

An improved replanning module for agent-based micro simulations of travel behavior

**Konrad Meister
Michael Balmer
Kay W. Axhausen**

Arbeitsberichte Raum- und Verkehrsplanung

**303
July 2005**

Contents

1	Introduction	1
2	Micro simulation framework	1
2.1	Activity plans	2
2.2	Iterations	2
2.3	Scoring	3
3	Methods	4
3.1	Old time allocation mutator	4
3.2	Implementation	5
3.3	Travel times	6
4	Results	7
4.1	Setup	7
4.1.1	Canton Zurich case study	7
4.1.2	Activity parameters and constraints	8
4.1.3	GA parameters	9
4.2	Stand-alone tests	9
4.3	Test as external strategy module	10
5	Discussion and outlook	13

List of Figures

1	Example activity plan	3
2	Change of score distribution with setup <i>population size=50; no evolution</i>	11
3	Relative change of scores with <i>population size=50; different evolution settings</i>	12
4	Comparison of average scores	13
5	Comparison of departure time choices by activity chain	14
6	Comparison of arrival time choices by activity chain	15

List of Tables

1	Activity parameter values	8
2	Opening hours as temporal constraints	9
3	Performance of different GA setups [replanned agents/s]	10

An improved replanning module for agent-based micro simulations of travel behavior

Konrad Meister	Michael Balmer	Kay W. Axhausen
IVT	IVT	IVT
ETH Hönggerberg	ETH Hönggerberg	ETH Hönggerberg
CH-8093 Zürich	CH-8093 Zürich	CH-8093 Zürich
Phone: +41-44-633 30 88	Phone: +41-44-633 27 80	Phone: +41-44-633 39 43
Fax: +41-44-633 10 57	Fax: +41-44-633 10 57	Fax: +41-44-633 10 57
meister@ivt.baug.ethz.ch	balmer@ivt.baug.ethz.ch	axhausen@ivt.baug.ethz.ch

July 2005

Abstract

An external strategy module for an agent-based micro simulation of traffic systems is presented. It modifies activity durations and departure times of activity plans, which are the agent-based representation of travel demand. The module combines broad search for alternative timing decisions with an optimization procedure of a utility function. The idea is to replace a replanning module that changed timing decisions randomly. Main results are relaxation of the whole simulation system to a better stationary state, and much quicker convergence. The difference in overall performance compared to the previous implementation of the replanning module is one order of magnitude.

Keywords

planomat, MATSIM, agent-based micro simulation of traffic systems

Preferred citation style

Meister, K., M. Balmer and K. W. Axhausen (2005) An improved replanning module for agent-based micro simulations of travel behavior, *Working paper*, **303**, Institute for Transport Planning and Systems (IVT), ETH Zurich, Zurich.

An improved replanning module for agent-based micro simulations of travel behavior

Konrad Meister	Michael Balmer
IVT	IVT
ETH Hönggerberg	ETH Hönggerberg
CH-8093 Zürich	CH-8093 Zürich
Tel: +41-44-633 30 88	Tel: +41-44-633 27 80
Fax: +41-44-633 10 57	Fax: +41-44-633 10 57
meister@ivt.baug.ethz.ch	balmer@ivt.baug.ethz.ch

July 2005

Zusammenfassung

Hier wird ein externes Modul für die agentenbasierte Simulation von Verkehrssystemen vorgestellt. Es verändert die Aktivitätendauern und Abfahrtszeiten in Aktivitätsplänen, der agentenbasierten Form der Verkehrsnachfrage. In diesem Modul wird eine umfassende Suche im Lösungsraum mit Optimierung einer Nutzenfunktion kombiniert. Das Ziel der Arbeit ist die Verbesserung eines bestehenden Moduls, welches zufällig neue Zeitinformationen wählt. Durch das neue Modul wird ein besserer Fixpunkt des Systems als vorher gefunden, ausserdem konvergiert es um eine Grössenordnung schneller.

Schlagworte

planomat, MATSIM, Mikrosimulation, Verkehrsmodell

Bevorzugter Zitierstil

Meister, K., M. Balmer und K. W. Axhausen (2005) An improved replanning module for agent-based micro simulations of travel behavior, *Arbeitsberichte Raum- und Verkehrsplanung*, **303**, Institut für Verkehrsplanung und Transportsysteme (IVT), ETH Zürich, Zürich.

1. Introduction

MATSIM is an iterative, agent-based micro simulation of traffic systems (Raney and Nagel, 2005). It mainly consists of a microscopic traffic flow simulation on one side and different modules adapting travel demand to generalized travel costs on the other side. They are called alternately until the system reaches its stationary state, which corresponds to user equilibrium in the case of traffic systems. In MATSIM, travel demand is represented by individual agents that follow an activity plan. Each activity plan is assigned a score. The higher the score, the better is the plan. Convergence to the stationary state is, among other measurements, judged by the development of the score aggregated over the whole agent population.

This paper is about *planomat*, a flexible module which adapts the activity plans to travel times the agent experiences during the subsequent simulations of traffic flow. Since changing generalized costs of travel affect each aspect of travel demand, it would be desirable that this module was as comprehensive, allowing for choice of

- activity durations,
- departure times,
- activity locations,
- trip modes,

and other desired attributes. In a first implementation, *planomat* adapts activity durations and the departure time of the first trip, which is seen as the start of the day. The motivation was to replace an existing "dummy" module which produces unsatisfying results (the so-called *time allocation mutator*). The main ideas for *planomat* are more exhaustive exploration of the search space and the use of a scoring function for goal-oriented search for alternate plans.

The paper is structured as follows. In section 2 aspects of the micro simulation framework are described that are necessary to understand the *planomat* functionality. The methods used for plan adaptation are described in section 3. Section 4 presents some results of the module stand-alone and in the micro simulation framework.

2. Micro simulation framework

In this section, the concepts required for understanding the *planomat* functionality are described shortly. For a comprehensive and more detailed framework description, see

(Raney and Nagel, 2005).

2.1 Activity plans

The representation of an agent's travel demand is an activity plan, an alternating sequence of *activities* and *trips*. As shown in the example in Figure 1, the framework uses XML to store and exchange plans.

person Each person is identified by an `id` by which its socio-economic attributes can be found in the synthetic population. A person holds several plans.

plan Each plan can be assigned a `score` according to a scoring function (see section 2.3). The attribute `selected="yes"` states that the plan was chosen for execution in the previous iteration of the traffic flow simulation.

activities Each activity `<act>` is characterized by a type, a hectare-based location coordinate, an associated network link, and its temporal extent defined by two of three attributes `start_time`, `end_time`, and `dur` (activity duration). The start of the plan is defined as the end time of the first activity, in this case `07:35:04`. A plan can be interpreted as a 24-hour wrap-around, so the end of the plan is also at `07:35:04` the next day. In the example shown, first and last activity are the same *home* activity (h).

trips The attributes of a trip `<leg>` include a mode, a departure time and a duration. Furthermore, it is characterized by a route.

2.2 Iterations

As pointed out, an iterative approach is used to find the stationary state in the dynamic traffic system. The overall simulation consists of the following steps:

1. *Initial plans* have to be generated as a first input to the traffic flow simulation. It contains all the assumptions about the agents' personal attributes, as well as approximations for the plan attributes, e.g. free speed travel times as an approximation for the trip duration. For each agent, a set of plans is generated and stored in the agent database.
2. The *plan selection mechanism* of the agent database chooses one plan per agent for execution.

Figure 1: Example activity plan

```
<person id="22018">
  <plan score="182.326" selected="yes">
    <act type="h" x100="703600" y100="236900" link="5757" end_time="07:35:04" />
    <leg num="0" mode="car" dep_time="07:35:04" trav_time="00:16:31">
      <route>1900 1899 1897</route>
    </leg>
    <act type="1" x100="702500" y100="236400" link="5749" dur="02:00" />
    <leg num="1" mode="car" dep_time="09:51:35" trav_time="02:26:36">
      <route>1899 1848 1925 1924 1923 1922 1068</route>
    </leg>
    <act type="s" x100="681450" y100="246550" link="2140" dur="02:00" />
    <leg num="2" mode="car" dep_time="14:18:11" trav_time="00:44:35">
      <route>1067 1136 1137 1921 1922 1923 1924 1925 1848 1899</route>
    </leg>
    <act type="h" x100="703600" y100="236900" link="5757" />
  </plan>
</person>
```

3. The *simulation of traffic flow* executes the plans, that is it "moves" agent objects through a model of the traffic network when trips are planned. The result of this are new travel times for each trip (attribute `trav_time` of element `<leg>`). Additionally, the plans executed are scored (see section 2.3). If the stop criterion of the simulation is met, go to step 6.
4. A subset of the agents is chosen for plan modification by so-called *external strategy modules*. These modules, of which *planomat* is one, can capture one or more travel behavior attributes. Currently, 10% of all agents are considered for replanning and rerouting respectively.
5. The new plan is stored in the agent database. Return to step 2.
6. End of the simulation.

2.3 Scoring

The idea was to write a goal-oriented module, so a scoring function is needed to give scores to the adapted plans. Currently, a utility function is used. It is described in detail in (Charypar and Nagel, 2005), a summary and the set of parameters currently used is presented in (Raney and Nagel, 2005).

Basically, recognize that while performing activities is rewarded, travel, early arrival, and late arrival are punished. The utility function essentially translates the layout of the plan into a numerical value, which can be thought of as a score or an actual value in monetary terms.

3. Methods

This section starts with a description of the functionality and the shortcomings of the module to be replaced. After that, the details of the current *planomat* implementation is described in detail.

3.1 Old time allocation mutator

(See the first two paragraphs of this section also in (Raney and Nagel, 2005)).

The old replanning module *time allocation mutator* takes the existing times of the plan and modifies them randomly. Note that there is no "goal" with this module, that is, the module does not try to improve any kind of score. Rather, the module makes a random modification, and the plans selection mechanism in conjunction with the scoring will make the agents improve toward better scores.

The exact details of the time mutator are as follows. This module reads the plans file, and for each plan alters the end time of the first activity by a random amount r_1 uniformly selected in the range $r_1 \in [-30 \text{ min}, 30 \text{ min}]$. Values that come before 00:00 (midnight) are reset to that time. It then alters the duration of each activity except the first and last by separate random values uniformly selected from the same range. The last activity does not need modification since it runs from whenever the agent arrives until 24:00 (midnight). The modified plans are written back out to a file.

Simulations with the *time allocation mutator* show that the system converges despite the random nature of time information mutation. This is due to the learning framework of the simulation which keeps good plans in its "brain" while discarding others. However, two problems arise. First, visual inspection of departure time distribution shows that the stationary state found cannot be the global optimum if the initial plans are not close to their optimal states (see Figures 5 and 6). This is because of the insufficient exploration within $\pm 30 \text{ min}$, although good activity durations and start times may be hours away from the initial solution. Second, the convergence speed (towards an optimum which is not the best possible) is unsatisfying. More, visual inspection of the average fitness tells it is still rising after >1000 iterations, which is far too much for any practical use (see Figure 4). Simple extension of the search range, e.g. $\pm 6 \text{ h}$ for all time information,

would probably find a better optimum. Then, a multiple of the number of iterations was needed since the search space would be fully enumerated by the replanning module. This search space is only slightly reduced by the learning framework because it keeps a small number of solutions (currently 6 alternative plans).

3.2 Implementation

The idea is now to use a scoring function to reduce the search space substantially. Here, the same utility function as in the agent database is used (see section 2.3).

For several reasons, the decision was made to use a Genetic Algorithm (GA) to find good solutions in the sense of the utility function:

Experience The GA method proved to be successful in various experiments for activity plan generation for individual agents or households ((Charypar and Nagel, 2005), (Meister *et al.*, 2005), (Schneider, 2003)). This is the first attempt to integrate this approach into a multi-agent simulation system.

Flexibility In the current setup of the module, a better time allocation could be much easier calculated. GAs are not the best choice to solve continuous problems like this, they were designed to rather solve combinatorial problems. A gradient-based optimization procedure would probably be much faster. However, the goal is to extend *planomat* to a comprehensive replanning module incorporating many aspects of travel demand. Location choice, mode choice and the choice of the activity pattern are such combinatorial problems, which are meant to be included later.

The exact details of *planomat* are as follows. After the plans are read in, the GA creates a number of alternatives for each single plan.

Mutation of start time The end time of the first activity is mutated by an amount s uniformly selected from range $s \in [-12 h, 12 h]$. This is the extension of the search space mentioned earlier. Values earlier than 00:00 and later than 24:00 are mapped to a time value between these limits of the day. This procedure equals a uniform choice all over the day.

Mutation of activity duration For each activity, a new duration is determined by multiplying the initial value with a separate factor $d = e^X$ with X being uniformly selected from the range $X \in [-0.5, 0.5]$.

The crossover operator recombines two existing plans to a new one by randomly choosing start time and activity durations from one of the parents.

The mutation operator is implemented similar to the alternative generation in the beginning, with a smaller search range parameterized with the *mutation probability* p_{mut} . When mutating a plan, with a probability of p_{mut}

- a new start time is chosen by adding an amount s uniformly selected from range $s \in [p_{mut} \cdot -12h, p_{mut} \cdot 12h]$, and
- an activity duration is multiplied with a factor $d = e^X$ with X being uniformly selected from the range $X \in [-p_{mut}/2, p_{mut}/2]$.

3.3 Travel times

The current shape of the utility function penalizes travelling linearly with the travel time. There are many possibilities to determine a travel time, dependent on the assumptions about the agents' knowledge.

Experienced agent The easiest way is simply to maintain the travel time which was the result of the previous iteration of the traffic flow simulation. This corresponds to the assumption that the agent uses his (most recent) experiences.

No knowledge One also could say that the expected travel time will be different, e.g. because the trip is likely to take place at another time of day. An approximation could be used here, e.g. travel time calculated with Euclidean distance between the two associated activities, or free speed network travel time.

Global knowledge The output of the simulation allows the construction of travel time trajectories for each link over the day. The agent could be provided this knowledge or parts of it. So it may consider traffic situations it didn't experience itself. This method could be used to e.g. evaluate the influence of traffic management systems.

We decided to use the recently experienced travel times. The inclusion of knowledge has the same reason as the decision to use a utility function: The stationary state, that is the user equilibrium, might be found anyway because of the learning framework of the agent database. But as we want the simulation to quickly converge, the utility function is now used twice: In the framework and here, in the *planomat*. Now, there is double chance that good plans are selected and bad plans dropped.

The other option is to include globally available knowledge. We didn't investigate this alternative very much, mainly because much more information would have to be processed. Also, we wanted to see the isolated effect of the quite obvious extension with the utility function first. It would be unrealistic to provide all experienced travel times

of all agents to all replanning agents, so a concept of information usage would have to be integrated here: Which traffic information is transmitted on the radio? How do people respond to these information?

Sometimes, the duration of one or more trips in the given plan is not defined. When a gridlock occurs in the system, agents stay stuck in the network. The simulation cannot resolve the gridlock in a "natural" way because of the absence of the opportunity for the agents to spontaneously change the route or the trip destination (no within-day-replanning). In order to solve the problem, agents are simply taken "out" of the simulation and therefore have an incomplete plan with some travel times undefined. In this case, the *planomat* makes an Euclidean distance approximation of the travel time, with a fixed average speed of 20 km/h.

4. Results

4.1 Setup

The setup includes a regional definition of the study area, the demand generation process, the specification of the traffic network and a list of assumptions about activity-related behavior as well as temporal constraints.

4.1.1 Canton Zurich case study

The case study used for testing the *planomat* is a simulation of the Greater Zurich area. It is described in detail in (Balmer *et al.*, 2005), a quick overview follows here.

First, a synthetic population of the Canton Zurich is generated, using data from the Swiss National Population Census. It is a list of approx. 1'200'000 agents with individual attributes like age or sex, and a hectar-based home location. Each agent is assigned an activity chain based on the Swiss Microcensus on travel behavior. For this assignment process, of the two methods proposed in (Balmer *et al.*, 2005) the one without using OD-Matrices is used. These activities are distributed in space by several location choice modules. The network used for the assignment with a microscopic traffic flow simulation is the Swiss National Traffic Network model.

For test reasons, the traffic of only a 1% sample of the whole agent population is simulated. Therefore, the network capacity was reduced to still produce some congestion and sensitivity of timing decisions to experienced travel times.

Table 1: Activity parameter values

Activity type	abbreviation	d_0 [h]	d_{min} [h]	T_{start}	T_{end}
home	h	12	8	—	—
work	w	8	6	9:00	—
work1	w1	4	2	—	—
work2	w2	4	2	—	—
work3	w3	8	6	—	—
education	e	6	4	9:00	—
education1	e1	3	1	9:00	—
education2	e2	3	1	—	—
education3	e3	6	4	—	—
shop	s	2	1	—	—
leisure	l	2	1	—	—

The different work and education activity types can be explained as follows. If an activity chain includes two *work* or *education* activities, it is assumed that their typical activity duration is half the complete-activity duration and will be renamed *work1* and *work2* resp. *education1* and *education2*. An example would be h-w1-l-w2-h. If a work or education activity is not the first in the activity chain, it is renamed *work3* or *education3* without the desired start time at 9:00, but all other attributes equal. An example of that would be h-s-w3-h.

4.1.2 Activity parameters and constraints

The scoring function requires several parameters, either activity or location specific.

Each activity is characterized by a typical duration d_0 , a minimum duration d_{min} and desired start/end times T_{start} , T_{end} . While the typical duration is a mandatory parameter to the utility function, desired time windows are optional. Table 1 is a list of parameter values used in this scenario.

Furthermore, there exist temporal constraints for the execution of activities, represented here by opening hours. An agent will fail to perform an activity outside these opening hours, and will have to wait instead. In this case, he doesn't gain any score. The temporal constraints are an attribute of a specific facility. In this setup, they are the

Table 2: Opening hours as temporal constraints

Activity type	opening time	closing time
home	—	—
work* (w, w1, w2, w3)	7:00	18:00
education* (e, e1, e2, e3)	7:00	18:00
shop (s)	8:00	20:00
leisure (l)	6:00	24:00

same all over the modelled region because more detailed data about opening hours was not available yet. This is why they appear activity-specific in Table 2.

4.1.3 GA parameters

From the many GA control parameters, *population size* and *number of generations* were varied to test the result sensitivities:

population size 20/50

number of generations 0 (no evolution), 100, 1000

Results are presented for the setups with *population size* = 50. The mutation probability is set to $p_{mut} = 0.3$.

4.2 Stand-alone tests

This section describes a simple test of the module as a stand-alone program. A plans file with about 12'000 agents is read in, modified, and written out again. The plans used as input are the result after the first iteration of the traffic flow simulation, that means, still with many opportunities for the agents to increase their score.

Figure 2 shows a before/after-comparison of the score distributions, with the setup *population size=50, no evolution* used. The peaks in the distribution represent different activity chains, which have different typical utility levels. The distribution is shifted to the right, as expected. The practical interpretation is that the agents were able to improve their score by adapting their time use to changed travel times.

Table 3: Performance of different GA setups [replanned agents/s]

population size	no evolution	100 generations	1000 generations
20	2000	720	120
50	1333	600	110

Figure 3 gives a closer look to the effect of different GA setups. With a population size of 50, results without evolution, 100 new plans and 1000 new plans per agent are compared. The relative score change is classified in 1% steps. It can be seen that there are only small differences in the resulting average fitness. While *no evolution* and *100 generations* produce plans worse than the input plan for some agents, the *1000 generations* configuration exclusively produces better plans. This is a very minor advantage considering the much higher computing time (see Table 3).

The before/after-comparison demonstrates the functionality of *planomat*. However, two different kinds of activity plans are compared here. The input plan is a "real world" result, the outcome of an assignment of a given demand distribution. The output of *planomat* can be interpreted the agent's "desired" plan under expected conditions (fixed travel times). This illustrates the "working file" character of an activity plan: Some attributes may be defined, some not. Some attributes are assignment results, some are the agent's desires or expectations.

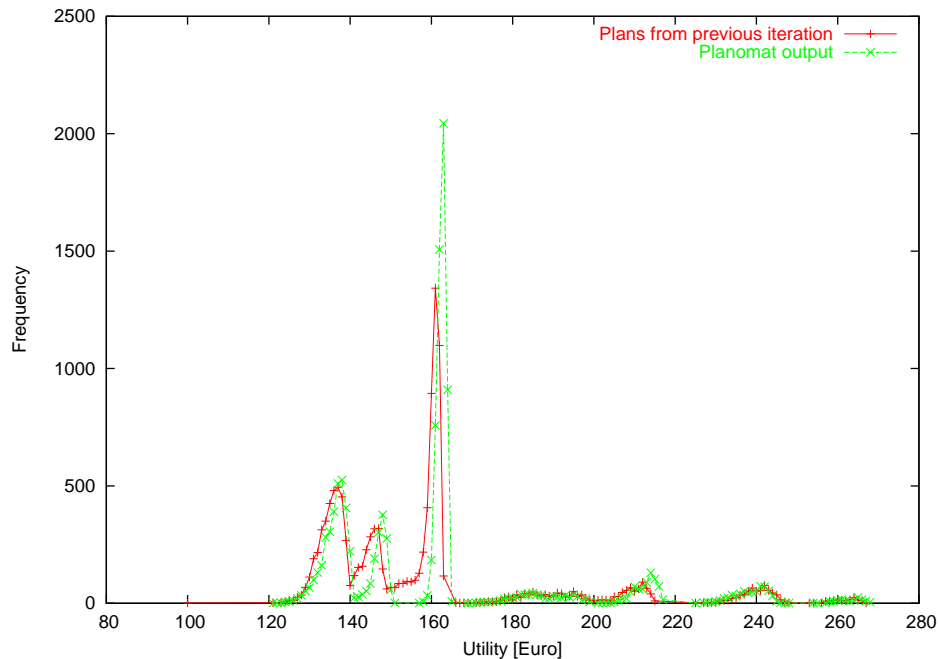
4.3 Test as external strategy module

The results presented in this section recur on the problems that arose with the usage of the simple "dumb" replanning module (compare section 3.1). This is why each results figure is of a *planomat* vs. *time allocation mutator* kind.

The setup used for the results presented here is *popsize=50, no evolution*. That means, for each agent, 50 alternatives for his most recent plan are generated. From these, the best one is returned. We also tried the setup *popsize=50, 1000 generations*, but the results showed almost no difference (see also Figure 3 for the minor differences).

Figure 4 shows the development of the average score across the whole agent population. Both curves show tendency towards a limiting value. While this convergence is clear to see for the *planomat* setup after 400 iterations at a value of ca. 157.2, the *time allocation mutator* configuration has a rising curve even after >1000 generations. It is nor clear if it converges to the same value.

Figure 2: Change of score distribution with setup *population size=50; no evolution*



Figures 5 and 6 show the departure and arrival time distributions at the beginning of the simulation and after 400 generations, for each setup. The main difference is in the distributions for the *h-l-h*, *h-l-l-h* and *h-l-s-l-h* type activity chains. While in the *time allocation mutator* setup, most leisure activities take place in the morning, the *planomat* distributes them all over the day with a peak in the evening. The latter is the expected result, since *leisure*-type activities are only constrained within 6:00 and 24:00, and the network is loaded least in the evening. The same effect can be observed for the shopping-dominated activity chains *h-s-h* and *h-s-s-h*, where the departure / arrival times distribute in the opening hour window 8:00 - 20:00. Furthermore, the afternoon commuter peak is more pronounced (regard activity chains *h-w-h*, *h-w-l-w-h*, *h-w-s-w-h* and *h-w-w-h*).

The suboptimal distribution in the *time allocation mutator* setup doesn't change after >1000 generations (not shown). We think that this is the reason for the suboptimal average score development. It has its cause in the insufficient exploration of timing alternatives only within $\pm 30 \text{ min}$. It is probable that the better solution could have been found with the "dumb" module also if the initial distribution of departure times and activity durations had been closer to the stationary state. But as we lacked data about realistic distributions, we assumed a uniform choice of departure time between 6:00 and 8:00 in the morning.

Figure 3: Relative change of scores with *population size=50*; different evolution settings

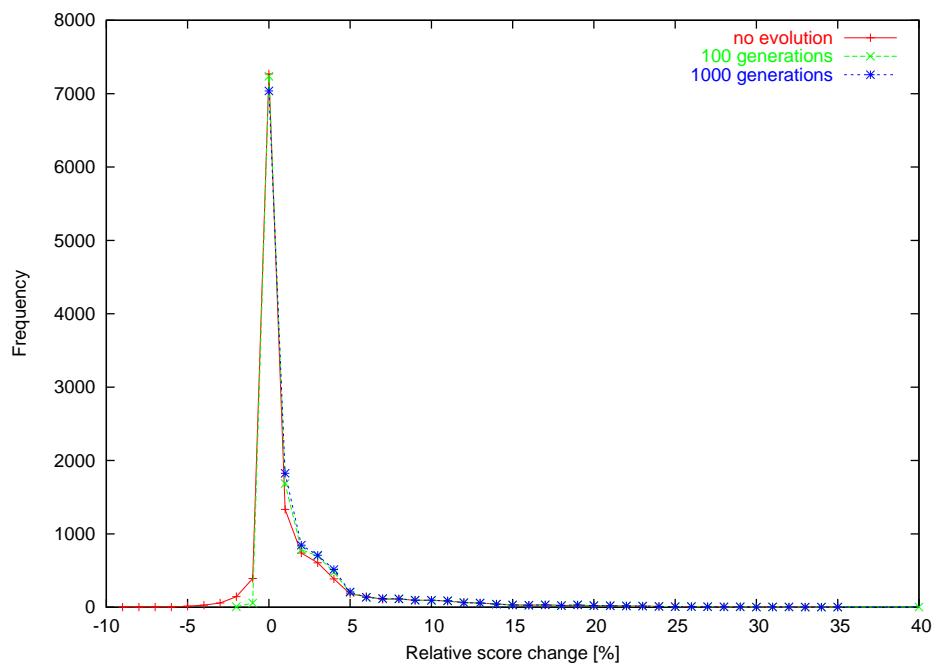
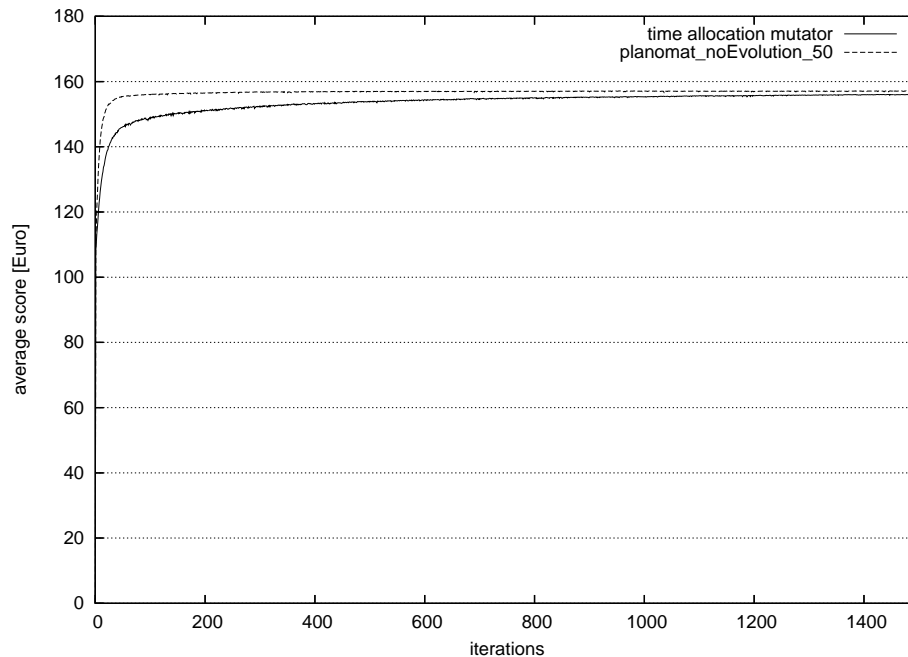


Figure 4: Comparison of average scores



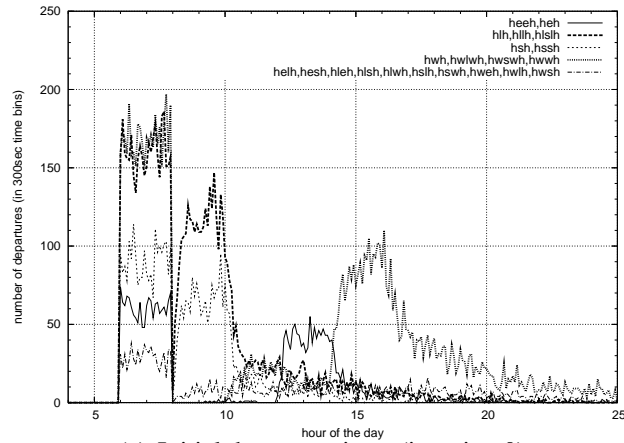
5. Discussion and outlook

An external strategy module for an agent-based micro simulation of traffic systems was presented. It modifies activity durations and departure times of activity plans, which are the agent-based representation of travel demand. The module combines broad search for alternative timing decisions with a goal-oriented search using a utility function. The idea was to replace a replanning module that changed timing decisions randomly. Main results are relaxation of the whole simulation system to a better stationary state, and much quicker convergence. The difference in overall performance compared to the previous implementation of the replanning module is one order of magnitude.

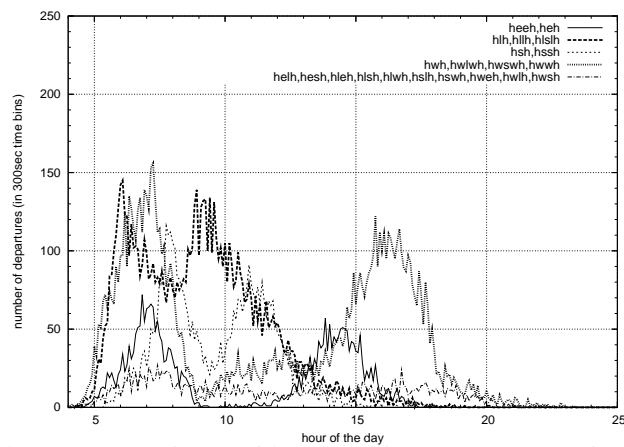
The results are very encouraging since convergence can be observed after an acceptable number of iterations (400). Knowing that, the simulation could be cancelled after <100 iterations. In the case of the 1% sample simulated, this equals an overall runtime of max. 15 hours, which has to be compared to week-long durations before.

The *planomat* is just one part in the whole micro simulation system. Next, tests have to be run with the full synthetic population. Initial travel demand is setup for two different regions, the Greater Areas of Zurich and Berlin (Balmer *et al.*, 2006). It is then possible

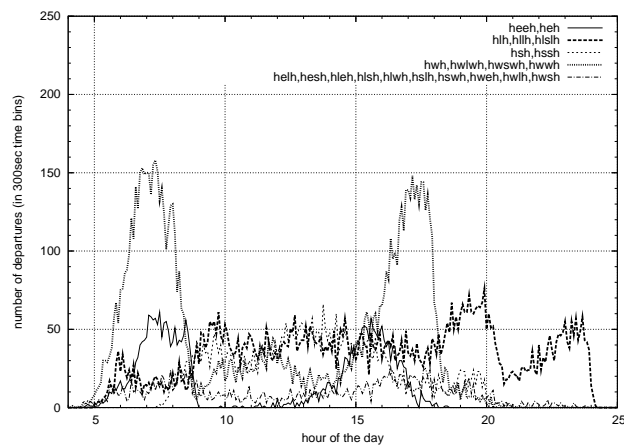
Figure 5: Comparison of departure time choices by activity chain



(a) Initial departure times (iteration 0)

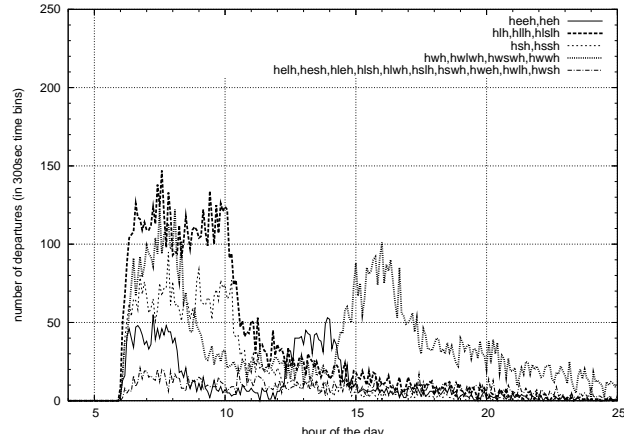


(b) Departure times with *time allocation mutator* (iteration 400)

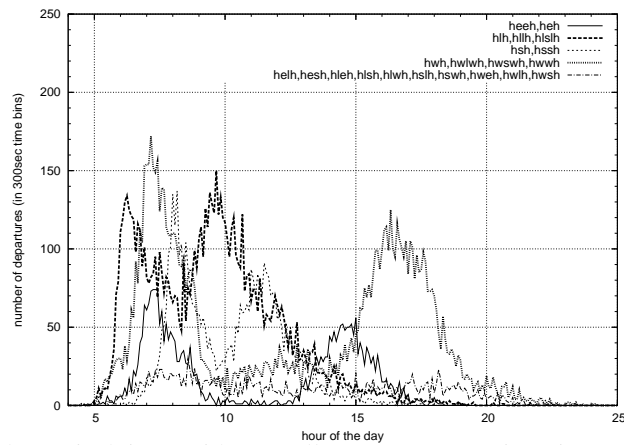


(c) Departure times with *planomat* (iteration 400)

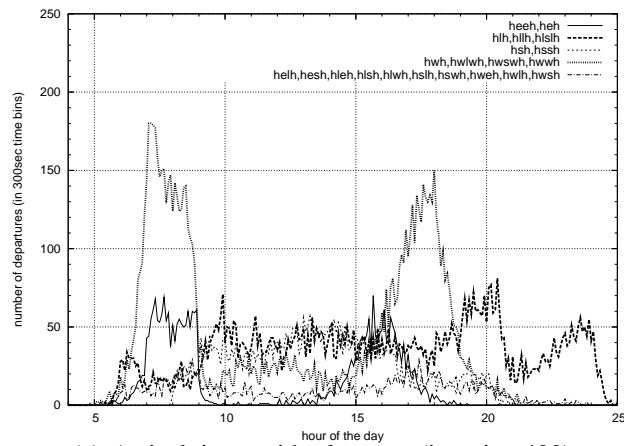
Figure 6: Comparison of arrival time choices by activity chain



(a) Arrival times in initial plans



(b) Arrival times with *time allocation mutator* (iteration 400)



(c) Arrival times with *planomat* (iteration 400)

to validate the departure/arrival time distributions against data from counting stations.

The vision for *planomat* is to include adaptation of further attributes of travel demand, like location choice or mode choice. Their introduction will not be an easy task as the timing choices performed currently, since traffic situations the agent didn't experience itself have to be evaluated. This requires the usage of globally available knowledge from the so called *events file* produced by the traffic flow simulation. In this case of a comprehensive replanning module, the evolution part of the GA will become more important than it is now, because the number of dimensions of the search space increases and combinatorial variables are well-suited problems for a GA.

Computing performance can be seen as sufficient. The number of 1333 replanned agents/s was achieved on a Pentium IV Xeon 2.4 GHz machine. For 10% of the modal split share of the Canton Zurich synthetic population, which are $\approx 56'000$ agents, this results in a run time of ≈ 45 s.

References

- Balmer, M., K. W. Axhausen and K. Nagel (2006) A demand generation framework for large scale micro-simulations. Submitted to Transportation Research Record.
- Balmer, M., M. Rieser, A. Vogel, K. W. Axhausen and K. Nagel (2005) Generating Day Plans Based on Origin-Destination Matrices, in *STRC 2005 Conference Proceedings*, Swiss Transportation Research Conference, Monte Verità, Ascona, Switzerland, March, 9-11 2005.
- Charypar, D. and K. Nagel (2005) Generating complete all-day activity plans with genetic algorithms, *Transportation*, **32** (4) 369–397.
- Meister, K., M. Frick and K. W. Axhausen (2005) A GA-based household scheduler, *Transportation*, **32** (5) 473 – 494.
- Raney, B. and K. Nagel (2005) An improved framework for large-scale multi-agent simulations of travel behavior, in *TRB 2005 Annual Meeting CD-ROM*, Transportation Research Board, Washington, D.C., January 9-13 2005.
- Schneider, A. (2003) Genetische Algorithmen zur Optimierung von Tagesplänen für Verkehrsteilnehmer. Term project, Computer Science, ETH Zürich.