

---

# **Activity Planning in a Continuous Long-Term Travel Demand Microsimulation**

**Maxim Janzen**

**Kay W. Axhausen**

**Institute for transport planning and systems**

**April 2015**

**STRC**

**15th Swiss Transport Research Conference**

Monte Verità / Ascona, April 15 – 17, 2015

Institute for transport planning and systems

## Activity Planning in a Continuous Long-Term Travel Demand Microsimulation

Maxim Janzen  
IVT  
ETH Zürich  
CH-8093 Zürich  
phone: +41-44-633 33 40  
fax: +41-44-633 10 57  
maxim.janzen@ivt.baug.ethz.ch

Kay W. Axhausen  
IVT  
ETH Zürich  
CH-8093 Zürich  
phone: +41-44-633 39 43  
fax: +41-44-633 10 57  
axhausen@ivt.baug.ethz.ch

April 2015

### Abstract

To date, travel demand generation for microscopic traffic flow simulations and travel demand models focuses on reproducing daily life behavior. This stands in contrast to the significant part of traffic volume caused by journeys related to activities not usually undertaken during daily life.

In the case of long distance travel demand it is necessary to simulate a long period of travel behavior, because long distance trips are usually rare and often last longer than one day. We use and adapt the **Continuous Target-based Activity Planning (C-TAP)** simulation in order to simulate long distance travel demand for a long time period. The adapted model, **Long-Term C-TAP**, was used for a simulation of one year. In comparison to other activity driven microsimulations the activities taken account in the **Long-Term C-TAP** simulation are highly abstracted. Besides the main activity called daily life (including home stays, work, daily shopping etc.) we simulate all stays (e.g. holidays) taking place more than 100km away from the place of residence.

One of the characteristics of long-term and long-distance travel behavior is the enlarged planning horizon, i.e. people tend to plan activities in advance. This fact has to be captured in an agent-based simulation, like **Long-Term C-TAP**. We will introduce a decision making algorithm based on the minimization of the discomfort, which values each possible series of activities. Each series is defined by activity types for each activity, time periods for each activity and (fixed) travel times between the activities. In order to allow an arbitrary planning horizon the main objective of this algorithm is scalability.

### Keywords

continuous target based model; long term simulation; long distance travel demand; microscopic travel demand simulation; C-TAP

# 1 Introduction

Microscopic travel demand simulations simulate the (traveling) behavior of virtual agents individually. One of the well known approaches is the one proposed by Balmer (2007): agents choose a daily schedule for their behavior and execute it. The execution results are reported and the agents can re-plan their schedule based on the results of all agents. This procedure is iterated until a stochastic user equilibrium with consistent travel demand is reached (Nagel and Flötteröd, 2009). Due to high computational complexity and memory issues (all current schedules have to be maintained) a reasonable simulated period is a single day. This is not sufficient for the task of long distance travel demand generation, because long distance traffic is a significant part of today's traffic and short term simulations are not capturing this part.

In case of long distance travel demand it is necessary to simulate a long period of travel behavior, because long distance trips are usually rare and take more time than short distance trips. We use and adapt the **C**ontinuous **T**arget-based **A**ctivity **P**lanning (C-TAP) simulation which is proposed by Märki (2014). It was shown that this approach is able to reproduce individual behavior of six weeks (Märki et al., 2012c). We modify this model in order to simulate long distance trips over an even longer period. The goal of the modified simulation is the generation of travel demand data for a whole year.

The Long-Term C-TAP simulation is one of the first microsimulations, which provides an estimation of long-distance travel demand. An estimator for long-distance travel demand is valuable, because it introduces a new possibility to justify political decisions in this policy domain. An application might be the evaluation of big infrastructural investments, like new bridges or tunnels, which is very useful for the cost-benefit analysis of these investments.

One major task within the development of the described simulation is the activity planning. In case of the Long-Term C-TAP simulation the planning is dynamic, i.e. no complete schedules are created, but for each agent a decision about his next activity has to be made each time he finishes his current activity. Nevertheless, the agents have to plan more than one activity at each decision point as real life people also tend to plan long-distance trips in advance. The implementation of the decision process is the focus of this paper, in particular the scalability in the size of the planning horizon has to be ensured.

The remainder of this paper is structured as follows. First, we present the Long-Term C-TAP model including its main functionality. After that, we focus on the activity planning model and an analytic description of the arising problem. In the next step an algorithm for a numerical solution of the problem is presented. We conclude the paper with a consideration on limitations of the current simulation and an outlook on potential future development.

## 2 Related Work

Agent based simulations have a long tradition in analysis and explanation of social behavior. Schelling (1971) is often referred to be the first developer of an agent based simulation. Microsimulations were also used to estimate travel demand ((Pendyala et al., 1997)) or to generate an activity-based travel forecast (e.g. Bhat et al. (2003) or Miller (1996)). Nowadays agent based simulations make a notable contribution to the field of transportation research (e.g. Balmer (2007)).

The target-based approach is related to the need based theory which was introduced by Arentze and Timmermans (2006). They developed also a model for activity generation with the assumption of utilities described as dynamic function of needs (Arentze and Timmermans, 2009). Märki proposes to use targets instead of needs as an explanation of human behavior (Märki et al., 2012b). He validated his model in Märki et al. (2012c) for short distance travel generation using a six-week continuous travel diary provided by Löchl et al. (2005).

Long distance trips have been also the focus of recent literature. The travel behavior have been analyzed several times, e.g. for the UK and the Netherlands Limtanakool et al. (2006). Some statistical long distance travel demand models have been developed (e.g. Erhardt et al. (2007)) as well as used for traffic forecast (e.g. Beser and Algers (2001)). Recently, different surveys were also analyzed to derive an outlook on the future of long distance travel demand (Frick and Grimm (2014)).

Finally, the usage of a continuous target-based model for a long term simulation was introduced recently by Janzen et al. (2014) and Janzen and Axhausen (2015). This adaptation is beneficial, because the statistical models of long-distance travel behavior focus on the current state of the world, which is not always sufficient. Thus, there is a need of a tool to predict travel demand after major infrastructural changes.

## 3 Continuous Target-Based Model

We introduce a microscopic travel demand model, which is used to generate long-term and long-distance travel demand. The core of microscopic models is built with agents representing virtual people. In contrast to iteration-based models (like the one used by Balmer (2007)) a continuous planning model does not iterate to a steady state, but generates continuously an activity schedule without a systematic replanning. One of the main advantages is the capability

of the simulation to generate arbitrarily long activity plans in linear runtime. Thus, it is a better basis for the generation of long term, long distance travel demand. Finally, we choose an event-driven simulation, which is more effective for our issues than a time-driven simulation, because the action between two events is not crucial to the simulation and maintaining a single event queue can be implemented simply in our simulation.

The simulation presented in this section was introduced by Märki et al. (2012b) and further developed by Märki et al. (2012a) and Märki et al. (2013). Finally, the extension to a long-term simulation was presented by Janzen and Axhausen (2015). We explain in this section the main ideas of Long-Term C-TAP, i.e. the behavioral targets, the activities and their interaction within the simulation algorithm.

### 3.1 Behavioral Targets

The core idea of (Long-Term) C-TAP is the usage of behavioral targets, which represent the motivation of the agents to perform an activity. Examples of long-distance and long-term motivations are holidays. In this case an agent might have the motivation to go on holidays for two weeks twice a year.

There are several options to define targets. In the following we present the types of targets proposed by Märki (2014):

- percentage-of-time target: indicates how much relative time within an observation window an agent would like to spend on a specific activity (e.g. the motivation to spend a specific amount of time on holidays within one year).
- frequency target: indicates how often an agent would like to execute a specific activity within an observation window (e.g. the motivation to go a specific number of times on holidays within a year).
- duration target: indicates how much time an agent would like to spend for a single execution of a specific activity (e.g. the motivation to spend a specific amount of time on each holiday trip).

Note that the first two target types include the definition of an observation window. But in case of the simulation presented here it is not necessary to include additional parameters to the calibration of the simulation, because it is sufficient if the observation window just equals the simulated time, i.e. one year. Nevertheless, the user can give some of the targets a bigger weight, if he decreases the observation window.

Obviously, the definition of all three targets for a single activity has redundancy as just any two of the three target types are necessary to fully describe the motivation of the agents for a single activity. In order to reduce runtime and complexity of the model we use just two of the three proposed targets. Experiments have shown that the best performance is achieved with the limitation to duration and percentage-of-time targets.

## 3.2 Activities and State Values

Activities are necessary to complete the concept of a target-based simulation, because the targets/motivations described above are satisfied by the execution of a corresponding activity. Activities also mark the trip purposes. The decision on the executed activities is based on state values. For each target we define a state value, which is necessary to measure the satisfaction. We need to introduce two types of state values:

- for targets with observation windows (percentage-of-time and frequency targets): the state value is the result of a convolution of the activity execution pattern with an exponential kernel, which is restricted to the length of the observation window. So it increases during the execution of the relevant activity, respectively decreases during non-execution.
- for duration targets: the state value is defined as the activity duration.

The level of satisfaction now is measured by the quadratic difference of state value and target value. This measurement is called *discomfort* and its influence within the model is described in detail in section 4.1.

## 3.3 Core Algorithm

We will now shortly present and discuss the main implementation issues of the Long-Term C-TAP model. The core algorithm of the Long-Term C-TAP simulation has a simple structure and is shown in algorithm 1.

The main procedure is a continuous, event-driven iteration over discrete points of time. This iterative process is implemented by the outer while-loop including the incremental computation of the consecutive time points in lines 7 and 8. Whenever an agent finishes the execution of an activity, the function *MakeDecision* (line 4) computes the next activity based on its current state, which has to be updated before (line 3). After that, the activity is executed until the computed execution end. Activity execution also includes traveling to the location of the activity. Recording these trips we obtain the travel demand. The simulation stops after a predefined

---

**Algorithm 1** Core C-TAP Algorithm (Pseudo Code)

---

```
1: while simulation end not reached do
2:   for all agent with no activity do
3:     state  $\leftarrow$  UpdateAgentState(agent)
4:     nextActivity  $\leftarrow$  MakeDecision(agent, state)
5:     agent.execute(nextActivity)
6:   end for
7:   nextTimeStep = minimum( all execution endpoints )
8:   proceed to nextTimeStep
9: end while
```

---

stopping condition is reached. This condition is usually a time period, which has to be simulated. In case of long term simulations a time period of one year is reasonable.

Considering this algorithm there is one important task remaining, namely the implementation of the *MakeDecision* function, which describes the activity planning. This challenge is the main topic of section 4.1. Our object is the generation of long distance travel demand. So we are not interested in every short trip, but just in those trips with long distances. We propose to use aggregated activities, e.g. we introduce a single activity representing daily life, which includes all short daily journeys like traveling to work, shopping, etc. In comparison to Märki (2014) this is a higher abstraction level of activities.

## 4 Activity Planning

As mentioned before the main challenge of the continuous agent-based simulations is the modeling of activity decisions. This is the focus of the following section. We will first describe the decision model, continue with an analysis of the arising optimization problem and finish the section with the solution implemented in the simulation.

### 4.1 Long-Term C-TAP Decision Model

In order to make decisions on the next performed activities one needs a measurement to value different options of activity performing. This valuation is the core of the decision process and should be simple and fast to compute. Given this we can compare different activities (and also different activity durations) and choose the best option. In the case of Long-Term C-TAP the

quality of a potential decision is measured by the *discomfort value*

$$D(t) = \sum_{k=1}^n (f_{target}^k(t) - f_{state}^k(t))^2 * w_k, \quad (1)$$

where  $n$  is the number of targets and  $w_k$  a bandwidth normalization factor. The function  $f_{target}^k(t)$  describes the target value at a given point of time  $t$ , while  $f_{state}^k(t)$  describes the state value at  $t$ . For simplicity and without loss of generality we assume that  $w$  equals 1.

The decision procedure is the following. Whenever a decision about the next activity of an agent has to be made, all possible combinations of next activities are computed. The number of planned activities is called planning horizon and is a parameter of the simulation (value is between 2 and 6). The next step is the calculation of the activity durations minimizing the discomfort value at the end of the planning horizon. Finally, the first activity of the optimal activity combination is chosen to be the next executed activity. The crucial part of this procedure is the minimization of the discomfort value, which is an exponential multi-dimensional optimization problem. We will describe this arising mathematical problem in detail and afterwards propose an algorithm to optimize the problem in reasonable runtime.

## 4.2 Analytic Description of the Discomfort Minimization Problem

The discomfort minimization problem for a specific agent and a fixed planning horizon is discussed in detail in this subsection. Following the proposition of the last section we assume that for every activity  $act_k$  ( $1 \leq k \leq n$ ) there exist two targets, namely a duration target  $T_{dur}^k \in \mathbb{R}$  and a percentage-of-time target  $T_{perc}^k \in [0, 1]$ . For simplicity we keep both types of  $T$  fixed, but an extension to a function of time would not require big changes in the following.

Given a planning horizon of  $m$  activities and a vector  $t = (t_1, \dots, t_m)$  of the respective activity durations, the discomfort of a single agent for a fixed series of activities  $act_{s_1}, \dots, act_{s_m}$  can be expressed as

$$D(t) = \underbrace{\sum_{k=1}^n (T_{perc}^k - v_m^k)^2}_{\text{percentage-of-time targets}} + \underbrace{\sum_{j=1}^m (T_{dur}^{s_j} - t_j)^2 * w_{s_j}}_{\text{duration targets}}, \quad (2)$$

where  $v_m^k$  is the state value of the percentage-of-time target corresponding to the activity  $act_k$  after execution of the last activity  $act_{s_m}$ . The first sum consists of the discomfort arising from percentage-of-time targets, while the second part sums the discomfort of the duration targets. Just the  $m$  performed activities generate duration discomfort, because all other activities do not



have any execution duration. Note that the series  $act_{s_1}, \dots, act_{s_m}$  can also contain an activity twice or more. In this case there would be multiple duration discomfort values for a single activity. The first does not include normalization factors, because its parts are already normalized to the  $[0, 1]$ -interval.

The remaining question in the discomfort calculation is the computation procedure of the  $v_m^k$ -values, i.e. how do the state values change during the execution of the activities. The state values are described by two exponential functions. First, there is a state value increasing function:

$$h_k(\hat{v}, \hat{t}) = 1 + (\hat{v} - 1)e^{-\tau_k \cdot \hat{t}}. \quad (3)$$

Second, we define also a state value decreasing function:

$$d_k(\hat{v}, \hat{t}) = \hat{v} \cdot e^{-\theta_k \cdot \hat{t}}. \quad (4)$$

In both cases  $\hat{v}$  is the state value before the increase or decrease applies.  $\tau_k$  and  $\theta_k$  are parameters, which are computed for every activity subject to multiple other variables. These values are not explained in detail here. Note also that valid values of  $v$  are between 0 and 1. Whenever an activity  $act_k$  is executed for a duration  $\hat{t}$ , the corresponding state value increases by  $h_k(\dots, \hat{t})$ , while any other state value of any activity  $act_o$  (with  $o \neq k$ ) decreases by  $d_k(\dots, \hat{t})$ .

Following this rules we can define the  $v$ -values for a fixed percentage-of-time target of  $act_k$  recursively as follows:

$$v_0^k = f_{state}^k(t_0) \quad (5)$$

$$v_i^k = \begin{cases} h_k(w_{i-1}^k, t_i) & \text{if } i > 0 \text{ and } s_i = k \\ d_k(w_{i-1}^k, t_i) & \text{if } i > 0 \text{ and } s_i \neq k \end{cases} \quad (6)$$

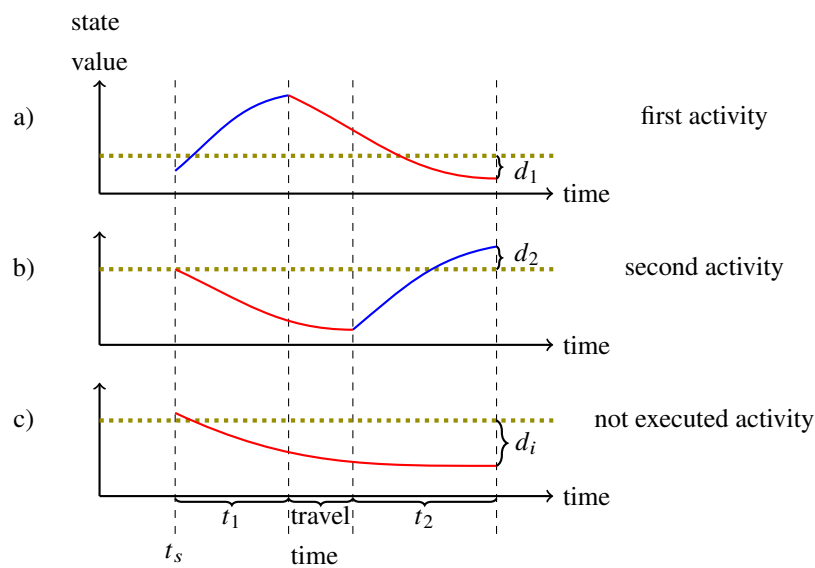
$$w_i^k = d_k(v_i^k, tr_i). \quad (7)$$

The current state value (at  $t_0$ ) is  $f_{state}^k(t_0)$  and the fixed travel time between an activity  $act_{s_i}$  and  $act_{s_{i+1}}$  equals  $tr_i$ . Thus,  $w_i^k$  describes the state value after traveling to the  $(i + 1)$ -th activity. During traveling all state values decrease, because in that time no activity target can be satisfied.

You can find an illustration of the state value function for a percentage-of-time target in figure 1. The state value corresponding to the activity, which is considered to be executed first, increases during the first execution period  $t_1$  and decreases during travel time and the second activity

execution period  $t_2$  (figure 1 a)). The state value of the second target/activity increases during  $t_2$  and decreases otherwise (figure 1 b)). Finally, all other state values decrease during the whole considered period (figure 1 c)). Though, the value we focus on is not the absolute state value but the discomfort which is the quadratic difference of the state value and the target. Possible (static) target values are plotted in the graph as green dotted lines. The sum of all quadratic differences at the end of the second activity execution is the overall discomfort we want to minimize and can be identified in the figure as the sum of  $d_1^2$ ,  $d_2^2$  and  $d_i^2$  for all not executed activities  $i$ .

Figure 1: Illustration of a state value function for a percentage-of-time target



Two important remarks regarding the implementation of any algorithm based on the discomfort computation have to be made. First, there is an important fact about the nested functions described above:  $h(h(v, t_1), t_2) = h(v, t_1 + t_2)$ . The same applies to the  $d$ -function. This property simplifies the computation of a single discomfort value. Second, there is one special activity in the C-TAP model, called daily-life activity. This activity does not have any duration target, but is just restricted by a lower bound.

After the full description of the discomfort evaluation function the question for a way to minimize this function remains. The discomfort is a recursively defined, multi-dimensional and non-linear function. This combination of attributes makes it very hard to find a clean analytic solution to the optimization problem, which can be implemented in the simulation. Thus, we will propose and describe a numerical solver to the problem in the next subsection.

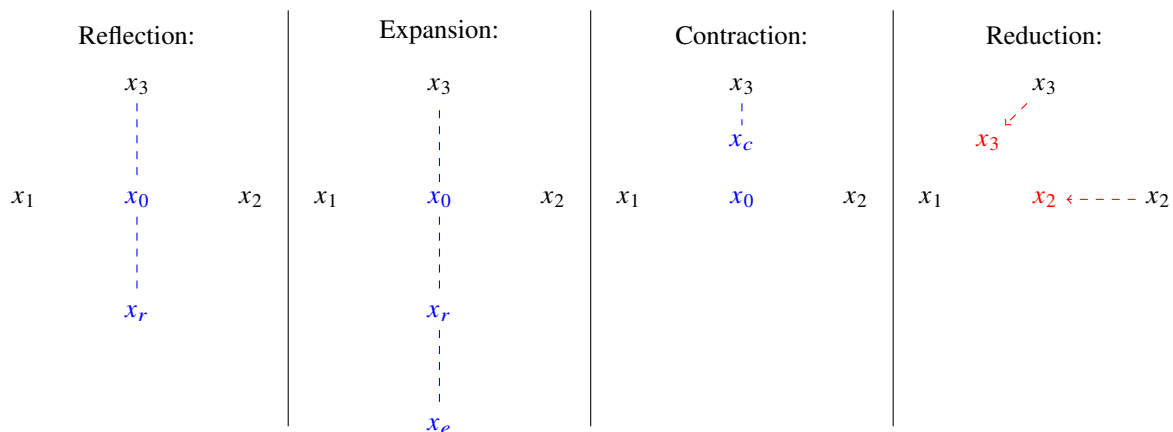
### 4.3 Numerical Solution of a single Discomfort Minimizing Problem

The main objectives of an algorithm, which solves the discomfort minimizing problem for a single agent and a fixed combination of activities, are fast runtime and good scalability for the planning horizon. These goals ensure that the overall simulation runtime does not increase to an insufficient level, when the planning horizon is extended.

The discussed problem is a multidimensional non-linear optimization problem. The description in the last section has shown why an analytic solution is not provided and a numerical solver is implemented instead. More precise, a direct search method is used. Direct search methods are advantageous, because they do not need any derivative but just function evaluations.

The direct search method implemented in our case is the *Nelder-Mead* algorithm, sometimes also called Downhill-Simplex algorithm (Nelder and Mead, 1965). The structure of the algorithm is simple and is shown in algorithm 2. The input for a  $n$ -dimensional problem contains three parts. First,  $n + 1$  points in the solution space. In our case a point is a vector of possible durations for the considered activity combination. Second, the optimized function  $f$  has to be provided, which is in our case the discomfort evaluation function. Finally, the algorithm stops, if the function values of the considered points do not differ more than a given  $\Delta$ . This is one of the possible ways to abort the main loop of the algorithm. One iteration of the Nelder-Mead algorithm for

Figure 2: Illustration of the Nelder-Mead Algorithm



two dimensions is illustrated in figure 2. After sorting the  $n + 1$  points by their function value (assume for the figure:  $f(x_1) < f(x_2) < f(x_3)$ ) the centroid  $x_0$  of the first  $n$  points is computed. Following this reflection ( $\rightarrow x_r$ ), expansion ( $\rightarrow x_e$ ) and contraction ( $\rightarrow x_c$ ) is performed. If none of the new points improves the current set by replacing  $x_{n+1}$ , the reduction is performed (all points are moved towards the best point  $x_1$ ). In summary one iteration just computes different

**Algorithm 2** Nelder-Mead Algorithm

---

```

1: while  $\max(x_i) - \min(x_i) > \Delta$  do
2:   ORDER vertices s.t.  $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$ 
3:   calculate the centroid  $x_0$  of  $x_1, x_2 \dots x_n$ 
4:   REFLECTION:  $x_r = x_0 + \alpha(x_0 - x_{n+1})$ 
5:   if  $f(x_1) \leq f(x_r) < f(x_n)$  then
6:      $x_{n+1} \leftarrow x_r$ 
7:   else
8:     EXPANSION:
9:     if  $f(x_r) < f(x_1)$  then
10:       $x_e \leftarrow x_0 + \gamma(x_0 - x_{n+1})$ 
11:      if  $f(x_e) < f(x_r)$  then
12:         $x_{n+1} \leftarrow x_e$ 
13:      else
14:         $x_{n+1} \leftarrow x_r$ 
15:      end if
16:    else
17:      CONTRACTION:  $x_c \leftarrow x_0 + \rho(x_0 - x_{n+1})$ 
18:      if  $f(x_c) < f(x_{n+1})$  then
19:         $x_{n+1} \leftarrow x_c$ 
20:      else
21:        REDUCTION:
22:        for all  $x_i$ , with  $i \in \{2, \dots, n + 1\}$  do
23:           $x_i \leftarrow \sigma(x_i - x_1)$ 
24:        end for
25:      end if
26:    end if
27:  end if
28: end while

```

---

points looking for improvement of the maintained set of points.

The algorithm is further driven by four parameters ( $\alpha$ ,  $\gamma$ ,  $\rho$  and  $\sigma$ ). The values  $\alpha = 1$ ,  $\gamma = 2$ ,  $\rho = -\frac{1}{2}$  and  $\sigma = \frac{1}{2}$  are known to be efficient for many problem descriptions (Lagarias et al., 1998) and are also chosen in our case.

The discomfort minimization problem has one characteristic that influences the implementation. The activity duration variables are bounded. The lower bound of 0 is obvious, but due to several reasons there exist also other lower and upper bounds, which reduce the solution space to a

simplex. Thus, the algorithm has to be modified at least for the reflection and the expansion steps. Whenever a new point outside of the solution space is chosen to be added to the current  $n + 1$  points, it is instead projected to the closest point on the surface of the solution simplex (Luersen et al. (2004); Wolff (2004)). Afterwards it is checked whether all points are on the same surface after the projection. If this is the case, the projected point is moved away from the surface by a small distance. This step is explained by the fact that given the  $(\alpha, \gamma, \rho$  and  $\sigma)$ -parameter as presented above the  $n + 1$  points form a non-degenerated simplex after one Nelder-Mead step, if the initial points did not form a non-degenerated simplex, too. Thus, moving the projected point further is done to preserve this feature to the considered points.

There are two limitations in the implemented version of the presented numerical solver, which are not mentioned so far. First, due to runtime issues the value for  $\Delta$  cannot be chosen arbitrary small. This can lead to a solution simplex that does not contain the optimum. Second, the algorithm might find just an local minimum. Actually, it is even not guaranteed that a local minimum is found. Thus, we have to restart the solver several times with different initial points in order to reduce the probability not to find the global minimum.

But one has to keep in mind, that the C-TAP simulation is predicting human behavior, which is always just an estimation, because human beings are not expected to act optimally. In addition, the exact numbers of the computed solution are not used in the model, but are just relative numbers to compare different options within the decision process. Keeping this in mind an estimation algorithm like the one described here is sufficient for the Long-Term C-TAP simulation, because it is expected to find fast a reasonable solution.

## 5 Future Challenges

There is still some work that has to be done in this part of the development of the Long-Term C-TAP model. First of all, the implementation of the presented Nelder-Mead algorithm has to be evaluated. Especially the scalability has to be verified. In the next step there is still some space for improvement within the implementation of the algorithm, e.g. it is not clear how to choose the best starting points or the best restart procedure. Finally, in order to have comparable results also other algorithms have to be checked and implemented. Another option to solve the discomfort minimization problem is the usage of a steepest descent method (Fletcher and Powell, 1963). These methods use the derivative to iterate towards a local optimum.

## 6 Conclusion

This paper presents the idea to use a continuous target-based activity planning model to simulate long distance travel demand within a full year. The focus is the activity planning problem, which contains optimization sub-problems. These sub-problems have been solved by a direct search method, the Nelder-Mead algorithm. Despite the introduction of a reasonable solver, there are some future challenges remaining. Nevertheless, the Long-Term C-TAP simulation reached the next step towards a stable long-term travel demand generator.

## 7 Acknowledgements

We are grateful for the INVERMO data, which was kindly made available by Bastian Chlond, KIT, Karlsruhe.

We would also like to acknowledge the Swiss National Science Foundation (SNF) for providing funds to the authors.

## 8 References

- Arentze, T. A. and H. J. P. Timmermans (2006) A new theory of dynamic activity generation, paper presented at the *85th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2006.
- Arentze, T. A. and H. J. P. Timmermans (2009) A need-based model of multi-day, multi-person activity generation, *Transportation Research Part B: Methodological*, **43** (2) 251–265.
- Balmer, M. (2007) Travel demand modeling for multi-agent traffic simulations: Algorithms and systems, Ph.D. Thesis, ETH Zurich, Zurich, May 2007.
- Beser, M. and S. Algers (2001) Sampers - the new swedish national travel demand forecasting tool, in L. Lundqvist and L.-G. Mattsson (eds.) *National Transport Models: Recent Developments and Prospects*, Advances in Spatial Science, 101–118, Springer, Stockholm.
- Bhat, C. R., S. Srinivasan, J. Y. Guo and A. Sivakumar (2003) Activity-based travel-demand modeling for metropolitan areas in Texas: A micro-simulation framework for forecasting,

- Final Report*, Texas Department of Transportation, University of Texas, Austin, February 2003.
- Erhardt, G. D., J. Freedman, A. Stryker, H. Fujioka and R. Anderson (2007) Ohio long-distance travel model, *Transportation Research Record*, **2003**, 130–138.
- Fletcher, R. and M. J. D. Powell (1963) A rapidly convergent descent method for minimization, *The Computer Journal*, **6** (2) 163–168.
- Frick, M. and B. Grimm (2014) Langstreckenmobilität: Aktuelle Trends und Zukunftsperspektiven, *Research Report*, Institut für Mobilitätsforschung, Munich.
- Janzen, M. and K. W. Axhausen (2015) Long-Term-C-TAP Simulation: Generating Long Distance Travel Demand for a full Year, paper presented at the *94th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2015.
- Janzen, M., D. Charypar and K. W. Axhausen (2014) Long Term Simulation of a Continuous Target-Based Model, paper presented at the *14th Swiss Transport Research Conference*, Ascona, May 2014.
- Lagarias, J. C., J. A. Reeds, M. H. Wright and P. E. Wright (1998) Convergence properties of the nelder-mead simplex method in low dimensions, *SIAM Journal of Optimization*, **9**, 112–147.
- Limtanakool, N., M. Dijst and T. Schwanen (2006) On the participation in medium. and long-distance travel: A decomposition analysis for the uk and the netherlands, *Tijdschrift voor economische en sociale geografie*, **97** (4) 389–404.
- Löchl, M., S. Schönfelder, R. Schlich, T. Buhl, P. Widmer and K. W. Axhausen (2005) *Untersuchung der Stabilität des Verkehrsverhaltens*, 1120, Eidgenössisches Departement für Umwelt, Verkehr, Energie und Kommunikation, Berne.
- Luersen, M. A., R. Le Riche and F. Guyon (2004) A constrained, globalized, and bounded Nelder-Mead method for engineering optimization, *Structural and Multidisciplinary Optimization*, **27** (1-2) 43–54.
- Märki, F. (2014) An Agent-Based Model for Continuous Activity Planning of Multi-Week Scenarios, Ph.D. Thesis, ETH Zurich, Zurich.
- Märki, F., D. Charypar and K. W. Axhausen (2012a) Location choice in a continuous model, paper presented at the *13th International Conference on Travel Behaviour Research (IATBR)*, Toronto, July 2012.
- Märki, F., D. Charypar and K. W. Axhausen (2012b) Target driven activity planning, paper presented at the *91st Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2012.

- Märki, F., D. Charypar and K. W. Axhausen (2012c) Validation of a continuous simulation model for daily travel, paper presented at the *12th Swiss Transport Research Conference*, Ascona, May 2012.
- Märki, F., D. Charypar and K. W. Axhausen (2013) Integration of Household Interaction with a Continuous Simulation Model, paper presented at the *13th Swiss Transport Research Conference*, Ascona, April 2013.
- Miller, E. J. (1996) Microsimulation and activity-based forecasting, paper presented at the *Activity-Based Travel Forecasting Conference*, New Orleans, June 1996.
- Nagel, K. and G. Flötteröd (2009) Agent-based traffic assignment: Going from trips to behavioral travelers, paper presented at the *12th International Conference on Travel Behaviour Research (IATBR)*, Jaipur, December 2009.
- Nelder, J. A. and R. Mead (1965) A simplex method for function minimization, *The Computer Journal*, **7** (4) 308–313.
- Pendyala, R. M., R. Kitamura, A. Chen and E. I. Pas (1997) An activity-based microsimulation analysis of transport control measures, *Transport Policy*, **4** (3) 183–192.
- Schelling, T. C. (1971) Dynamic models of segregation, *The Journal of Mathematical Society*, **1** (2) 143–186.
- Wolff, S. (2004) A local and globalized, constrained and simple bounded nelder-mead method, *Technical Report*, Bauhaus-Universität Weimar.