Conference Paper

Population generation for continuous long-distance travel demand simulations

Author(s):
Janzen, Maxim; Axhausen, Kay W.

Publication Date:
2015-07

Permanent Link:
https://doi.org/10.3929/ethz-b-000105907

Rights / License:
In Copyright - Non-Commercial Use Permitted
Population Generation for Continuous Long-Distance Travel Demand Simulations

Maxim Janzen
Kay W. Axhausen

Institute for Transport Planning and Systems, ETH Zurich

14TH INTERNATIONAL CONFERENCE ON TRAVEL BEHAVIOUR RESEARCH
WINDSOR - 19-23 July, 2015
Population Generation for Continuous Long-Distance Travel Demand Simulations

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

July 2015

Abstract

Microsimulations have become more important in the field of travel demand generation over the last years. However, most of the simulations developed focus on daily-life or short-term travel demand, although the share of the overall traffic volume caused by journeys not undertaken in daily life is increasing. The main goal of this research is the development of a new microsimulation, which is able to generate long-distance travel demand for a long period of time, say 6 or 12 months. An important step in this development is the implementation of a population generator. This paper describes in detail the population of the underlying microsimulation. It shows the problems facing a synthesis of a population for a long-term simulation and presents diverse solutions including an advice for the population generation of Long-Term C-TAP. The presented solutions involve the concept of reproducing (joint) distributions as well as the idea of using statistical models.

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

Preferred citation style
1 Introduction

To date, travel demand generation for microscopic traffic flow simulation and travel demand model focuses on reproducing and predicting 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. These activities and the corresponding journeys are the focus of the microsimulation discussed in this paper.

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 apply the Long-Term Continuous Target-based Activity Planning (C-TAP) model, which was already used to simulate travel demand for one year (Janzen and Axhausen, 2015). The model was initially proposed by Märki (2014) to simulate in detail mid-term travel demand (six-weeks).

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. 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, tunnels or airports, which is very useful for the cost-benefit analysis of this investments. Additionally, results for long-distance travel demand can be combined with short-term traffic simulations to get a complete image of the real world.

One major task within the development of the described simulation is the creation of a population. The generation of a representative population is crucial for all microsimulations. A synthetic population for Long-Term C-TAP has to provide all attributes, which are needed for a definition of the virtual agents within the simulation.

We assume that appropriate disaggregated data sources are available for the region considered. These sources are either surveys or mobile phone positioning data. A survey (INVERMO, Chlond et al. (2006)) has been already used to create a synthetic population as a copy of the real world respondents (Janzen and Axhausen, 2015). Two limitations are associated with this simple approach. On the one hand the population has not arbitrary size, on the other hand the input data set is huge (the whole survey), but preferably has to be reduced to a smallest possible number of parameters. We will show in this work how to overcome these limitations and as a side effect also ensure more control to the user.
The remainder of this paper is structured as follows. First, we summarize the Long-Term C-TAP model including the concepts of targets, activities and the continuous decision making process. After that, we provide a complete definition of the needed population. In the next step, three different ideas of a flexible population generation are presented. Two of the approaches and their implementations are compared in terms of runtime and their level of accuracy. The problems limiting the usability of the third approach are shown as well. We conclude the paper with a consideration on limitations of the current population generation due to the given data source and have an outlook on potential improvements.

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 (Bhat et al., 2003 or Miller, 1996). Nowadays agent based simulations make a notable contribution to the field of transportation research (Balmier, 2007).

Microscopic travel demand simulations simulate the (traveling) behavior of virtual agents individually. One of the well known approaches is the one proposed by Balmier (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.

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 (Erhardt et al. (2007)) as well as used for traffic forecast (Besser and Algiers, 2001). Recently, different surveys were also analyzed to derive an outlook on the future of long distance travel demand (Frick and Grimm, 2014).
The usage of a continuous target-based model for a long term simulation was introduced recently by 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.

Finally, population synthesis used for microsimulations is part of recent research, too. It can be solved by different approaches (Müller and Axhausen, 2011). Usually it is divided in two main stages, fitting and allocation (Bowman, 2009). This approach has to be modified in the case long-distance travel demand, because it is based on marginal distributions (e.g. census data), which are not available on a big scale for long-distance travel behavior.

### 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 arbitrary 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 is a comparably simple task in this case.

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. During the execution of the simulation all agents are trying to satisfy their pre-defined targets. 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. We use the following two types for our simulation:

- 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).
- 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 target type 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.

### 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 percentage-of-time 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 subsection 3.4.

### 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.
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 ← UpdateAgentState(agent)
4:         nextActivity ← MakeDecision(agent, state)
5:         agent.execute(nextActivity)
6:     end for
7:     nextTimeStep = minimum( all execution endpoints)
8:     proceed to nextTimeStep
9: end while

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 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 subsection 3.4.

3.4 Continuous Activity Planning

The implementation of a continuous activity planning has to handle an important restriction. No replanning is possible. Hence, a decision cannot be revoked. As a consequence the decision process has to include the activities executed prior to the decision and predicting the best future activities as well. We value any potential decision by the discomfort function:

\[
D(t) = \sum_{k=1}^{n} (f_{\text{target}}^k(t) - f_{\text{state}}^k(t))^2 \cdot w_k,
\]

where \( n \) is the number of targets and \( w_k \) a bandwidth normalization factor. The function \( f_{\text{target}}^k(t) \) describes the target value at a given point of time \( t \), while \( f_{\text{state}}^k(t) \) describes the state value at \( t \).

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 core of the activity planning is the discomfort function, which is based on two value types. On the one hand, there are state values, which are the result of the previous agents behavior (e.g. a state value of a percentage-of-time target is low, if the corresponding activity is not executed). On the other hand, there are target values, which are the static part of an agent. Thus, the definition of the targets is a topic of the simulation initialization. More precise, it is the main task of the population synthesis. This relationship also justifies the implementation of an accurate agent generator. All targets are part of the population and all decisions heavily depend on targets. Therefore, there is a direct connection between the population and the outcome of the simulation. The population of Long-Term C-TAP and its generation is topic of the next section.

4 C-TAP Population

The population of the Long-Term C-TAP simulation is crucial for its outcome and therefore also for the value of this simulation. Thus, it is mandatory to implement an accurate population generator. This section introduces a definition of a population or rather of the virtual agents composing the population. Afterwards, diverse approaches for a generation are presented.

4.1 Definition

The (Long-Term) C-TAP simulates the travel behavior of virtual agents, which are defined by behavioral targets. Hence, the outcome of the simulation is dependent on the definition of those targets. As shown by Janzen and Axhausen (2015) we need to define a percentage-of-time target as well as a duration target for each activity of each agent. We assume that the set of remote activity purposes is predefined. These activity purposes are usually considered to be: work related purpose (e.g. business trips), private purpose (e.g. visiting family) and holidays. The set of purposes is completed by the daily-life purpose, which sums up all activities undertaken at a place that is closer than 100km to the agents home. In addition, daily commuting is also considered to be a daily-life activity even though it might include a trip that is longer than 100km.

For a definition of an agent in the Long-Term C-TAP simulation it is useful to use a hierarchical structure consisting of travel purposes, activities and targets. Following
this structure the parameters needed for a complete description of an agent are the following:

- for each purpose type besides daily life: a set of activities (e.g. long holidays and short holidays)
- for the daily-life purpose: one single activity
- for each activity:
  1. the share of time attempted to be spent on this type (defining a percentage-of-time target)
  2. the duration attempted to be spent on each execution of this type (defining a duration target)

This set of parameters is a comparably simple description of the travel behavior of a single person. Though, it is sufficient for the level of detail Long-Term C-TAP is simulating (Janzen and Axhausen, 2015). In summary, a pair of targets for each activity defines a virtual agent and a set of agents forms a synthetic population for the Long-Term C-TAP model.

Nevertheless, the definition of all targets and activities does not complete the initialization of the simulation. First, the user can also introduce preferences regarding the weekday and specific calendar weeks for the execution of each activity. These preferences are set as a global parameter. For instance, private trips likely take place on weekends and business trips are very unlikely done on the 51st and 52nd calendar week. Thus, these preferences are not part of the population synthesis. Second, for each activity a set of possible locations has to be provided and in addition for all pairs of locations a travel time has to be computed. Both aspects are ignored in the population generation described in this chapter, because it is not directly connected to the definition of the targets which are the core of the virtual agents.

4.2 Population Generation

The Long-Term C-TAP simulation requires a synthetic population consisting of a set of agents, which themselves are defined by behavioral targets. Three algorithms for population synthesis are presented below.

We assume in this section that a long-term survey is available, where respondents have reported their long-distance trips. This is reasonable, because on one hand such surveys are existent (like Chlond et al. (2006)) and on the other hand other useful and reliable sources for our purpose (like mobile phone positioning data) are very rare. We assume also that the input data source contains information for the whole simulation period. In our case this period is one year. This assumption is done, because scaling up a smaller data set leads to a lost of information on seasonal preferences.
A long-term survey has been already used to create a synthetic population for Long-Term C-TAP by copying the real world respondents (Janzen and Axhausen, 2015). This is a very inflexible way to generate a population. We will show how to improve this simple approach. There are three main requirements for a population generator, which have to be met. First, it has to create a realistic image of the simulated population. Second, the size of the population has to be arbitrary. Third, the required input data has to be as small as possible. To achieve this two requirements we aggregate the given data as much as possible without losing the important information and afterwards synthesize a population from the aggregated data set. We will implement and test three different approaches realizing this framework:

1. Independent attributes: for each necessary variable we compute the independent distribution. Based on this distribution we perform the inverse transform sampling for each attribute.
2. Joint distributions: the different targets for each activity are not independent. So a Copula function (Embrechts, 2009) is computed based on the joint distribution and used for the initialization of the population.
3. Statistical model: a statistical model of a the long term travel behavior will be computed based on given data or used from existing sources. The significant sociodemographic parameters will be taken from a real world population to generate a synthetic population using the statistical model.

The implementation details of all three approaches are presented in the remainder of this subsection.

### 4.2.1 Independent Attributes

As described in the previous section each agent of a synthetic population is defined by a number of attributes. For our first approach we assume that all attributes are independently distributed. We choose this idea in order to have a reference strategy with the lowest possible level of complexity. We also assume that the distributions for the following variables are available:

- the number of activities per purpose,
- the duration of an activity per purpose and
- the number of activity executions within one year (equals to the frequency).

This assumption is reasonable, if a long-term travel survey is available. The distributions can be extracted easily in this case. Note that we cluster trips from the survey with the same purpose and same duration to a single activity (as defined in subsection 4.1). Given these distributions we can run the algorithm 2. The input of the algorithm mainly consists of these independent distributions and creates agents in a loop until the desired size of the population is reached. For every newly created agent a number of activities for every
purpose is generated following the corresponding distribution. The respective number of activities is then created and assigned to a duration target and a percentage-of-time target, which are both created following the respective distributions. The latter is computed by the multiplication of frequency and duration.

Algorithm 2 Independent Distribution Population Algorithm (IDPA)

Input: Population size $n$ and for all purposes $P$ the following Distributions:

- $N_P$ (number of activities),
- $D_P$ (duration),
- $F_P$ (frequency)

Output: A synthetic population for Long-Term C-TAP

1: while \( \text{size(Population)} < n \) do
2: Create new agent $V$
3: for all Purposes $P$ do
4: \[ \text{num}_\text{act}_P \leftarrow \text{ITS}(N_P) \]
5: for all $i \in [1, \cdots, \text{num}_\text{act}_P]$ do
6: Create new activity $A$ for $V$
7: \[ \text{freq}_\text{target}_A \leftarrow \text{ITS}(F_P) \]
8: \[ \text{dur}_\text{target}_A \leftarrow \text{ITS}(D_P) \]
9: \[ \text{perc}_\text{target}_A \leftarrow \text{dur}_\text{target}_A \ast \text{freq}_\text{target}_A \]
10: if $\text{perc}_\text{target}_A \neq 0$ and $\text{dur}_\text{target}_A \neq 0$ then
11: Add $\text{perc}_\text{target}_A$ and $\text{dur}_\text{target}_A$ to target set of $V$
12: end if
13: end for
14: end for
15: if $V$ has a non-empty target set then
16: Add home activity $H$ and the respective $\text{dur}_\text{target}_H$ to $V$
17: Add $V$ to population
18: end if
19: end while

In order to extract values from the given independent distributions we use the Inverse Transform Sampling (ITS), which can be applied here since we can compute the cumulative distribution function (Devroye, 1986). Finally, there is also a chance that at an agent has an empty set of targets after an iteration of the creation process. This happens, if all numbers of activities for all purposes equals to zero. In this case the agent is not added to the population. In the other case the agent receives also a home/daily-life activity and is added to the population.

Of course, arbitrary continuous distributions cannot be used here but have to be discretized. The distributions of number of activities and frequency (which is measured by the number of executions within one year) are discrete by their nature. The distribution of the activity duration is computed after segmenting the duration space into intervals. A call of ITS then returns a specific interval. Within each interval we assume a uniform distribution. The level of accuracy of the whole algorithm is of course dependent on the segmentation process.
4.2.2 Joint Distribution of Attributes

In the previous approach we assumed that all attributes of an agent are independent variables. This assumption is simplifying the real world too much. To overcome this simplification we adjust the previous algorithm, such that just two types of joint distributions are used instead of several independent ones. First of all, it is reasonable to presume that the number of activities for one purpose is not independent from the number of activities for other purposes. Thus, our second algorithm requires the joint distribution of the number of activities per year for all purposes, i.e. we need a function which provides a probability for the number of activities subdivided by purpose. For example, the probability for one business trip, two private trips and one holiday trip per year is quite high as this is quite typical for today’s society. After creating all activities a pair of targets is needed for each activity. It is obvious that also the percentage-of-time target and the duration target should not be computed independently. Therefore, we assume also that for each travel purpose a joint probability distribution for the frequency (defined as number of executions per year) and duration is available.

In both cases of the joint distributions the parameters are either discrete (number of activities, frequency) or discretized (duration) as in the previous algorithm. Thus, an inverse cumulative function can be computed and a multivariate ITS can also be applied. This is done in algorithm 3. The structure of the algorithm is mainly the same as in algorithm 2, but the inverse sampling is based on multi-dimensional functions.

Algorithm 3 Joint Distribution Population Algorithm (JDPA)

Input: Population size \( n \) and for all purposes \( P \) the following joint distributions:
\( JD_{\text{Num}} \) (number of activities for all purposes),
\( JD_{FD_P} \) (frequency-duration per purpose \( P \))

Output: A synthetic population for Long-Term C-TAP

1: while \( \text{size(Population)} < n \) do
2: Create new agent \( V \)
3: \( (\text{num}_\text{act}_{P1}, \cdots, \text{num}_\text{act}_{Pn}) \leftarrow \text{ITS}(JD_{\text{Num}}) \)
4: for all Purposes \( P \) do
5: for all \( i \in [1, \cdots, \text{num}_\text{act}_P] \) do
6: Create new activity \( A \) for \( V \)
7: \( (\text{freq}_\text{target}_A, \text{dur}_\text{target}_A) \leftarrow \text{ITS}(JD_{FD_P}) \)
8: \( \text{perc}_\text{target}_A \leftarrow \text{dur}_\text{target}_A \times \text{freq}_\text{target}_A \)
9: end for
10: end for
11: Add \( V \) to population
12: end while

In contrast to algorithm 2 there is no possibility that an agent without any activity is generated unless the probability for zero activities for all purposes is positive. This leads
4.2.3 Simulating a Statistical Model

Another way to define agents for Long-Term C-TAP is based on the assumption that all attributes of an agent are connected with each other. More specific, we assume that they all depend on an some virtual sociodemographic attributes. Sociodemographic attributes like age or sex were not considered so far, because they are not relevant for the core simulation. In order to generate a virtual population using this idea first a model describing trip attributes in dependency of sociodemographic values is needed. Given this model we can generate activities and targets for a virtual population, which includes the required agent attributes. This is shown in algorithm 4.

Algorithm 4 Statistical Model Algorithm

Input: Population $POP$ with sociodemographic attributes and for each purpose $P$ models $MOD_A_P$, $MOD_D_P$ and $MOD_F_P$ explaining the number of activities, the duration and the frequency

Output: A synthetic population for Long-Term C-TAP

1: for all agents $V$ from $POP$ do
2:     for all Purposes $P$ do
3:         $num_{act_P} \leftarrow MOD_A_P(V)$
4:             for all $i \in [1, \cdots, num_{act_P}]$ do
5:                 Create new activity $A$ for $V$
6:                 $freq_{target_A} \leftarrow MOD_F_P(V)$
7:                 $dur_{target_A} \leftarrow MOD_D_P(V)$
8:                 $perc_{target_A} \leftarrow dur_{target_A} \ast freq_{target_A}$
9:             if $perc_{target_A} \neq 0$ and $dur_{target_A} \neq 0$ then
10:                Add $perc_{target_A}$ and $dur_{target_A}$ to target set of $V$
11:        end if
12:     end for
13: end for
14: end for

The level of complexity is higher in comparison to the first two approaches. Also more input data is needed here, because the algorithm requires statistical model and in addition a virtual population with sociodemographic attributes. The biggest limitation is the fact that statistical models have to computed prior to a simulation. Finding a good model to explain long-distance travel behavior is a non-trivial task, which requires a lot of additional work. But on the other hand, once good models are available this population generator is expected to be the most accurate one. In addition, the user of the simulation can easily adjust the input in order to meet any other society structure.

to exactly $n$ iteration if the outer while-loop, which is likely to be less than in the case of independent distributions.
4.2.4 Discussion of the three Approaches

All approaches presented here are following the same main strategy. They analyze a given long-term survey, reduce the number of parameters to a small set and finally use these parameters to generate a population of arbitrary size. Nevertheless, they differ in the way they are implemented. Our three initial goals for a population generator were the following. The populations have to be of arbitrary size. They have to be a realistic image of the real world. And the size of the required input data should be small. All three presented algorithms can generate populations of arbitrary size. The accuracy is investigated in section 5. But in terms of the required amount of data the first two algorithms are superior to the algorithm using statistical models. The latter needs in addition to the statistical models also a population with sociodemographic attributes. Therefore, ignoring the accuracy the first two algorithms are preferred.

5 Results

We implemented all three approaches and try to investigate the differences in runtime as well as how well they match the real world. It turns out that the algorithm using statistical models cannot be initialized in a useful way. So the comparison of the real world matching is reduced to the other two approaches. This measurement of accuracy will be reviewed on a detailed level by a comparison of a survey with the computed populations based on this survey.

5.1 Initialization of the Algorithms

As we want to compare the presented algorithms, we have to make sure that they all have the same conditions. First of all, we use the same survey as data source (Chlond et al., 2006). We also use the same travel purposes, namely the work-related purpose, the private purpose and the holiday purpose. Also the locations and travel times, which are needed for a full calibration of the simulation, are uniformly distributed in the respective space for all algorithms.

The independent distributions as well as the joint distributions needed for the first two algorithms can be extracted easily from the given survey. The statistical models needed for the third algorithm can not be extracted automatically. Several model types were evaluated but no model could be found, which explains the duration or frequency of trips in dependency of the sociodemographic variables provided by the underlying survey. None of them has turned out to be significant in any investigated model. Thus, one has to use more complex models than it has been done so far.
5.2 Runtime

The population synthesis is the main part of the initialization and thus also a substantial part of the whole simulation. Therefore, we also investigate whether the runtimes of the presented approaches differ. We use all three algorithms to generate different sizes of populations in order to see the scalability of generators. To avoid capturing single outliers each generator is run 10 times for each considered population size. The runtime of the statistical model algorithm is also evaluated, even though no good model was available. For the runtime evaluation linear regression models were estimated. All sociodemographic attributes (sex, age, level of education and employment type) were not significant, but not rejected in order to have a model of a reasonable size.

The average runtimes including the maximal deviations in the parenthesis are shown in table 1. All runs were done on the same machine (4 CPUs of 1200 MHz and 8 GB RAM) and no parallelization technique was implemented here.

<table>
<thead>
<tr>
<th>Population Size</th>
<th>Independent Variables</th>
<th>Joint Distributions</th>
<th>Statistical Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>100,000 agents</td>
<td>44.7s (2.4s)</td>
<td>45.5s (0.5s)</td>
<td>47.6 (1.2s)</td>
</tr>
<tr>
<td>200,000 agents</td>
<td>83.2s (0.9s)</td>
<td>88.4s (1.9s)</td>
<td>91.2 (2.3s)</td>
</tr>
<tr>
<td>300,000 agents</td>
<td>127.0s (3.6s)</td>
<td>132.5s (4.7s)</td>
<td>145.8 (6.3s)</td>
</tr>
</tbody>
</table>

Several interesting facts can be observed in the runtime evaluation. First of all, the scalability of all three algorithms is good. The runtime is increasing linearly with the number of agents. This is not surprising, because all approaches have one main loop iterating over the agents. Another remarkable issue is the low variation in the runtime. Thus, all algorithms are robust in terms of runtime.

The most important fact is the overall low level for the runtimes. We can generate several hundred thousands of agents within a couple of minutes. These runtimes are sufficient for long-term simulations and do not need any further improvement. Finally, it is also noticeable that all runtimes are similar within the same population size. This similarity is explained by the big share of runtime taken for the initialization of the agents, which is the same for all approaches. The differences can be found in the way the attributes are computed, not in the way they are stored.

Note that the runtime analysis does not include the (manual) work that has to be done prior to the population generation. This includes the computation of distributions or required models.
5.3 Validation

More important than a low runtime is the accuracy of the approaches. In our case we need to validate whether the synthetic populations created by the implemented methods, which were proposed in section 4.1, represent the real world.

To measure the accuracy we create a reference population, which is built based on the INVERMO data set (Chlond et al., 2006). For every respondent a virtual agent is created and for each reported journey a new activity is added. Also the percentage-of-time targets and duration targets are just extracted from the reported journeys. Finally, activities with the same purposes and similar durations are clustered to a new activity and new targets respectively.

The same survey is also used to extract the distributions needed for the Independent Distribution Population Algorithm (IDPA) and Joint Distribution Population Algorithm (JDPA). Both algorithms are run in order to create 1944 agents, which is exactly the number of agents in the reference population. We compare the activities of the two synthetic populations and the reference population. More specific we compare the number of activities clustered by purpose, frequency and discretized duration.

The results of this comparison are summed up below. The detailed results can be found in the appendix A. As expected the numbers of the JDPA are much closer to the reference population than the IDPA results. The JDPA population is a good image of the respondents of the underlying survey, because it reproduces in detail the reference population. Therefore, the JDPA is the approach suggests to use for the future. The advantage of the IDPA is the smoothness of the results. The durations and activities are better distributed, which is not the exact copy of the reference population but is more realistic, though.

The advantages of both algorithms can be combined by an easy procedure. The joint distributions required for the JDPA are computed. But before using them they are artificially smoothed in a step prior to the JDPA population synthesis. This idea is the suggested population synthesis unless a good statistical model is found, which explains properly the motivations of long-distance travel behavior.

The validation results also lead to the question whether the method using statistical models can be an alternative. Even if we assume that statistical models are easily available, the validation results of JDPA can not be improved much more. Additionally, a population with attributes is required before the start of the algorithm. Thus, it is questionable whether the advantages of an implementation of this approach do cancel out the drawbacks.
6 Conclusion

The presented paper focuses on the analysis of the population needed for continuous long-term microsimulation, like Long-Term C-TAP. The synthetic population needs to be artificially generated and represent the real world. Three approaches were provided to solve this task. While one of the approaches turned to be not practicable due to the lack of statistical models, the other two methods created reasonable results. The best results are achieved by the Joint Distribution Population Algorithm, which is therefore the preferred algorithm for future usage of the Long-Term C-TAP. It meets all three goals, which were set in this paper. The size of required input data is small, the generated population is a good image of the real world and the population can be of arbitrary size.

The main result of this work is the possibility to generate realistic synthetic populations of arbitrary size. This fact improves the Long-Term C-TAP simulation and pushes it towards a complete long-term travel demand simulation, which can be used for an analysis of infrastructural investments.

7 Acknowledgment

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 author.

References


A Accuracy of the Population Generators

In order to value and compare the accuracy of the IDPA and JDPA, three different populations are computed and compared. The first one is the reference population reference population, which is created as a copy of INVERMO study respondents. The other two are the synthetic populations generated by IDPA and JDPA, where the INVERMO survey was used for the initial distributions. All three populations consist of agents with activities and targets as it is required for the Long-Term C-TAP simulation. We compare for the three populations the total number of activities grouped by purpose, frequency and duration.

The results subdivided by purpose can be found in table 2 (holiday purpose), table 3 (work purpose) and table 4 (private purpose). All tables are structured the same way: each cell of the table contains three numbers. The first value belongs the reference population, while the second and the third are results of the IDPA and JDPA. Each value is the number of created activities with the duration of the corresponding column and the frequency of the corresponding row. The level of accuracy can now be measured by a comparison of the three numbers cell by cell.

In order to support the analysis of the tables graphically high (normalized) deviations in comparison to the reference population are colored. High deviations in the IDPA population are colored blue and high deviations in the JDPA population are colored green. Higher opacity of the color equals to higher deviation.

The analysis of the synthetic populations shows that the JDPA population is very close to the reference populations. There is almost no substantial difference between these two. On the other hand, the IDPA population has much bigger differences to the reference population. But the distribution of the absolute number of activities is much smoother in comparison to the other two populations. Nevertheless, the deviation of the reference population is too high in absolute and relative numbers as well.

Note that the tables in this appendix show results of a single run of IDPA and JDPA. But both algorithms were executed more than ten times without any bigger change in the results. This fact shows also that both algorithms are robust.

In summary, the JDPA creates much better results in terms of reproducing the given distributions. This result was expected as the algorithm is capturing the joint distributions of the given data sources.
Table 2: Holiday activities for the reference population and the IDPA and JDPA synthetic populations grouped by duration and frequency

<table>
<thead>
<tr>
<th>Frequency [1/365days]</th>
<th>0h-2h</th>
<th>2h-6h</th>
<th>6h-12h</th>
<th>12h-24h</th>
<th>24h-2days</th>
<th>2days-3days</th>
<th>3days-7days</th>
<th>7days-14days</th>
<th>14days-28days</th>
<th>&gt;28days</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27</td>
<td>58</td>
<td>225</td>
<td>181</td>
<td>189</td>
<td>234</td>
<td>419</td>
<td>356</td>
<td>72</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>17</td>
<td>14</td>
<td>159</td>
<td>116</td>
<td>140</td>
<td>197</td>
<td>440</td>
<td>426</td>
<td>248</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>48</td>
<td>252</td>
<td>208</td>
<td>186</td>
<td>218</td>
<td>420</td>
<td>355</td>
<td>75</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>37</td>
<td>14</td>
<td>26</td>
<td>16</td>
<td>52</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>14</td>
<td>18</td>
<td>22</td>
<td>18</td>
<td>63</td>
<td>74</td>
<td>40</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>40</td>
<td>17</td>
<td>31</td>
<td>18</td>
<td>58</td>
<td>58</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3: Work related activities for the reference population and the IDPA and JDPA synthetic populations grouped by duration and frequency

<table>
<thead>
<tr>
<th>Frequency [1/365days]</th>
<th>0h-2h</th>
<th>2h-6h</th>
<th>6h-12h</th>
<th>12h-24h</th>
<th>24h-2days</th>
<th>2days-3days</th>
<th>3days-7days</th>
<th>7days-14days</th>
<th>14days-28days</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>69</td>
<td>84</td>
<td>79</td>
<td>122</td>
<td>53</td>
<td>81</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>32</td>
<td>44</td>
<td>70</td>
<td>131</td>
<td>59</td>
<td>146</td>
<td>33</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>60</td>
<td>63</td>
<td>85</td>
<td>118</td>
<td>58</td>
<td>72</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>9</td>
<td>17</td>
<td>22</td>
<td>36</td>
<td>11</td>
<td>9</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>8</td>
<td>14</td>
<td>13</td>
<td>22</td>
<td>6</td>
<td>24</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>7</td>
<td>24</td>
<td>20</td>
<td>38</td>
<td>4</td>
<td>9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>10</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4: Private activities for the reference population and the IDPA and JDPA synthetic populations grouped by duration and frequency

<table>
<thead>
<tr>
<th>Frequency [1/365days]</th>
<th>0h-2h</th>
<th>2h-6h</th>
<th>6h-12h</th>
<th>12h-24h</th>
<th>24h-2days</th>
<th>2days-3days</th>
<th>3days-7days</th>
<th>7days-14days</th>
<th>14days-28days</th>
<th>&gt;28days</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/365 days</td>
<td>199</td>
<td>275</td>
<td>295</td>
<td>358</td>
<td>386</td>
<td>201</td>
<td>255</td>
<td>56</td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>2/365 days</td>
<td>91</td>
<td>184</td>
<td>246</td>
<td>310</td>
<td>397</td>
<td>263</td>
<td>420</td>
<td>165</td>
<td>64</td>
<td>0</td>
</tr>
<tr>
<td>3/365 days</td>
<td>192</td>
<td>283</td>
<td>319</td>
<td>363</td>
<td>389</td>
<td>190</td>
<td>242</td>
<td>47</td>
<td>26</td>
<td>8</td>
</tr>
<tr>
<td>4/365 days</td>
<td>30</td>
<td>41</td>
<td>45</td>
<td>61</td>
<td>59</td>
<td>39</td>
<td>44</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5/365 days</td>
<td>13</td>
<td>21</td>
<td>25</td>
<td>44</td>
<td>27</td>
<td>20</td>
<td>17</td>
<td>6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6/365 days</td>
<td>33</td>
<td>43</td>
<td>47</td>
<td>54</td>
<td>59</td>
<td>45</td>
<td>40</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>7/365 days</td>
<td>5</td>
<td>7</td>
<td>14</td>
<td>19</td>
<td>18</td>
<td>14</td>
<td>10</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8/365 days</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9/365 days</td>
<td>6</td>
<td>2</td>
<td>18</td>
<td>15</td>
<td>15</td>
<td>9</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/365 days</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/365 days</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/365 days</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>