent's route. The toll costs will then be included if the agent traveled across tolled links or
20 entered a road pricing cordon.

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

26 As there is no interaction between agents in mentalsim, it makes sense to only simulate
27 newly generated plans, that do not have a score associated with them yet. This cuts down on the
28 expected computational load even further, as each iteration only generates a small number of
29 new plans, depending on the rate of replanning prescribed by the replanning strategy.

30 **MentalSim simulation modes**

31 The MentalSim module was designed to operate in two modes; *global* mode and *subset* mode.

32 *Global mode*

33 MentalSim simply replaces the regular QSim for m regular iterations. It thus operates on all
34 newly generated plans passed to it by the replanning modules, with replanning applied across
35 the entire agent population through random selection of agents at the end of each iteration.

36 *Subset mode*

37 Select agents at random, say 10% of the population, and only operate on this subset in the inner
38 loop. Once the subset of agents has been selected, a single QSim-executed plan is randomly
39 selected and copied for each one. The agents' QSim memory is then preserved, to be restored to
40 them when they exit the inner loop. Each agent is then passed to the MentalSim inner loop with
41 only the single copied plan in memory.

1 The agents then build up a series of MentalSim-executed plans until they reach their memory
2 limit, following which they discard poorly performing plans until the maximum number of
3 MentalSim iterations have been reached. The memory limit used in the inner loop can be set
4 to a different size than that used in the outer loop, in order to expand the search space. Then,
5 a single plan is selected for each agent and passed back from the MentalSim inner loop to the
6 QSim outer loop. All other MentalSim plans except the selected plan are discarded, and each
7 agent's stock of QSim executed plans is restored to memory. The selected MentalSim plan is
8 then executed with all other agent plans in a QSim iteration.

9 The reasoning behind the two modes of operation is that agents might become 'delusional'
10 from having their QSim-executed plans mixed up with MentalSim plans. QSim-executed plan
11 scores reflect the 'real' performance of a plan, while MentalSim plan scores only reflect the
12 expected performance of that plan. With the general mode of operation, high enough rates of
13 replanning and a large number of MentalSim iterations in the inner loop will result in agents
14 very quickly having their memory filled with plans that have only been executed in MentalSim.

15 For example, suppose a simulation where 24 MentalSim iterations are run for every QSim
16 iteration, where the rate of replanning is 30% and where the maximum number of plans an agent
17 can hold in memory is 5. From the binomial distribution, we know that the expected number of
18 agents with less than 5 plans generated during the 24 MentalSim iterations comes to only 11.1%.
19 This might hold serious consequences for the quality of our solution, as 'true' information on
20 plan performance is lost.

21 The subset strategy would clearly prevent this situation, by discarding MentalSim plans and
22 preserving the original QSim-executed plans of each agent selected for mental simulation. The
23 only plans that persist in memory will be in the outer loop, and will always have been executed
24 and scored in a QSim iteration.

EXPERIMENTAL SETUP

25 MentalSim performance was tested in a set of simulation experiments, with the following aim:

- 26 1. Test MentalSim behavior in terms of convergence rate when varying parameters such as
27 replanning rates, QSim:MentalSim iteration ratio, and modes of operation;
- 28 2. Test MentalSim's computational performance and solution quality compared to a baseline,
29 QSim-only simulation run of a large-scale scenario with a complex replanning strategy
30 consisting of many modules.

31 Simulation scenario

32 We used the MATSim development scenario of Swiss car traffic crossing or operating within
33 a 30km radius circle around Bellevue, Zurich, as used in many studies (see 6). We use the
34 same 10% sample from that study, as well as the facility information by the same authors. This
35 scenario is supplemented with an arbitrary morning toll on all links exceeding a capacity of
36 4,000 vehicles per hour. The following re-planning modules were used in equal measure, with
37 the total replanning rate (proportion of agents replanned) varied as part of the experimental
38 setup:

- 39 1. activity start time and duration adjustment;
- 40 2. re-routing using travel times from the previous iteration;
- 41 3. subtour mode choice – switches the mode of transport of a randomly selected subtour to
42 car/public transport given that, for this scenario, all agents have access to cars;

1 4. secondary activity location choice: shopping and leisure activities are switched to a
2 randomly chosen location from a set of qualifying facilities;

3 Public transport is not explicitly simulated. Instead, trips using public transport are ‘tele-
4 ported’ from origin to destination with a travel time that is twice that of the free speed shortest
5 path (13).

RESULTS

6 **Characterizing solution state**

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

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

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

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

19 *Average executed QSim score*

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

23 *Departure profile RMSD*

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

27 *Mode share*

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

30 The minimum value (e.g. *reference value*) for each measure is that of the case where only
31 the random number seed differs from the baseline setup. We refer to this case as the *reference*
32 *case*.

33 **Varying QSim:MentalSim ratio**

34 When keeping the replanning rate constant, we found that increasing the number of MentalSim it-
35 erations between QSim iterations increases the rate of convergence, as can be seen from Figure 2.
36 In this figure, we compare the utility vs. number of QSim iterations for two QSim:MentalSim

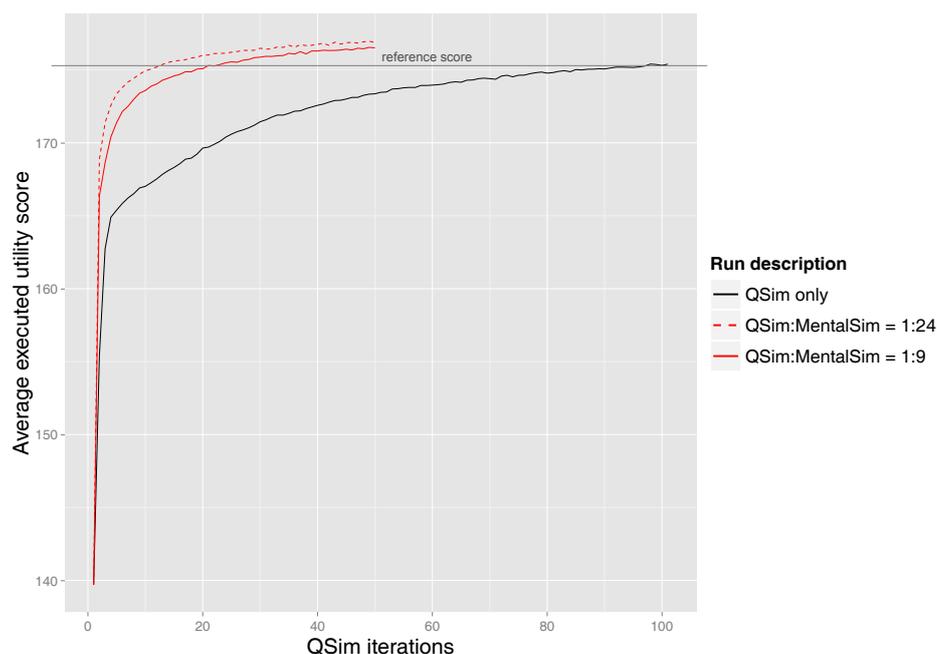


FIGURE 2 Average executed score versus QSim iterations for two ratios of QSim:MentalSim (red), compared with a reference QSim-only run.

1 ratios (red) against the reference case(black).

2 In general, for a given intermediate utility score, the number of QSim iterations required
 3 to achieve that score is inversely proportional to the total number of iterations executed during
 4 the simulation, e.g. QSim + MentalSim iterations. So, for the given comparison, we reach
 5 the utility of 175.4 in 101 QSim + 0 MentalSim = 101 iterations for the QSim only run,
 6 24 QSim + (23 × 9) MentalSim = 231 iterations for the QSim:MentalSim = 1 : 9 run, and
 7 14 QSim + (13 × 24) MentalSim = 326 iterations for the QSim:MentalSim = 1 : 24 run.

8 **Global vs. subset mode**

9 *Score evolution*

10 It was found that the executed score evolution over iterations initially proceeds slightly slower
 11 for subset mode than global mode, if the replanning rate and QSim:MentalSim ratio is held
 12 constant. Also, the spread of plan scores held in memory differs for the first few dozen iterations,
 13 as can be seen in Figure 3. The ribbons in that figure indicate the difference between the average
 14 best score and average worst score in agent memory. Subset mode maintains a larger diversity
 15 in plans for the first few dozen iterations, but then converges to the same spread as the global
 16 mode of operation.

17 The difference between the spread and rate of convergence for the two modes of operation
 18 was found to increase as the number of MentalSim iterations in the inner loop increases.
 19 This probably happens because QSim-executed scores are retained for the subset mode opera-
 20 tion, while the global mode of operation does not treat QSim-executed plans differently from
 21 MentalSim-executed plans. In global mode, poorly performing plans are discarded the moment
 22 the agent's memory limit is reached, be that during a MentalSim or QSim iteration. In subset
 23 mode, the agents' sets of plans grow more gradually with increasing iterations, as the number of

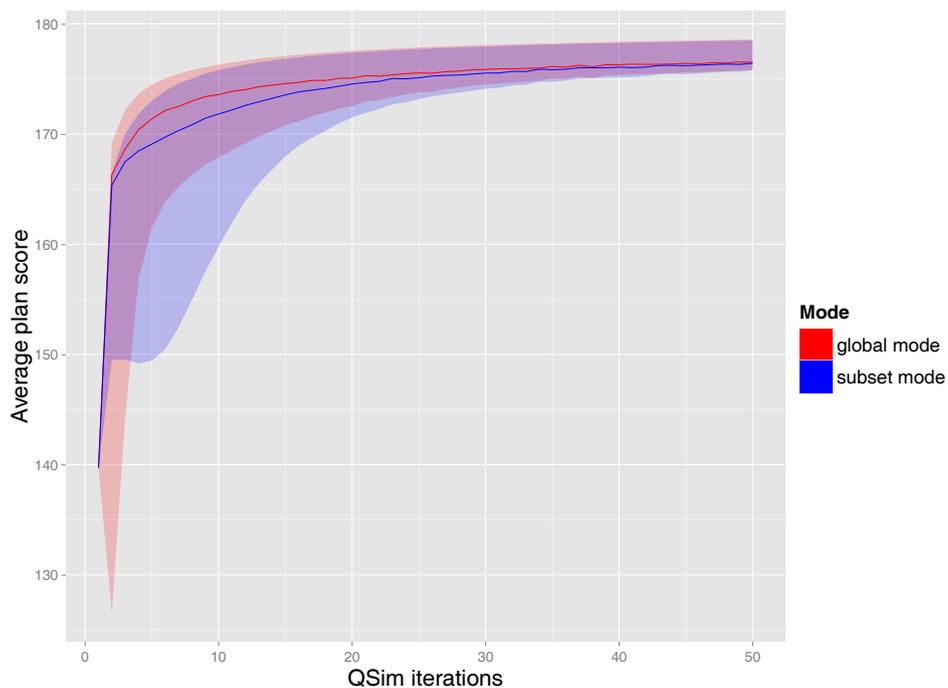


FIGURE 3 Comparison of average score evolution in the large-scale scenario for global mode (red) and subset mode (blue), QSim:MentalSim = 1:9 (0.3 replanning rate). Translucent ribbons indicate the spread of plan scores in agent’s memory.

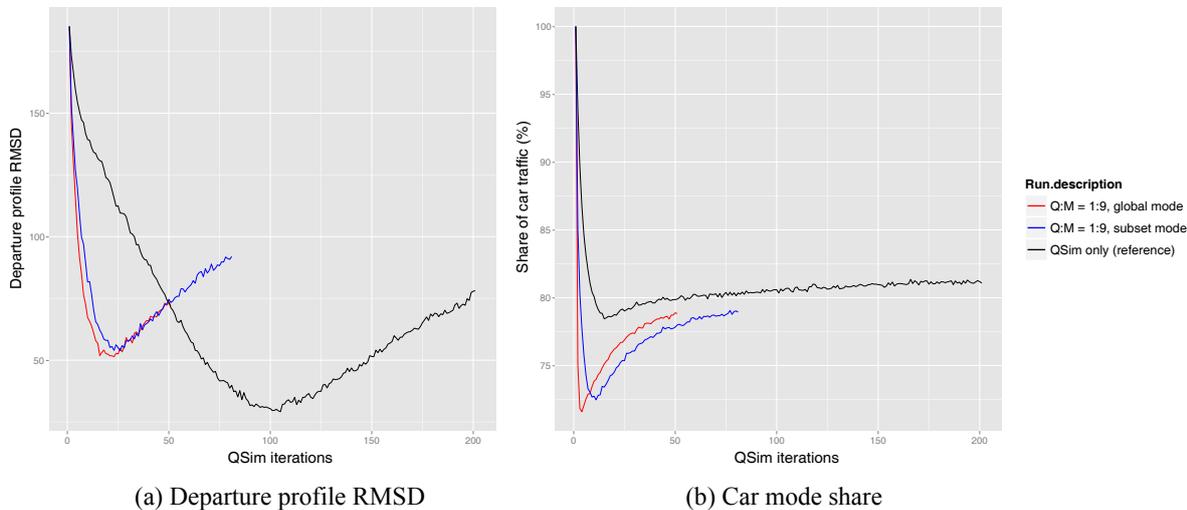


FIGURE 4 Departure profile RMSD and car mode share comparison for the two runs in Figure 3 against the reference run.

- 1 plans held in memory only increase with QSim iterations.
- 2 *Solution state*
- 3 Departure profile RMSD and mode share for both modes of operation are compared against the
- 4 reference run in Figure 4. Here we can see that both modes of operation reach their minimum

1 RMSD value at the iteration where their score equals the reference score of 175.4. However both
2 values are significantly larger than the minimum attained by the reference run at 101 iterations.

3 The reason for this difference is probably due to the different mode shares produced by the
4 MentalSim runs when compared to the reference run (Figure 4b). The swing towards public
5 transport is much larger for the MentalSim runs than for the reference run. The routing and travel
6 time of public transport is independent of network conditions for our simulations, as public
7 transport was not explicitly simulated in order to save simulation time. The mental simulation
8 gives many more agents the chance to consider that during the initial iterations, with lots of car
9 congestion, public transit is an attractive alternative. An agent's optimal departure time with
10 public transit is, however, different from the same agent's optimal departure time with car. In
11 the long run mode share seems to converge towards the reference value.

12 In a simpler experiment investigating this overshoot effect, it was found that executing a
13 few QSim-only runs before switching on the inner loop reduces the effect. During the first 5-10
14 iterations (depending on the rate of re-routing) the largest congestion effects are rapidly reduced
15 by the router, with a proportionately rapid improvement in utility. Then the influence of the
16 router on utility diminishes, and the long process of improvement through random response and
17 selection takes over (the flatter part of the utility curve).

18 **Performance test**

19 It was found that, even though the average QSim executed score improves faster with increasing
20 iterations for all MentalSim-enabled runs, plotting these values versus wall clock time show
21 a radically different indication of performance, as can be seen in Figure 5. It compares the
22 influence of QSim:MentalSim ratio, number of computational cores and replanning rate on
23 simulation (wall clock) time. Here it is clear that the MentalSim strategy is only effective as the
24 number of cores committed to the simulation is increased.

25 Figure 6 shows the wall clock time it takes, with different set-ups, to reach a certain level
26 of convergence, as described earlier. One notices that the computing (= wall clock) time
27 for replanning scales inversely linear in the number of cores. That is, with an ever growing
28 number of cores, that number will shrink more and more. This is due to the computational (and
29 conceptual) decoupling of the replanning: every agent replans for herself. Second, one notices
30 that replacing most of the regular QSim runs with MentalSim runs, as discussed in this paper,
31 results in significantly reduced QSim contributions to the overall wall clock time, even if one
32 counts in the additional time for the MentalSim and the additional overhead. At this point, it
33 was possible to reduce the computing time by more than a factor of two, when comparing the
34 16 core results from the regular approach to the fastest version of using the 16 core machine
35 with MentalSim.

36 An interesting result here is that lowering the replanning rate, while increasing the number
37 of MentalSim iterations in the inner loop gives the best overall performance, with its most
38 significant component being time spent on overhead operations. The reasons for this improved
39 performance in comparison to the other 16 core MentalSim run will be explored in the next
40 section.

DISCUSSION

41 The mental simulation approach was designed to be consistent with the pre-existing simulation
42 logic of MATSim, and it appears to produce comparable results. In all cases, using the mental

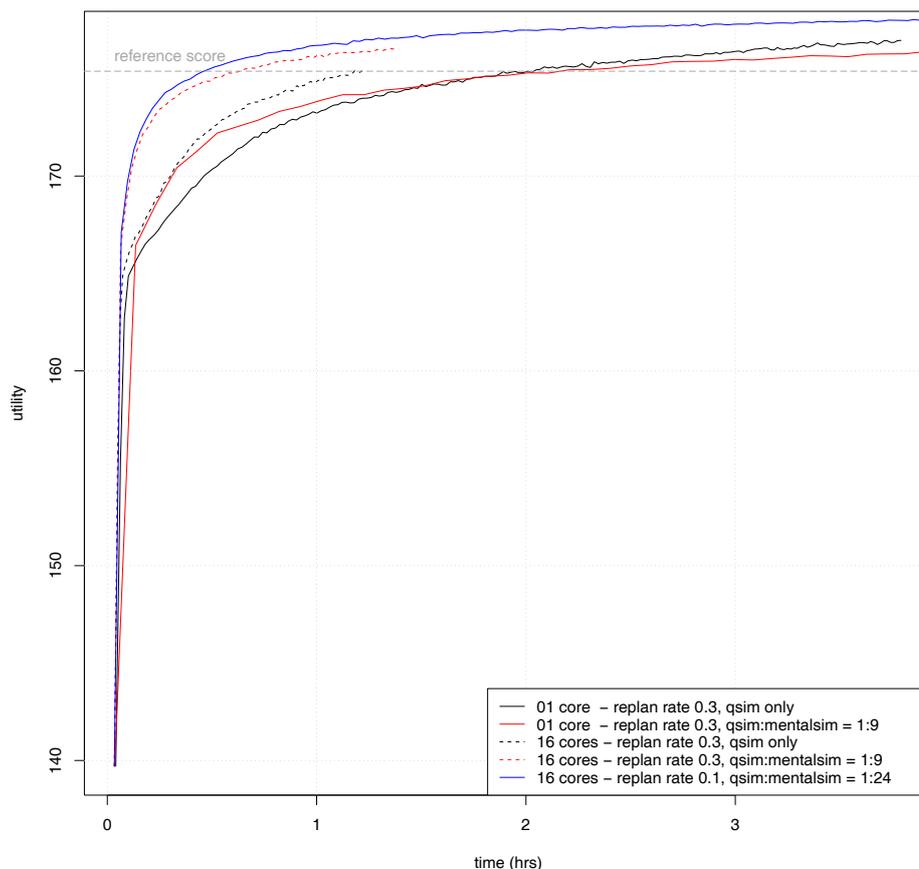


FIGURE 5 Score evolution vs time for large-scale scenario, comparing the influence of QSim:MentalSim ratio, number of computational cores and replanning rate (MentalSim module operating in global mode only).

1 simulation approach reduces the number of time-consuming QSim iterations required to achieve
 2 a given average plan score.

3 **Global vs. subset mode**

4 The reasoning behind subset mode was that agents could become ‘delusional’ by having plans
 5 with scores only from MentalSim. This ‘delusion’ was expected to manifest itself by too many
 6 of them changing their travel routes and timing, with no interaction to give congestion, with the
 7 result of just moving the congestion around in the network. This does not seem to be the case
 8 for the scenario used in these experiments.

9 An alternative reasoning might be that if MentalSim produces reasonably accurate events
 10 from good travel time information, and the number of MentalSim iterations in the inner loop
 11 are kept to a reasonable number for the given replanning rate, then global mode has the effect
 12 of keeping all plan scores in an agent’s memory at likely values for the current QSim iteration,
 13 rather than some earlier simulation state.

14 For the subset strategy, the number of plans in an agents’ memory grows with increasing
 15 QSim iterations at the same expected rate as for a QSim-only run with the same replanning rate.

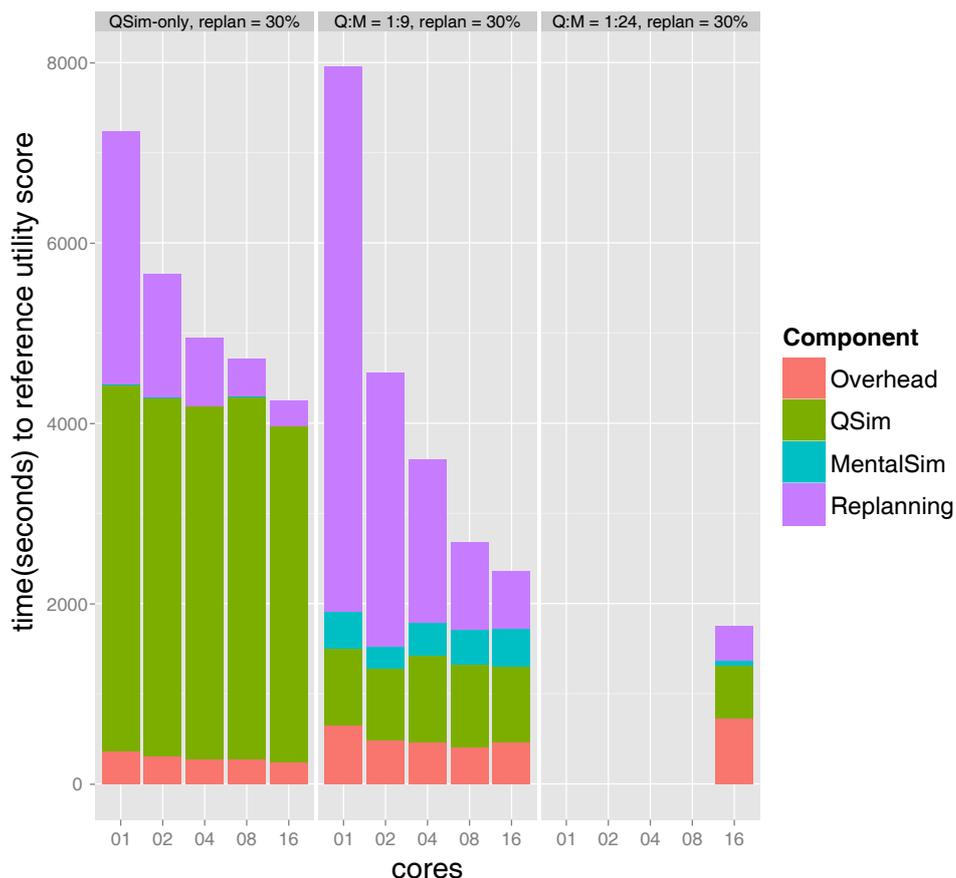


FIGURE 6 Computation time contributions vs number of cores for QSim only (0.3 replanning rate), QSim:MentalSim = 1:9 (0.3 replanning rate) and QSim:MentalSim = 1:24 (0.1 replanning rate) at the reference score (grey line in Figure 5).

1 The only difference is that the newly generated plans are expected to perform better than those
 2 generated by the QSim-only run, because they have been repeatedly adjusted and evaluated
 3 based on the last QSim’s travel times. But, supposing an agent memory of five plans, it will still
 4 take any agent at least five QSim iterations to reach their memory limit. It also means that, at
 5 any time, at least one plan in the agent’s memory will have been scored on information that is at
 6 least five QSim iterations old.

7 For global mode, agent memories grow much more rapidly with increasing QSim iterations,
 8 and a large proportion of their plans will have been evaluated using the most recent travel time
 9 information. Within a single generation of the outer loop, a large number of agents can be
 10 expected to have their memories filled. Consequently, for any given agent, it is unlikely that any
 11 given plan will have been scored on information that is more than one or two QSim iterations
 12 old.

13 As for ‘delusion’ — it appears that, as long as information is dependable, agent responses
 14 are, for most part, reasonable.

1 **Performance**

2 As expected, the mental simulation approach scales well with an increasing number of cores. Our
3 experiments revealed that the interplay of replanning rate and number of MentalSim iterations
4 in the inner loop have an important influence on convergence rate. Having a relatively low
5 replanning rate with a higher number of MentalSim iterations in the inner loop produces the
6 target score in less QSim iterations and less wallclock time.

7 At first glance, this is a surprising result, because the expected number of plans generated
8 from one QSim iteration to the next is comparable for the two 16-core mental simulation
9 runs in Figure 6. The first run has a replanning rate of 0.3 and QSim:MentalSim ratio of 1:9.
10 Consequently, in 1+9 iterations, the expected number of new plans produced per agent comes to
11 3, with a standard deviation of 1.44. In comparison, the second run has a replanning rate of 0.1
12 and QSim:MentalSim ratio of 1:24, so in 1+24 iterations, it produces only 2.5 new plans per
13 agent on average, with a standard deviation of 1.5.

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

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

CONCLUSION AND OUTLOOK

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

25 **Reducing overhead**

26 The next development step will be to integrate MentalSim into the core MATSim framework,
27 and reduce the number of overhead operations occurring between MentalSim iterations. These
28 operations include the calculation and writing out to disk of departure profiles, travel times and
29 log files, and are a significant contributor to total wallclock time, limiting the improvement gains
30 from parallelization.

31 **Public transport**

32 In this paper, public transport trips are not explicitly simulated in the QSim iterations, but instead
33 teleported through the network. Preliminary tests with mental simulation have shown promising
34 results for scenarios that explicitly simulate public transport in the presence of private vehicle
35 traffic (see 14), but further investigation is required.

36 **Social network coordination and ride-sharing**

37 The ultimate purpose of developing the mental simulation approach is to explore MATSim's
38 capability to test integrated, complex scenarios. If solution spaces are huge if agents replan
39 independently from each other, they become massively vast when one starts to consider the

1 possibilities that open up when plans are coordinated within households and social networks. So
2 far, only sub-problems of this integrated transport problem with social networks and ride sharing
3 have been addressed in the MATSim development context, using best-response replanning
4 modules employing complex metaheuristics. Mental simulation in combination with simple
5 replanning modules will be investigated as an alternative to the best-response approach.

6 **Parallel simulations**

7 The present paper inserts the MentalSim so that it stays close to the pre-existing simulation
8 logic. In global mode, the QSim is just replaced by the MentalSim; in subset mode, a sequence
9 of MentalSim runs produce a single plan for selected agents, to be executed in the next QSim
10 run. Even though performance gains are the result of the MentalSim module's capability to fully
11 exploit parallel computation, the simulation logic is still serial.

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

20 Once the MentalSim capability is integrated into MATSim, however, this opens the door to
21 more sophisticated approaches. For example, an agent could mentally optimize a public transit
22 plan over many mental iterations and only then compare it to an already optimized car plan.

23 Also, such optimizations could run in parallel when computing resources are under-utilized
24 during QSim runs (recall that the queue simulation is more reliable and easier to maintain in
25 single-threaded mode).

26 Extending this idea, one could imagine a situation where several instances of an agent
27 population are run in parallel, each with different initial conditions. Information on plan
28 performance across different runs could then be monitored by a supervisory process; taking the
29 best plans from these parallel threads and putting them together in a primary simulation.

ACKNOWLEDGEMENTS

30 The authors thank Dominik Grether from VSP, TU Berlin for his technical assistance during
31 development of the MentalSim modules. This work is funded, in part, by a grant from the
32 National Research Fund of Singapore.

REFERENCES

- 33 1. Balmer, M., M. Rieser, K. Meister, D. Charypar, N. Lefebvre and K. Nagel (2009) MATSim-
34 T: architecture and simulation times, *Multi-Agent Systems for Traffic and Transportation*
35 *Engineering*, 57–78.
- 36 2. Goldberg, D. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*,
37 Addison-Wesley.
- 38 3. Hraber, P., T. Jones and S. Forrest (1994) The ecology of Echo, *Artificial Life*, **3** (3) 165–190.

- 1 4. Balmer, M. (2007) Travel demand modeling for multi-agent transport simulations: Algorithms and systems, Ph.D. Thesis, Swiss Federal Institute of Technology (ETH) Zürich,
2 Switzerland.
3
- 4 5. Meister, K., M. Balmer, K. Axhausen and K. Nagel (2006) planomat: A comprehensive
5 scheduler for a large-scale multi-agent transportation simulation, paper presented at the *6th*
6 *Swiss Transport Research Conference*, Monte Verita, Ascona.
- 7 6. Horni, A., D. M. Scott, M. Balmer and K. W. Axhausen (2009) Location choice modeling
8 for shopping and leisure activities with MATSim, *Transportation Research Record: Journal*
9 *of the Transportation Research Board*, **2135** (-1) 87–95.
- 10 7. Dubernet, T. and K. Axhausen (2012) Including joint trips in a multi-agent transport
11 simulation, paper presented at the *6th Swiss Transport Research Conference*, Monte Verita,
12 Ascona.
- 13 8. Arentze, T. and H. Timmermans (2001) Inductive learning approach to evolutionary decision
14 processes in activity-scheduling behavior: theory and numerical experiments, *Transporta-*
15 *tion Research Record: Journal of the Transportation Research Board*, **1752** (-1) 1–7.
- 16 9. Rieser, M., K. Nagel, U. Beuck, M. Balmer and J. Rumenapp (2007) Truly agent-oriented
17 coupling of an activity-based demand generation with a multi-agent traffic simulation,
18 *Transportation Research Record*, **2021**, 10–17.
- 19 10. Nicolai, T. W., L. Wang, K. Nagel and P. Waddell (2011) Coupling an urban simulation
20 model with a travel model – A first sensitivity test, paper presented at the *Computers in*
21 *Urban Planning and Urban Management (CUPUM)*, Lake Louise, Canada. Also VSP WP
22 11-07, see www.vsp.tu-berlin.de/publications.
- 23 11. Waddell, P., A. Borning, M. Noth, N. Freier, M. Becke and G. Ulfarsson (2003) Microsimu-
24 lation of urban development and location choices: Design and implementation of UrbanSim,
25 *Networks and Spatial Economics*, **3** (1) 43–67.
- 26 12. Charypar, D. and K. Nagel (2005) Generating complete all-day activity plans with genetic
27 algorithms, *Transportation*, **32** (4) 369–397, ISSN 0049-4488.
- 28 13. Rieser, M., D. Grether and K. Nagel (2009) Adding mode choice to a multi-agent transport
29 simulation, *Transportation Research Record: Travel Demand Forecasting 2009*, **2132**,
30 50–58.
- 31 14. Rieser, M. and K. Nagel (2009) Combined agent-based simulation of private car traffic
32 and transit, paper presented at the *Proceedings of The 12th Conference of the International*
33 *Association for Travel Behaviour Research (IATBR)*, Jaipur, India. Also VSP WP 09-11, see
34 www.vsp.tu-berlin.de/publications.