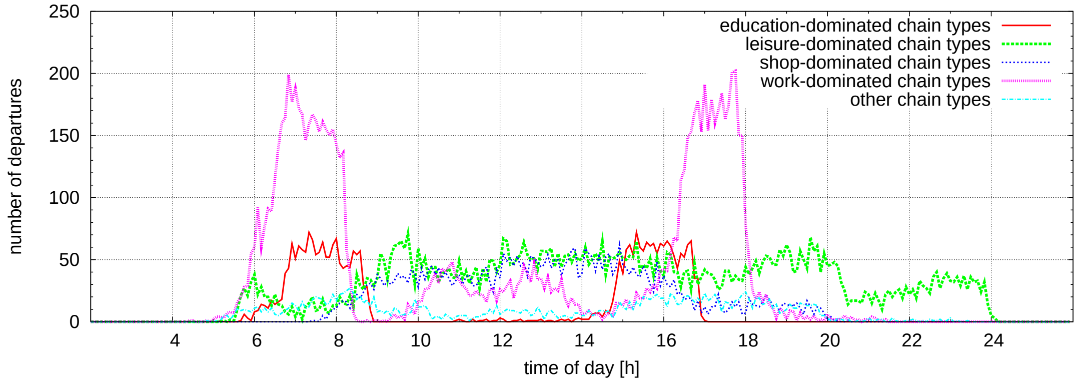


planomat:
A comprehensive scheduler for a large-scale multi-agent transportation simulation

Konrad Meister, IVT, ETH Zurich
Michael Balmer, IVT, ETH Zurich
Kay W. Axhausen, IVT, ETH Zurich
Kai Nagel, VSP, TU Berlin

Conference paper STRC 2006

STRC | **6th Swiss Transport Research Conference**
Monte Verità / Ascona, March 15-17, 2006

planomat:
A comprehensive scheduler for a large-scale multi-agent transportation simulation

Konrad Meister
IVT, ETH Zurich
ETH Hönggerberg, CH-8093 Zurich
Phone: +41-44-633 30 88
meister@ivt.baug.ethz.ch

Michael Balmer
IVT, ETH Zurich
ETH Hönggerberg, CH-8093 Zurich
Phone: +41-44-633 27 80
balmer@ivt.baug.ethz.ch

Kay W. Axhausen
IVT, ETH Zurich
ETH Hönggerberg, CH-8093 Zurich
Phone: +41-44-633 39 43
axhausen@ivt.baug.ethz.ch

Kai Nagel
VSP, TU Berlin
Salzufer 17-19, D-10587 Berlin
Phone: +49-30-314 23 308
nagel@vsp.tu-berlin.de

March 15 - 17, 2006

Abstract

An external strategy module for an agent-based micro simulation of traffic systems is presented. This module called *planomat* 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 for a scoring function that evaluates daily activity plans. The module is integrated into the existing framework MATSIM, which simulates traffic systems consisting of several 100’000 agents entirely on activity level. In this paper, a test version of the Canton Zurich is simulated, the biggest metropolitan area of Switzerland. Main results are relaxation of the whole simulation system to a better stationary state than in previous versions of the simulation framework. This is shown by departure/arrival time distributions. The number of required iterations was significantly reduced to 100, which is one-two orders of magnitude better than before.

Keywords

planomat, scoring function, time allocation, Large-scale agent-based micro-simulation

1. Introduction

MATSIM is an iterative, agent-based micro-simulation framwork of traffic systems (Raney and Nagel, 2005). It mainly consists on one side of a simulation of traffic flow and on the other side of different modules adapting travel demand to generalized travel costs. 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 (this is why it is called a micro-simulation framework). 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, modes, and other desired attributes. In a first implementation described here, *planomat* adapts activity durations as well as the trip departure times.

The motivation for this work was to replace an existing "dummy" module called *time allocation mutator* which produces unsatisfying results (see Raney and Nagel, 2005, p. 16). The main improvements in *planomat* are the exploration of the complete search space (instead of only a small part) and the use of a scoring function for goal-oriented search for alternate plans (instead of random strategy generation). In the history of the MATSIM project, this paper reports on the integration of Charypar and Nagel's approach to strategy optimization into the framework (Charypar and Nagel, 2005).

The paper is structured as follows. Our concept of an agent-based microsimulation of traffic systems is presented in section 2. Details on the new module *planomat* are given in section 3. Section 4 describes input data, assumptions about activity parameters as well as algorithm details. Results concerning choice of activity timing and system performance are presented in section 5. Finally, an outlook is given in the last section.

2. Micro simulation framework

In this section, the concepts required for understanding the *planomat* functionality are described briefly. For a comprehensive and more detailed framework description, see Raney and Nagel (2005).

2.1 The activity plan concept

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 (W3C, 2006). The most important XML elements are the following.

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

plan Each plan can be assigned a `score` according to a scoring function (see section 2.2). 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`. In the example shown, first and last activity are the same activity (`"h"`, which means home). The location coordinates refer to "Swiss Grid", the Swiss geodetic reference system (Swisstopo, 2006).

trips The attributes of a trip `<leg>` include a mode, a departure time and a duration. A trip can be characterized by a route, which is a sequence of numbers of the network nodes that are passed.

Read the example plan as follows:

- Agent No. 22018 is at home until 7:35:04. His home location *"h"* is at the coordinates (703600;236900).

Figure 1: Example activity plan

```
<person id="22018">
  <plan score="157.72" 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="w" x100="702500" y100="236400" link="5749" dur="08:12:05" />
    <leg num="1" mode="car" dep_time="16:03:40" trav_time="01:10:22">
      <route>1899 1848 1925 1924 1923 1922 1068</route>
    </leg>
    <act type="l" x100="681450" y100="246550" link="2140" dur="01:20:00" />
    <leg num="2" mode="car" dep_time="" trav_time="00:34:35">
      <route>1067 1136 1137 1921 1922 1923 1924 1925 1848 1899</route>
    </leg>
    <act type="h" x100="703600" y100="236900" link="5757" />
  </plan>
</person>
```

- He leaves his home to drive to work (*"w"*). This trip takes 16 minutes and 31 seconds, using the route along the nodes `1900 1899 1897`.
- The agent stays at work more than 8 hours, then leaves for a leisure activity (*"l"*). The trip from the work location on route `1899 1848 1925 1924 1923 1922 1068` to the leisure location takes about 1 hour and 10 minutes.
- After leisure, the agent returns home after a trip of ≈34 minutes.
- Read the plan as a 24-hour wrap-around, so the end of the home activity is also at 7:35:04 the next day.
- The plan has a score of 157.72€.

An activity plan can be interpreted in different ways: It can be either a *strategy* expressing what the agents wants/plans to do, or a *demand description* what an agent actually did in a certain iteration. The character of a plan is even more general: Since many attributes are not required, it is essentially a *working file* in the demand generation process[1].

[1]see the DTD at `http://www.vsp.tu-berlin.de/projects/Matsim/data/dtd/plans_v4.dtd`

2.2 Scoring

The quality of an activity plan is measured by a score. The corresponding scoring function was introduced first by Charypar and Nagel (2005), and is with slight modifications also used in our current work on traffic micro simulation. This subsection presents the basic parts of the utility function, while subsection 2.3 demonstrates its use in the micro simulation framework. Since here is given a compressed description, the interested reader is referred to the original paper by Charypar and Nagel.

The score of an activity plan U_{plan} is given by the sum of the utilities of all performed activities i, and the travel disutilities for trips necessary to get from one activity location to the other:

$$U_{plan} = \sum_{i=1}^{n} U_{act}(type_i, start_i, dur_i) + \sum_{i=2}^{n} U_{trav}(loc_{i-1}, loc_i)$$

The utility of an activity i is the sum of four terms, each of which is modeling a certain aspect of the utility function.

$$U_{act,i} = U_{dur,i} + U_{wait,i} + U_{late.ar,i} + U_{early.dp,i} + U_{short.dur,i}$$

$U_{dur,i}$ denotes the utility of executing an activity for a certain duration, $U_{wait,i}$ denotes the (dis)utility of waiting for an activity to start (for instance waiting for a shop to open), $U_{late.ar,i}$ and $U_{early.dp,i}$ denote penalties for coming too late or leaving too early that activity respectively, and $U_{short.dur,i}$ is a penalty if an activity is performed for too a short time.

U_{trav} denotes the (dis)utility of traveling from the location of activity $i-1$ to the location of the current activity i.

There is no penalty for *not* performing an activity that might have been planned. Only performed activities contribute to the plan score.

Utility of performing an activity

All terms in the activity utility function except U_{dur} are modeled to be linear in time needed for that activity aspect. The time performing an activity is assumed to have a logarithmic impact on activity utility to reflect diminishing marginal utility:

$$U_{dur} = \begin{cases} \beta_{dur} \cdot t^* \cdot \ln(\frac{t_{dur}}{t_0}) & (t_0 \leq t_{dur}) \\ 0 & (0 \leq t_{dur} < t_0) \\ \beta_{neg.dur} \cdot |t_{dur}| & (t_{dur} < 0) \end{cases} \text{, with}$$

$$t_0 = t^* \cdot \exp^{-10/p \cdot t^*}.$$

t_{dur} denotes the actual activity duration. t^* is the so called *operating point* of the activity, the duration at which the marginal utility equals β_{dur}. So, the value of t^* can be interpreted as the typical duration of an activity, while its effect in the activity plan context is the following: The t_i^* yield the ratios of the durations of different activities in equilibrium.

t_0 is the activity duration at which the logarithmic curve has its null. It is chosen proportional to the operating point, and is influenced by the priority p of the activity. Usual values for p are 1,2,3..., with 1 being the highest priority. The higher the priority, the smaller will be t_0. In busy plans, high-priority activities tend to stay in the plan while low-priority activities will be dropped when for instance traffic conditions worsen. In the current state of our work on activity generation, we use fixed, revealed activity chains, and activity dropping is not allowed. All activities have the same priority $p = 1$. This is why this issue is not described in more detail here.

The utility of performing an activity with a positive duration cannot be negative. Due to the interpretation of an activity plan as 24 hour-wrap round, in the first iterations of the micro simulation framework negative durations can occur. They are penalized linearly with $\beta_{neg.dur}$. This reflects a very undesired plan where it took the agent more than 24 hours to fulfil its plan.

Penalties

The penalty terms of the utility function are penalized linearly according to Vickrey's model of departure time choice (e.g. Arnott *et al.*, 1993):

$$U_{trav}(t_{trav}) = \beta_{trav} \cdot t_{trav},$$

$$U_{wait}(t_{wait}) = \beta_{wait} \cdot t_{wait},$$

$$U_{late.ar}(t_{start}, t_{latest.ar}) = \begin{cases} \beta_{late.ar} \cdot (t_{start} - t_{latest.ar}) & (t_{start} > t_{latest.ar}) \\ 0 & (t_{start} \leq t_{latest.ar}) \end{cases}$$

(where t_{start} is the starting time of the activity and $t_{latest.ar}$ the latest possible starting time of that activity),

$$U_{early.dp}(t_{end}, t_{earliest.dp}) = \begin{cases} \beta_{early.dp} \cdot (t_{earliest.dp} - t_{end}) & (t_{end} < t_{earliest.dp}) \\ 0 & (t_{end} \geq t_{earliest.dp}) \end{cases}$$

(where t_{end} is the ending time of the activity and $t_{early.dp}$ the earliest possible ending time of that activity), and

$$U_{short.dur}(t_{start}, t_{end}) = \begin{cases} \beta_{short.dur} \cdot (t_{shortest.dur} - (t_{end} - t_{start})) & (t_{end} < t_{start}) \\ 0 & (t_{end} \geq t_{start}) \end{cases}$$

(where $t_{shortest.dur}$ is the shortest desired duration for that activity).

Summary of parameters

The parameters of the utility function have the following values:

$\beta_{dur} = 6$€/h,

$\beta_{trav} = -6$€/h,

$\beta_{wait} = 0$€/h,

$\beta_{late.ar} = -18$€/h,

$\beta_{early.dp} = 0$€/h,

$\beta_{short.dur} = 0$€/h,

$\beta_{neg.dur} = -18$€/h.

The parameters for the penalty terms are chosen to reflect the relations in Vickrey's model of departure time choice:

$\beta_{wait} : \beta_{trav} : \beta_{late.ar} = 1 : 2 : 3$

This relation is not obvious on first sight when looking at the parameter values:

$\beta_{wait} : \beta_{trav} : \beta_{late.ar} = 0 : -6 : -18$

Considering the opportunity costs of *not* performing an activity while waiting or traveling, one has to subtract β_{dur} from β_{wait} and β_{trav}. So, the effective parameter values are the following:

$\beta_{wait,eff} : \beta_{trav,eff} : \beta_{late.ar,eff} = -6 : -12 : -18$,

which means the Vickrey type model is yielded. These values are different from the ones used in Charypar and Nagel (2005), who already discussed the issue of opportunity costs.

Figure 2 demonstrates the utility calculation using the example activity plan shown in Figure 1.

2.3 Simulation

The task of a simulation is to find the stationary state of the system modeled. In the case of our transport system model, the stationary state is the state where an agent cannot improve its score by altering the plan. So the objective function of the simulation system is to maximize the overall score:

$\max(\sum_{a=1}^{n} U_{plan,a})$, with n being the number of agents simulated.

As pointed out, an iterative approach is used to solve this maximization problem. where travel times as a representative for generalized travel costs are the central feedback element. The overall simulation system 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. For instance, travel times are directly proprotional to the physical distance without any network capacity effects. 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 (usually one that was modified before, otherwise random selection).
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 2.2). If the stop criterion of the simulation is met, go to step 6.

4. A subset of the agents is chosen for plan modification/new plan generation 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, further 10% for rerouting only.

5. One or more external strategy modules are run. For each agent one plan is returned. The new plan, considering the updated travel times, is stored in the agent database. Return to step 2.

6. End of the simulation.

The stop criterion mentioned is the amount of improvement in overall score after subsequent iterations. If it falls under a certain value ϵ, the stationary state is probably found.

3. Methods of planomat

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 are described.

3.1 Old time allocation mutator

The first two paragraphs of this section are taken from Raney and Nagel (2005, p. 16).

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 $r_1 \in [-30\,min,\ 30\,min]$. 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 Figure 4). This is because of the insufficient exploration within $\pm 30\,min$, 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. This is far too much for any practical use, since one iteration takes about 40 minutes on a well equipped Single-CPU system. Simple extension of the search range, e.g. $\pm 6\,h$ 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 *time allocation mutator*.

3.2 Implementation details

The idea is now to search new solutions in the entire search space and find an optimum in it using a scoring function. Here, the same utility function as in the agent database is used (see section 2.2).

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.*, 2005b; Schneider, 2003). This paper is about 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.

Input data and alternative creation

For each agent, the selected plan and the recently experienced travel time information are read in. While former comes from the agent database, the latter comes from the result of the previous run of the traffic flow microsimulation. This information is structured in so-called *events*, small data packages containing what agent did what at which time in which place. From departure and arrival events, the travel times can easily be computed. In previous versions of planomat, different (worse) sources for travel time information were used (compare Meister *et al.*, 2005a).

The start time of the plan, that is the end time of the first activity, is uniformly selected between 00:00 and 24:00. The same is done for each activity duration. All other attributes are kept as they came from input (as described, in the current state of the work planomat only optimizes time allocation).

Recombination and mutation

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 alters each time information in a certain range parameterized with the *mutation probability* p_{mut}:

- A new start time is chosen by adding an amount s uniformly selected from range $s \in [p_{mut} \cdot -12h,\ p_{mut} \cdot 12h]$. Values that come before 00:00 (midnight) are reset to that time.
- An activity duration is multiplied with a factor $d = e^X$ with X being uniformly selected from the range $X \in [-p_{mut}/2,\ p_{mut}/2]$.

After both the creation and the recombination/mutation operations, the new plan is stretched/compressed to a duration of 24 hours to be comparable to its competitors in the GA population.

Selection and output

Every time a new activity plan was created by the GA, it is evaluated with the scoring function. Since the number of plans held in the GA population at one time is constant, good plans are kept while bad ones are dropped. After a fixed number of recombination/mutation operations, the optimization is canceled. The best plan currently in the population is chosen as a new strategy for the given agent, and so given back to the agent database.

GA parameters

Table 1 gives a brief overview of the various GA parameters that have to be configured. All these parameters have to be chosen according to the nature of the problem to be solved. This is often done on a gut level, so is in this case.

4. Canton Zurich Scenario

The scenario 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.

Table 1: GA parameters

Variable	Description	Value
$popsize$	Constant population size.	`50`
n_{gen}	Number of generations. Here, if n_{gen} individuals were generated by the crossover/mutation operations, the optimization is canceled.	`1’000`
p_{mut}	Probability that one element of an activity will mutate according to its respective mutation operator.	Initial: `0.30`, exponentially decreasing to `0.07`
τ_{mut}	Each time a new indivdual was inserted into the population, p_{mut} is adapted. The higher τ_{mut}, the quicker p_{mut} decreases.	
$mindiff$	Minimum fitness difference between two individuals. If a new plan with almost the same score is generated, it will be dropped in favor of the one that is already present.	`0.10`

4.1 Study area: Canton Zurich

The case study used for testing the *planomat* is a simulation of the Canton Zurich, the biggest metropolitan area in Switzerland. The demand generation process, as well as the framework used for it, is described in detail in Balmer *et al.* (2006).

First, a synthetic population of the Canton Zurich is generated, using data from the Swiss National Population Census. It is a list of ≈1’200’000 agents with individual attributes like age or sex, and a hectare-based home location (Frick and Axhausen, 2004). Each agent is assigned an activity chain based on the Swiss Microcensus on travel behavior (Rieser, 2004). These activities are distributed in space by several location choice modules (Marchal and Nagel, 2005). The network model used for the assignment with a microscopic traffic flow simulation is the Swiss National Traffic Network model (Vrtic *et al.*, 2002).

For test reasons, the traffic of only a 1% sample of the whole agent population is simulated. In order to produce comparable results to full scenario where 46% of alle agents are simulated, the network capacity was reduced to $< approx 2$%. So, some congestion

Table 2: Activity parameter values

Activity type	abbreviation	t^* [h]	$t_{shortest.dur}$ [h]	$t_{latest.ar}$	$t_{earliest.dp}$
home	`h`	12	8	—	—
work	`w`	8	6	9:00	—
work1	`w1`	4	2	9:00	—
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	—	—

All activities have the same priority $p = 1$.
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 an 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`.

occurs and sensitivity of timing decisions to experienced travel times can be observed.

4.2 Activity parameters and constraints

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

Each activity is characterized by a typical duration t^*, a mimimum duration $t_{shortest.dur}$ and desired start/end times $t_{latest.ar}$, $t_{earliest.dp}$. While the typical duration is a mandatory parameter to the utility function, the minimum duration and desired time windows are optional. Table 2 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, it doesn't gain any score or even loses

Table 3: Opening hours as temporal constraints

Activity type	opening time	closing time
home (`h`)	—	—
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`

some in case of $\beta_{wait} < 0$. The temporal constraints are an attribute of a specific facility. In this setup, they are the 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 3.

For analysis, the activity chain types are summarized into five groups:

education-dominated chain types `heeh`, `heh`

leisure-dominated chain types `hlh,hllh,hlslh`

shop-dominated chain types `hsh,hssh`

work-dominated chain types `hwh,hwlwh,hwswh,hwwh`

other chain types `helh,hesh,hleh,hlsh,hlwh,hslh,hswh,hweh,hwlh,hwsh`

5. Results

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.

Figure 3 shows the development of the average score across the whole agent population. Both curves show tendency towards a limiting value. For the *planomat* setup, the slope is no more visible after 100 iterations, converging at an average score value of ≈160 €. The *time allocation mutator* configuration has a rising curve still visible after >1000 generations (not shown). It is not clear if it converges to the same value. This improvement is in the range of one-two orders of magnitude regarding the required number of iterations.

Figure 4 shows the departure time distributions at the beginning of the simulation and after 400 generations, for each setup. The main difference is in the distribution of the leisure-dominated activity chain types. While in the *time allocation mutator* setup, most leisure activities take place in the morning, the *planomat* distributes them all over the day. The latter is the expected result, since *leisure*-type activities are only constrained within 6:00 and 24:00. One would expect a bigger evening peak for leisure activities. It is not that pronounced because opening time constraints are probably too lax, e.g. cinemas and bars usually don't open at 6:00 AM but in the afternoon, and also close later. A similar effect can be observed for the shopping-dominated activity chain types, where the arrival/departure times at/from the shops distribute in the opening hour window 8:00-20:00. Furthermore, the afternoon commuter peak is more pronounced (regard work-dominated activity chain types).

The suboptimal time allocation distribution in the *time allocation mutator* setup doesn't change after >1000 iterations (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 $\pm 30\ min$ per iteration. 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. But as the stationary state found should be independent of the initial conditions, this demonstrates the necessary step taken with the introduction of *planomat*.

6. Discussion and outlook

6.1 System requirements

All following figures apply to a Pentium IV Xeon system, 2.4 GHz, 4 GB RAM, SuSE Linux 9.2, gcc version 3.3.5. The entire simulation system presented here was run using only one CPU.

The most critical requirement of the current *planomat* implementation is computing time, scaling linearly with the number of agents to be replanned. The replanning performance is at $\approx$100 agents/s. For the 1% Canton Zurich scenario described, this means a runtime of $\approx$13s. In the full scenario, where a fixed car-mode-share of 46% of all agents is assumed, replanning takes almost 10 minutes. These stand-alone figures are difficult to evaluate, since the goal is to minimize overall runtime. It is influenced by the runtime of the external strategy modules, the simulation of traffic flow, and the number of iterations required for a satisfying level of convergence.

The traffic flow simulation used mainly scales with network size, only little with the number of agents (Cetin, 2005). In our single-CPU setup, it takes 8-15 minutes to simulate the whole day (including I/O). So, while in the 1% case the computing time required for planomat can be neglected, it is of considerable size when all car-driving agents are simulated. The minimization of computing time as well as the number of iterations required is a main focus of our further work. Charypar *et al.* (2006) perform experiments with Evolutionary Strategies as optimization method to reduce computing time.

Memory requirements are no limiting factor to performance, since optimization is done agent by agent. Plans information as well as simulation events information are streamed, which means I/O data takes virtually no memory at all. Most memory is required by the GA population of max. 50 activity plans which require $<<$1MB of RAM.

6.2 Improvement of the location choice concept

One upcoming modeling goal is the improvement of the location choice concept. The basic difference will be that location choice for secondary activities will be part of the replanning process, instead of its currently limited role as a preprocess to initial demand generation (Marchal and Nagel, 2005).

At first, we will improve the data basis. Up to now, the number of overall workplaces in a spatial aggregate was assumed as predictor for the utility gained there, regardless of the activity type. This is insufficient because the functional organization typical for urban areas is not considered at all. We create an activity-fine set of facilities based on landuse information available on hectare-level for all Switzerland, called the Swiss National Enterprise Census provided by the Swiss Federal Statistical Office (BfS, 2001). Opening time windows will be no more activity-specific, but location-specific. Data about opening times still have to be imputed/revealed. Furthermore, the synthetic facilities will have an activity-specific capacity which in the first run will be proportional to the number of workplaces. An open question is how to include location capacity constraints into the agents' decision making.

For each agent, a choice set of locations is generated. Here, an approach based on revealed activity spaces is chosen. Refer to *activity space* as a continuous spatial representation of the locations visited by a person in a certain time range. We will use activity space generation algorithms developed in Vaze *et al.* (2005). It is then task of the *planomat* to find the best location for each activity in the sense of the scoring function. The complexity of the search space is thus extended with a non-scalar dimension *activity location*. Earlier GA experiments show that this task is feasible, although it will take more time than the comparably simple time allocation problem (Charypar and Nagel, 2005; Meister *et al.*, 2005b).

References

Arnott, R., A. de Palma and R. Lindsey (1993) A Structural Model of Peak-Period Congestion: A Traffic Bottleneck with Elastic Demand, *The American Economic Review*, **83** (1) 161–179.

Balmer, M. (2005) Matsim utility function, Internal presentation, Institute for Transportation Planning and Systems (IVT), ETH Zurich, Nov. 2005.

Balmer, M., K. W. Axhausen and K. Nagel (2006) A demand generation framework for large scale micro-simulations, in TRB (2006).

BfS (2001) *Eidgenössische Betriebszählung 2001 - Sektoren 2 und 3*, Swiss Federal Statistical Office (BfS), Neuchâtel, URL `http://www.bfs.admin.ch/bfs/portal/de/index/dienstleistungen/servicestelle_geostat/datenbeschreibung/eidgenoessische_betriebszaehlung2.html`.

Cetin, N. (2005) Large-scale parallel graph-based simulations, Ph.D. thesis, ETH Zurich, Zurich.

Charypar, D., K. W. Axhausen and K. Nagel (2006) Implementing activity-based models: Accelerating the replanning process of agents using an evolution strategy. Submitted to 6^{th} Swiss Transportation Research Conference.

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

Frick, M. and K. W. Axhausen (2004) Generating synthetic populations using IPF and Monte Carlo techniques, in STRC (2004), URL `http://www.strc.ch/2004.html`.

Marchal, F. and K. Nagel (2005) Modeling location choice of secondary activities with a social network of cooperative agents, in TRB (2005).

Meister, K., M. Balmer and K. W. Axhausen (2005a) An improved replanning module for agent-based micro simulations of travel behavior, *Working Paper*, **303**, Institute for Transportation Planning and Systems (IVT), ETH Zurich, Zurich, URL `http://www.ivt.ethz.ch/vpl/publications/reports/ab303.pdf`.

Meister, K., M. Frick and K. W. Axhausen (2005b) 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).

Rieser, M. (2004) Generating Day Plans from Origin-Destination Matrices, Semester project, Institute for Transportation Planning and Systems (IVT), ETH Zurich.

Schneider, A. (2003) Genetische Algorithmen zur Optimierung von Tagesplänen für Verkehrsteilnehmer. Term project, Computer Science, ETH Zürich.

STRC (Ed.) (2004) *STRC 2004 Conference Proceedings*, Ascona, Mar. 2004, URL `http://www.strc.ch/2004.html`.

Swisstopo (2006) *Swiss map projections*, URL `http://www.swisstopo.ch/en/basics/geo/system/projectionCH`.

TRB (Ed.) (2005) *TRB 2005 Annual Meeting CD-ROM*, Transportation Research Board, Washington, D.C., Jan. 2005.

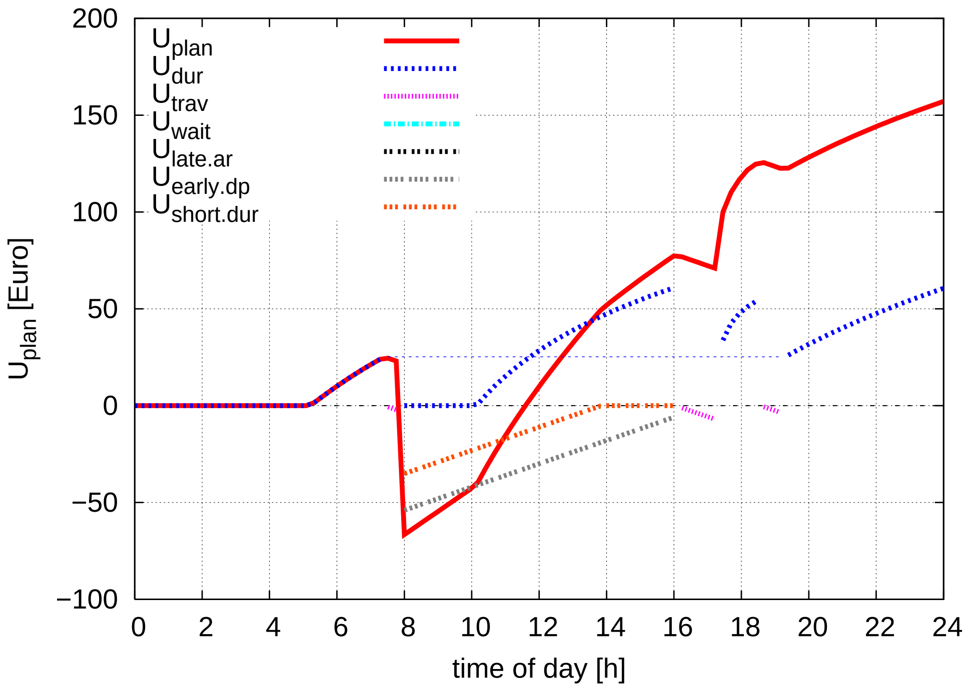
TRB (Ed.) (2006) *TRB 2006 Annual Meeting CD-ROM*, Transportation Research Board, Washington, D.C., Jan. 2006.

Vaze, V. S., S. Schönfelder and K. W. Axhausen (2005) Continuous space representations of human activity spaces, *Working Paper*, **295**, Institute for Transportation Planning and Systems (IVT), ETH Zurich.

Vrtic, M., P. Fröhlich and K. W. Axhausen (2002) Schweizerische Netzmodelle für Strassen- und Schienenverkehr, in T. Bieger, C. Laesser and R. Maggi (Eds.), *Jahrbuch 2002/2003 Schweizerische Verkehrswirtschaft*, 119–140, SVWG, St. Gallen, Schweiz.

W3C (2006) *Extensible Markup Language (XML)*, World Wide Web Consortium (W3C), URL `http://www.w3.org/XML/`.

Figure 2: Utility plot of example activity plan



The graph U_{plan} represents the plan score depending on time of day as this plan was canceled at that certain time of day. One clearly sees positive utility of activity performance (log-shape graphs), the various penalties (linear graphs starting on the x-zero axis) as well as the overall plan score yielded at 24:00.
The very low score value between 8:00 and 10:00 can be explained as follows: On one hand, only the home activity and a small part of the work activity including the (penalized) home-work trip were performed. On the other hand, the penalties for early departure $U_{early.dp}$ and short activity performance $U_{short.dur}$ are very high.
The activity parameters used here are listed in Table 2, which is part of the scenario description in section 4.
For explanatory reasons, in this figure $\beta_{early.dp} = \beta_{short.dur} = -6$€/h, instead of 0€/h.
Based on (Balmer, 2005, p.15 ff.).

Figure 3: Comparison of average scores

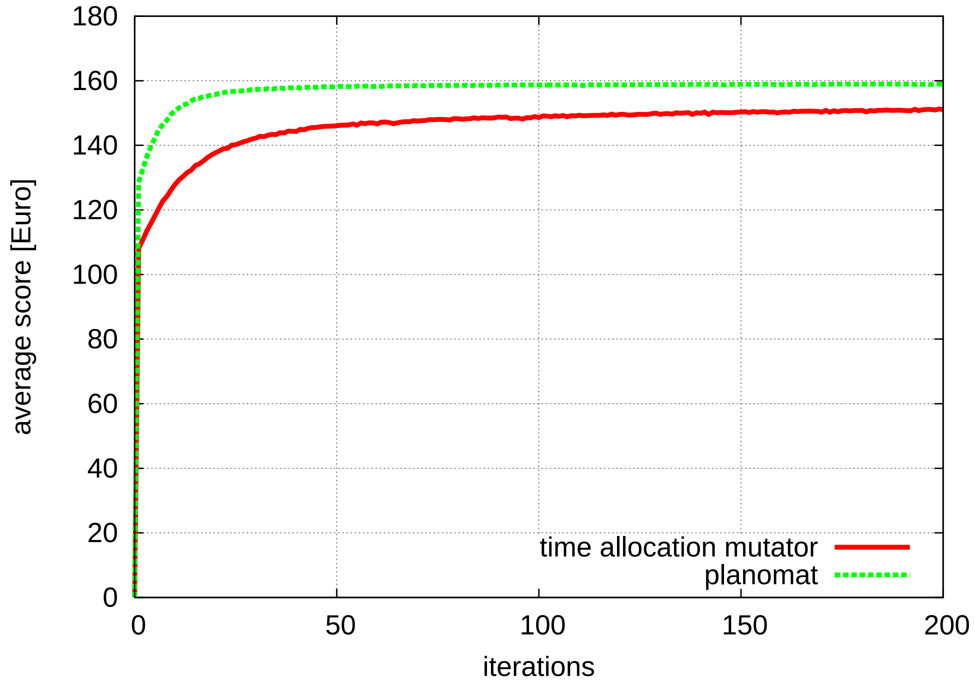
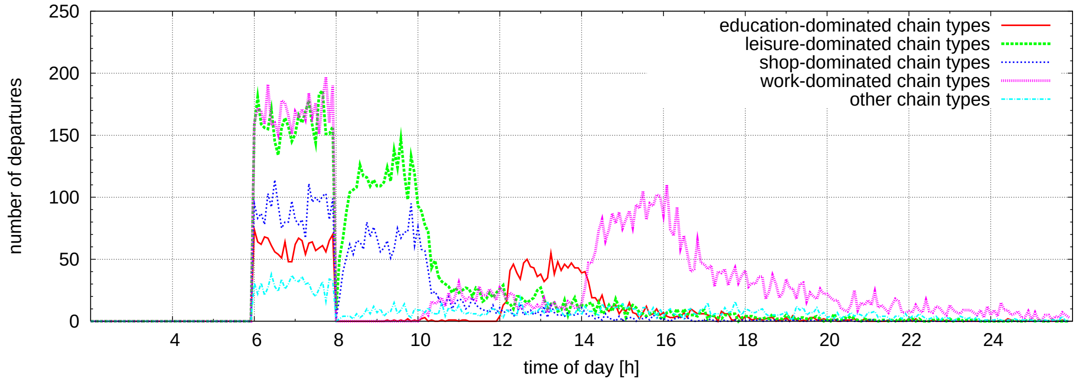
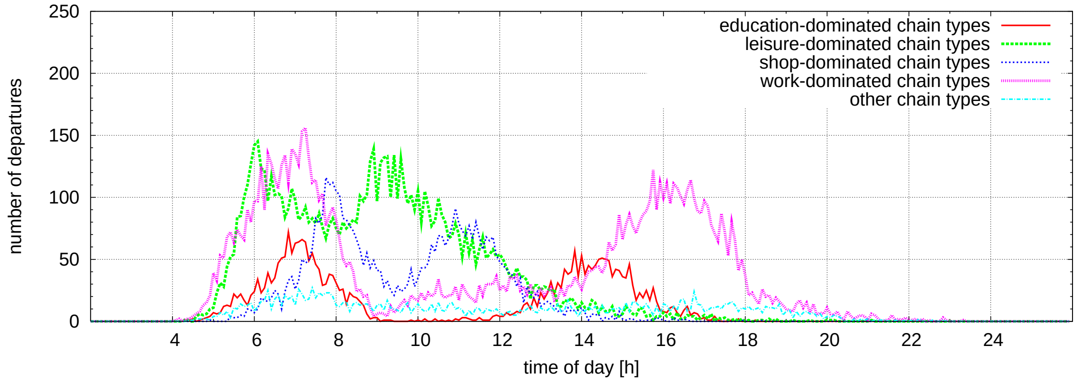


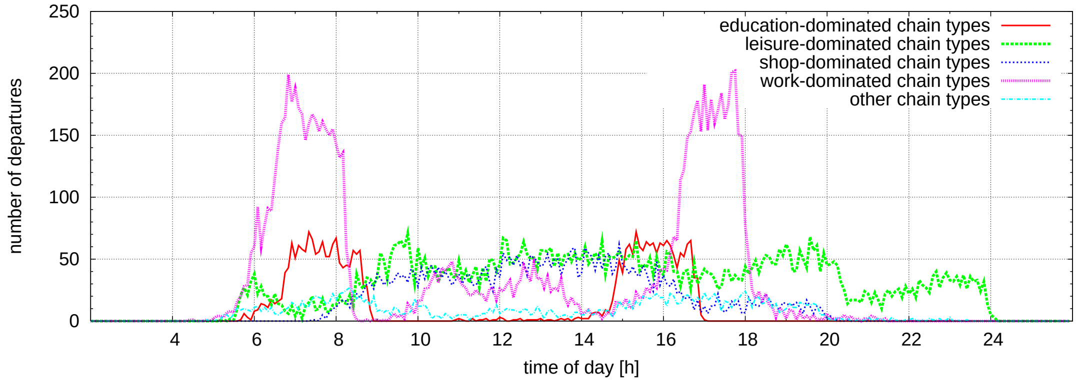
Figure 4: Comparison of departure times by activity chain type



(a) Departure times in initial plans (iteration 0)



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



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