Divide-and-conquer one-step simulator for the generation of synthetic households

Marija Kukic * Xinling Li * Michel Bierlaire *

April 8, 2023

Report TRANSP-OR 230408
Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

*École Polytechnique Fédérale de Lausanne (EPFL), School of Architecture, Civil and Environmental Engineering (ENAC), Transport and Mobility Laboratory, Switzerland, {marija.kukic, li.xinling, michel.bierlaire}@epfl.ch

Abstract

This paper presents a novel simulation approach for generating synthetic households, addressing several literature gaps from the methodological viewpoint. The generation of hierarchical datasets such as complete households is challenging since it must guarantee replication of the marginal distributions of each attribute while maintaining the consistency between the layer of individuals and the layer of households. Usually, these layers are generated in two sequential processes. This paper focuses on designing a one-step simulator that simultaneously integrates the relationships within both layers. One of the major advantages is that it reduces the risk of generating illogical households. In order to deal with the curse of the dimensionality of the simulation method, we propose a so-called divide-and-conquer way of modeling, that simplifies the problem by reducing the number of variables so that we maintain the best trade-off between the accuracy and efficiency of the generation process. We test our method in a case study based on the 2015 Swiss census data, where we compare our method with state-of-the-art approaches. The results suggest that we can achieve twice as fast household generation by preserving the same accuracy compared to other simulation methods.

Keywords: Population synthesis, Markov Chain Monte Carlo simulation, Gibbs sampling, Activity-Based models

1 Introduction

Transportation science today is tasked with predicting the complex mobility needs of individuals, which necessitates the use of advanced mobility and travel demand models. The models for predicting activity and travel-related decisions of individuals are called Activity-Based Models (ABM) (Axhausen, 2000; Castiglione et al., 2014). The performance of these models strongly relies on the quality of the data used for calibration. Therefore, it is of paramount importance that we provide high-quality data that mimics the activity and travel-related decisions.

Highly-sensitive data such as the population census and travel activity information are extremely valuable in transportation science as they provide detailed insights into traveler behavior. Such data are used to either inform decision-makers or to design accurate simulation models of travelers. Traditional population census or travel survey datasets contain personal information about individuals and households. Nevertheless, the datasets made available to researchers and analysts do not fully represent the whole population, and the unprocessed data are unavailable due to privacy policies. Often they are either anonymized by removing characteristics from the dataset or by extracting micro samples that represent a small subset of the entire data. This is a problem in transportation and travel activity modeling, as the activity forecasting model's quality depends on the level of detail of the model descriptors.

To circumvent privacy and availability issues, synthetically generated data can be used. Synthetic data have similar statistical properties as the real population of interest at the aggregated level. However, they do not allow the identification of individuals (to address the privacy issue) and compile all the necessary data for scientific analysis or required by the municipalities and other stakeholders who do not have access to raw data (to address the availability issue).

In the context of synthetic population generation, the synthetic data can be at the level of individuals or households. The literature shows that synthetic population generators mostly support the generation of individual characteristics only, which consequently affects the existing research efforts in ABM. Individual information is crucial for analyzing and understanding the travel behavior, but it should not be deprived of the social context such as the household. Integrating household data into ABM methodologies would expand the model's capabilities to capture multi-individual decisions and understand mobility patterns by taking into account interactions, constraints and influence of the household (Olde Kalter and Geurs, 2016; Pougala et al., 2022).

The synthesis of households is more complex than the synthesis of individuals for multiple reasons. Apart from reproducing the individuals' characteristics, these algorithms must correctly capture relationships between household members. It is important to note that the improper replication of these multivariate distributions can yield illogical observations (e.g., households in which the child is older than the parents). Moreover, generating the complete households requires replicating more characteristics, which means capturing more correlations. An increase in complexity may result in a significant efficiency drop, given that the method's performance may differ based on the scale of the problem.

In this paper, we develop a one-step simulation framework for generating the synthetic households. We integrate the process of generating the individuals and their matching into households into one-step. We refer to it as "one-step", to distinguish it from the state-of-the-art (e.g. Casati et al., 2015) "two-step" model which first generates the household's attributes and the attributes of the owner, and then assigns the rest of the individuals to the previously generated observations. The originality of our approach lies in the fact that we generate the individual and household characteristics all at the same level while maintaining all the dependencies among them. This way of generation removes the necessity for labeling the household members based on their roles which makes our one-step method flexible to produce diverse household structures. Moreover, the existing two-step method assumes a sequential relationship among the individuals.

By ignoring some of the relationships in the sequential approach it is possible that some real-world constraints are not satisfied. In order to verify data representativity and consistency, it is necessary to provide evidence that the generated distributions reflect the distributions of the real sample. In this paper, we analyze, discuss and apply the existing validation methods in order to show that the model-driven structure allows us to have control over the generation process and to enforce the realistic relationships between different household members.

Finally, we address one of the main drawbacks of the simulation methods listed in the literature which is the difficulty to deliver acceptable results in terms of accuracy and efficiency while working with high-dimensional datasets (Farooq et al., 2013; Casati et al., 2015). With an increase in the dimensionality, the probability mass becomes more concentrated in highly correlated areas and sparser in the low-correlated areas, which leads to longer computational time since the algorithm struggles to explore the distribution effectively. We propose an improvement of our simulator called the "Divide-and-Conquer" (DAC) one-step method as explained in Section 3.1. We show how a different way of constructing conditionals may influence the overall performance of the algorithm. In an attempt to reduce

dimensionality and simplify the conditionals, we apply expert knowledge and investigate the correlation rate between different characteristics before starting the generation process. This way we aim to obtain the same accuracy as if we used the full conditionals, but in a more efficient manner. In fact, these modifications can be applied to any generating process using a Gibbs Sampler, which paves the way for using simulation-based methods in the era of big data.

The remainder of this paper is organized as follows: Section 2 covers a detailed review of the previous research in this field. In Section 3, we introduce and formally specify the proposed methodology. Finally, in Sections 4 and 5 we present the obtained results, summarize the research contributions and specify some ideas for future research.

2 Literature review

The existing methodologies for synthetic population generation can be categorized based on several criteria. Firstly, the methods can be individual-centered (i.e. one-layered population) or household-centered (i.e. two-layered, multi-level, hierarchical population) depending on the type and structure of the data they generate. The main difference between these two types of methods stems from the types of characteristics used to describe the population. The one-layered population is described only with individual characteristics such as age, gender, etc. On the other hand, the two-layered population consists of the household characteristics such as size, type, number of cars, etc. expanded with the set of associated individuals described by their own set of characteristics.

In Section 2.1, we analyze the state of the research dedicated to synthetic population generation by giving an overview of existing methodologies for both levels. Since our approach is founded upon the simulation, in Section 2.2 we delve into details on existing synthetic population simulation-based techniques. In Section 2.3, we provide an overview of the most commonly used validation metrics and describe their limitations in validating hierarchical tabular data.

2.1 Synthetic population generation: from synthetic individuals to synthetic households

Over time, an extensive literature has been developed on the analysis and comparison of different synthetic population methods. Miranda (2019) has produced a systematic review covering several decades of synthetic population generation methods applied to transportation models. Yaméogo et al. (2021) carried out an

analysis of synthetic population generation focusing only on the synthetic generators that handle hierarchies such as households.

Based on the findings of these two studies, we can divide synthetic population generation methods as presented in Table 1. We separate the methods into columns based on the characteristics they generate. Most of them were originally focused on replicating marginal distributions of only individual variables (Beckman et al., 1996). Soon after, these methods were extended to reproduce also marginals of household characteristics (Arentze et al., 2007; Choupani and Mamdoohi, 2016; Auld and Mohammadian, 2010; Farooq et al., 2013; Xu and Veeramachaneni, 2018; Garrido et al., 2019; Badu-Marfo et al., 2020; Lederrey et al., 2022). Although they can provide a good fit of marginals for both household and individual characteristics, they do not maintain the relationships between individual and household layers. Finally, the third column contains the methods that simultaneously estimate joint probabilities at the individual and household levels, thus maintaining the consistency between these two layers (Ye et al., 2009; Casati et al., 2015).

Different rows represent different groups of methods based on the paradigm they rely on: synthetic reconstruction, combinatorial optimization and statistical learning. Statistical learning methods can be further categorized into simulation-based (i.e. model-based generation) and machine learning (ML) techniques (i.e. data-driven generation). It is interesting to notice that a significant amount of research has recently been invested in developing and adapting machine learning techniques for synthetic generation of individuals. Although ML techniques might be considered as state-of-the-art for generating individuals, to the best of our knowledge, no machine learning method has been proposed for successful generation of complete households.

The first methodology that appeared for generating the synthetic individuals was based on Iterative Proportional Fitting (IPF) (Beckman et al., 1996). This approach is also known as the matrix fitting table. The concept behind the IPF is to take each marginal one at a time and change the sample's contingency table to reflect the aggregate property of the population. In the case of IPF, an increase in the number of characteristics causes exponential growth of the number of cells in the contingency table. Consequently, many combinations of characteristics with a low number of individuals lead to empty cells in the contingency table. This problem is known as the "zero-cell issue" and it has been shown that IPF fails to converge in some cases due to it (Ben-Akiva and Lerman, 1985). In addition to the scalability issue, it only produces the deterministic realization of synthetic population (Farooq et al., 2013).

Albeit being flawed in different ways, IPF has been used as a foundation for many works that proposed incremental improvements to handle household structure (Guo, 2007; Arentze et al., 2007; Choupani and Mamdoohi, 2016; Auld and Mohammadian, 2010). The problem with these methodologies is that they all require two separate steps - generation and matching. In other words, the individual and household characteristics are generated separately, applying different logic to match the people into households. Even though the marginals of the generated household characteristics might seem accurate, the separated or sequential way of generation does not necessarily maintain the relationships between households and previously generated individuals (Zhu and Ferreira, 2014).

In order to generate a complete household with all the necessary relationships between household members, the Iterative Proportional Updating (IPU) was developed (Ye et al., 2009). IPU synthesises the population by matching household and individual distributions simultaneously (Saadi et al., 2016). However, several authors pointed out there is no theoretical proof of convergence for the IPU method, that it suffers from the same issues as IPF, and requires a disaggregated initial sample which is rarely available (Lenormand and Deffuant, 2013; Zhu and Ferreira, 2014).

Another category of methods that appeared for household and individual generation was based Combinatorial Optimization (CO) (Barthelemy and Toint, 2013; Abraham et al., 2012). These methods are iterative and can work only with the available marginals. At the beginning of the process, the initial pool of households is picked randomly from the sample. In each iteration, the fit of the current sample is calculated after applying actions such as adding, updating or swapping one randomly chosen household from the sample. The algorithm stops once it reaches the desired precision. Compared to the other studies, the CO methods showed poor efficiency in generating large populations (Yaméogo et al., 2021).

In general, simulation based methods have shown good results in overcoming the previously mentioned issues related to IPF. Farooq et al. (2013) introduced a synthesis algorithm based on the Markov Chain Monte Carlo for generating the individuals (iMCMC). This method implements the Gibbs Sampling algorithm by drawing from the pre-formed conditional distributions. Compared to the IPF, the simulation method is stochastic, which helps generating a heterogeneous sample. Moreover, the simulation method is sample-free, meaning that it does not require disaggregated data as the input. In order to adapt this methodology in the context of household generation, Casati et al. (2015) propose an extension of iMCMC, the so-called "two-step" method which relies on the rule-based assignment of the

	Synthetic individuals	Synthetic households	Associated individuals
Statistical reconstruction	**1996** *Beckman et al.* **Creating synthetic baseline populations**	**2007** *Arentze et al.* **Creating synthetic household populations**	**2009** *Ye et al.* **Iterative Proportional Updating**
Simulation-based	**2013** *Farooq et al.* **Simulation based population synthesis**		**2015** *Casati et al.* **Hierarchical MCMC**
Machine Learning	**2018** *Xu et al.* **Tabular Generative Adversarial Network** **2019** *Borysov et al.* **Variational Autoencoder** **2020** *Badu-Marfo et al.* **Composite Travel Generative Adversarial Network** **2022** *Lederrey et al.* **DATGAN: Integrating expert knowladge into deep learning for synthetic tabular data**		**X**

Table 1: The overview of existing synthetic population methods with selected publications

household roles (owner, spouse, children and others). Since we use this method as a baseline for comparison with our method, we describe it in detail in Section 2.2.1.

In Yaméogo et al., 2021, authors compare different synthetic reconstruction and statistical learning methods for synthetic household generation based on various criteria as shown in Table 2. There is a counterintuitive phenomena that despite the fact that IPU has revealed different flaws and is outperformed by simulation-based methods, it is still the most cited and most frequently used method. Moreover, most of the publicly available population synthesizers use IPF as the core algorithm (Templ et al., 2017). Potential reason might be that implementation of simulation methods is usually not publicly available or tested only by their designer with a specific dataset making it less general and difficult to reproduce. This leads to a conclusion that better scalability, better fit of marginals and good efficiency do not necessarily make one method more desirable than the other. However, this might impact future users of the generated data, as the outputs of the transportation models highly depend on their input data.

	IPU	hMCMC
Referent Sample	disaggregated	no constraints
Sample size	large	small
Number of characteristics	few	few
"Zero-cell"	yes	no
Dissemination	widely used	rarely used
Heterogeneity	deterministic	stochastic

Table 2: The comparison of existing methods for synthetic household generation

As previously shown in Table 1, ML methods have recently become a popular tool for generating synthetic individuals mostly because simulation-based methods typically fail to deliver high-quality data in the context of big data. Nowadays, the Generative Adversarial Network specialized for tabular data (TGAN) by Xu and Veeramachaneni (2018) is considered to be the state of the art for generating synthetic individuals. This approach has shown great success in generating high-dimensional datasets in an accurate and computationally efficient manner. GANs implicitly learn the probability distribution of a dataset and may generate samples from it (Goodfellow et al., 2014). They consist of two neural networks called the

generator and the discriminator. The generator is trained to learn how to generate data from random noise to deceive the discriminator. The discriminator is trained to discriminate between the real and the generated data. However, this approach is completely data-driven and there is no control over the generation process, so sometimes it may happen that the generated observations do not satisfy real-world constraints.

Recently, Lederrey et al., 2022 developed a methodology to incorporate expert knowledge in ML models in the context of the synthetic data generation process. Compared to the standard GAN architecture, where the generator is typically a feedforward neural network that only takes random noise as an input, DATGAN uses an additional Directed Acyclic Graph (DAG) as an input to specify the relationships between the variables. The generator's structure is built following the provided DAG such that each node corresponds to a Long Short Term Memory (LSTM) cell and the edges represent the connections between the LSTM cells. Thus, the idea of adding a DAG aims to add more control to the generation process, as the LSTM cells in the generator enable the previous output to influence the current state of the neural network, which has been shown to improve the quality of generated data.

To the best of our knowledge, none of the ML techniques is adapted to be used in the domain of household generation. This might be due to the fact that it is difficult to impose the relationships between these two layers as it is difficult to integrate expert knowledge. On the other hand, the simulation approach is model-driven, allowing us to control the generation process by properly designing conditional distributions. It is also data-independent which comes from the fact that conditionals can be created by combining data, domain knowledge assumptions, and models (Discrete Choice Model, Machine Learning). However, as explained in Section 2.2 it is challenging to use simulation methods for generating the high-dimensional datasets due to the so-called "curse of dimensionality" phenomena. The fact that Gibbs Sampler is not suitable for generating the big datasets seems to have pushed the community towards using the ML techniques.

2.2 Simulation methods for synthetic population generation

Synthetic population is composed of vectors X described by both discrete and continuous random variables that represent individual and households characteristics denoted as $(X_1, X_2, ..., X_n)$. We only have partial knowledge of a unique joint distribution $\pi(X)$ that these variables form. The goal of Gibbs Sampler is to replicate this distribution iteratively using conditional distributions. The conditional distri-

butions are calculated beforehand for each variable and provided to the algorithm as the input. They can be obtained from different sources: from empirical distributions extracted from the data, from theoretical models with parameters calibrated on the data, from distributions reported in the literature, etc. At each iteration, using the inverse transform, the GS draws a value of one randomly picked variable from the probability distributions conditional to the fixed values of other variables.

The accuracy of the algorithm is highly influenced by the way in which we construct the probability vectors that are provided as input. By designing the conditionals, we model the relationships between the attributes and define the rules to be followed during the generation process. In the literature, the authors usually use full conditionals, in which one variable is drawn conditional to all others, assuming that this is the best way to capture all relationships. The full conditionals are not always available due to the fact that they might involve a lot of variables which causes that they cannot be expressed as a known probability distribution with a finite number of parameters. In order to address the unavailability problem, the previous authors proposed simplifying the conditionals by removing the attributes that are less informative according to expert knowledge. However, if the conditionals are too simple, then there is a possibility that some relationships are omitted which can result in generating unrealistic observations.

As previously mentioned, GS shows several limitations while dealing with high-dimensional problems (e.g., households generation) due to the so-called "curse of dimensionality". This means that the execution time grows exponentially with an increase in the number of variables. Due to sparsity and highly correlated areas, it becomes more difficult for the algorithm to explore the whole space. Because of this, the algorithm tends to iterate for a long time and requires a lot of draws to converge to the unique joint distribution. Moreover, in some extreme cases of total correlation between the two characteristics, it may end up in a degenerative state and fail to converge. To prevent this, the correlation among variables must be investigated beforehand. In Section 2.2.1 we explain the adaption of GS in the context of household generation.

2.2.1 Two-step simulation method for synthetic households generation

The two-step method involves multiple Gibbs samplers (i.e., one per each step) as shown in Figure 1. In the first step, the household size (X_{hs}), owner's age (X_{ao}) and owner's gender (X_{go}) are generated and fixed for the second step. They are generated using the full conditionals as follows:

$$\pi(X_{hs}|X_{ao} = x_{ao}, X_{go} = x_{go})$$

$$\pi(X_{ao}|X_{hs} = x_{hs}, X_{go} = x_{go})$$
$$\pi(X_{go}|X_{hs} = x_{hs}, X_{ao} = x_{ao})$$

Then, in the second step, the sequence of the following individuals described by age and gender is generated. The structure of the household always follows the same order of individuals labeled by roles (i.e. spouse, child, and other). For the generation of each successive individual, only the links with the previously generated members are considered.

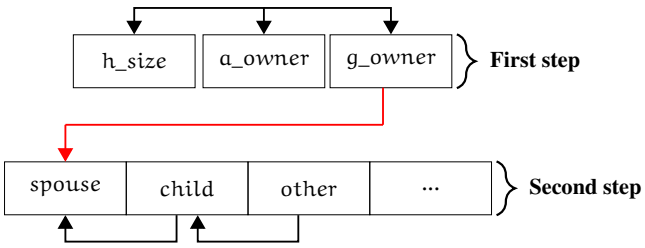

Figure 1: Two-step methodology

The owner represents the whole household since all the individuals depend on the owner's characteristics. Let us define $Y_i = (X_{hs_i} = x_{hs_i}, X_{ao_i} = x_{ao_i}, X_{go_i} = x_{go_i})$ as the i-th tuple of fixed attributes generated in the first step. Similarly, let us define $Y_i^s = (X_{as_i} = x_{as_i}, X_{gs_i} = x_{gs_i})$ and $Y_i^c = (X_{ac_i} = x_{ac_i}, X_{gc_i} = x_{gc_i})$ as the i-th tuple of spouse's and child's age and gender, respectively. The characteristics of the spouse (X_{as},X_{gs}) and child (X_{ac},X_{gc}) are conditioned on the characteristics of all previous individuals as follows:

$$\pi(X_{as}|Y_i, X_{gs} = x_{gs})$$
$$\pi(X_{gs}|Y_i, X_{as} = x_{as})$$
$$\pi(X_{ac}|Y_i, X_{as} = x_{as}, X_{gs} = x_{gs}, X_{gc} = x_{gc})$$
$$\pi(X_{gc}|Y_i, X_{as} = x_{as}, X_{gs} = x_{gs}, X_{ac} = x_{ac})$$

In order to keep the conditionals simple, the rest of the household members are characterized by age ($X_{a_{ot}}$) and gender ($X_{g_{ot}}$) are generated only conditional on the owner's, spouse's and child's characteristics:

$$\pi(X_{g_{ot}}|Y_i, Y_i^s, Y_i^c, X_{g_{ot}} = x_{g_{ot}})$$
$$\pi(X_{a_{ot}}|Y_i, Y_i^s, Y_i^c, X_{a_{ot}} = x_{a_{ot}})$$

Although this method simultaneously combines household and individual characteristics giving a good approximation of previously mentioned variables, there are some limitations. Firstly, it is difficult to expand this method since it lacks an explanation on how to introduce additional relations should we wish to add more characteristics (e.g., employment, education, marital status) other than age and gender. It also remains unclear if the addition of such variables would hinder the algorithm's performance due to the curse of dimensionality. Secondly, in order to make this method work, it is mandatory to assign roles in the pre-processing phase to all individuals based on their age and gender. This limits the method from generating diverse household types. For instance, it is impossible to generate a single parent with a child since the second member is always assumed to be a spouse. In an attempt to solve this problem, in this paper, we aim to use the household type as an essential variable in all conditionals which allows the generation of any household structure. Finally, it is computationally heavy to always generate the whole sequence of individuals in each iteration of the algorithm.

2.3 Validation of the synthetic data

In the transportation field, the most frequently used methodology for assessing the synthetic tabular data is Standardized Root Mean Squared Error (SRMSE)(Müller and Axhausen, 2011; Pritchard and Miller, 2012). The existing literature mainly uses SRMSE to validate the fit of each characteristic separately (i.e. marginals distributions) or a combination of arbitrarily selected characteristics using the Eq. 1 (Zhu and Ferreira, 2014; Garrido et al., 2019; Badu-Marfo et al., 2020).

$$SRMSE = \frac{\sqrt{\sum_{i=1}^{m} \cdots \sum_{j=1}^{n} \frac{(\pi_{i,\ldots,j}^{synth} - \pi_{i,\ldots,j}^{real})^2}{N}}}{\sum_{i=1}^{m} \cdots \sum_{j=1}^{n} \frac{\pi_{i,\ldots,j}}{N}} \tag{1}$$

Here, π^{synth} and π^{real} represent the frequency count of each unique combination of attributes $(i, ..., j)$, in the real and synthetic samples, respectively, where m and n are the numbers of possible categories of these attributes. N denotes total number of different combinations of values for attributes $(i, ..., j)$. In other words, one calculates the occurrence of unique values for each combination of

real and corresponding synthetic columns and compares them. However, the previous authors state that SRMSE is not reliable for the assessment of all combinations of attributes for high-dimensional datasets (Garrido et al., 2019). The high-dimensional datasets are usually sparse, which results in an unreasonably higher value of SRMSE. This is a consequence of the fact that N increases even if the count difference does not change (Zhu and Ferreira, 2014). Consequently, to test the fit of synthetic data, the authors usually report the SRMSE for each characteristic independently or for a combination of arbitrarily chosen characteristics that are considered the most important. However, while generating hierarchical structures such as households, with all their constituent individuals, the consideration of all multivariate distributions is crucial to demonstrate that the realism of the data and the consistency between the layer of individuals and the layer of households are preserved. By consistency, we refer to verifying that the rules derived from the expert knowledge are respected (e.g. children are not older than their parents). Realism, on the other hand, implies that the generated individual who satisfies the real-world constraints is also someone who is a good representative of the population (e.g., a small percentage of old children living with old parents). The SRMSE for each synthetic column can indicate a perfect fit (i.e. equals to zero), while all multivariate distributions might be illogical. Thus, the comparison of marginals only is not sufficient to validate the plausibility of synthetic households.

Lederrey et al., 2022 address previously mentioned problems by redefining the SRMSE to systematically test all possible combinations of all variables on different aggregation levels. With an aggregation level, we specify the number of columns that are jointly assessed. If N_v is the set of n_v columns in the dataset, instead of calculating the one frequency list for arbitrarily chosen subset $N'_v \subset N_v$, for a specified aggregation level $a \in \{1, 2, 3\}$, they calculate SRMSE for $\binom{n_v}{a}$ frequency lists. The final result is the average of all previously obtained scores. Note that this framework also allows calculating other obtained statistics such as Mean Absolute Error (MAE), Coefficient of determination (R^2), and Root Mean Square Error (RMSE).

3 Methodology

This section presents the one-step divide-and-conquer methodology for generating complete synthetic households building on the Gibbs sampler proposed by Farooq et al. (2013). We propose a methodology to generate synthetic households

composed of N individuals. A synthetic household is characterized by a sequence of household-specific variables $Z_1, ..., Z_K$ (such as household type, total number of cars, and total number of driving licenses), and N sets of individual-specific variables $X_{n_1}, ..., X_{n_L}$, where $n = 1, ..., N$ (such as age, gender, marital status, employment, and driving license). The main idea of the "one-step" approach is to consider the joint distribution of the $K + N \cdot L$ variables, and to draw from it directly using Gibbs sampling.

The general concept of Gibbs sampling relies on draws from univariate conditional distributions. In order to obtain a practical implementation of the GS for the generation of synthetic households, the specification of these conditional distributions is needed. These distributions can be constructed from contingency tables generated from the real data set, or from predictive models, estimated from the real data sets. In both cases, it is not practical to represent a full conditional. Indeed, in the case of the contingency table, a high-dimensional table contains a lot of zero cells. And in the case of a predictive model, a high number of variables is usually associated with poor statistical precision, overfitting, and poor predictive power.

The key modeling decisions rely therefore on the choice of variables that must be explicitly involved in the conditional, and the ones that can be omitted. Clearly, each of these modeling decisions depends on the specific context and data availability. In order to have a concrete description of the methodology, we focus on a specific, although reasonably general case, involving the following variables: at the household level, household type (Z_t) and number of cars (Z_c) are included; at the individual level, age (X_{na}), gender (X_{ng}), marital status (X_{nm}), employment status (X_{ne}), and whether having a driving license (X_{nl}) as shown in Table 3.

Attribute	Values
Household type (Z_t)	**Single, Single with children, Pairs, Pairs with children, Non-family**
Household size (N)	**1-17**
Number of cars (Z_c)	**1, 2, 3, 4, more than 5**
Age (X_{na})	**0-14 14-18 18-24 24-44 44-65 older than 65**
Gender (X_{ng})	**M, F**
Marital status (X_{nm})	**Single, Married, Widowed, Divorced, Unmarried**
Employment (X_{ne})	**Employed, Unemployed, Education**
Driving license (X_{nl})	**yes = 1, no = 2**

Table 3: Data description

3.1 The investigation of conditionals - divide-and-conquer

The goal of the DAC one-step methodology is to simplify conditionals to find the best trade-off between accuracy and efficiency while assuring the generation of realistic observations. To make our modeling decision, we test full and simplified conditionals for each variable. Full conditionals involve all variables, while simplified conditionals contain only one variable that is considered as the most important based on expert knowledge. Additionally, we propose the divide-and-conquer approach that keeps only the variables that are critical for preserving realism or includes the variables derived based on the existing ones to help simplify the generation process. For the sake of simplicity, we illustrate an example of generating three attributes using three different configurations of conditionals: household type (Z_t), number of cars (Z_c) and set of variables $\mathcal{A} = \{X_{nl} | n \in \{1, ..., N\}\}$ where each variable indicates if the n-th person has a driving license. Although the household size (N) is not stochastically generated in GS (see Section 3.4.3), it is still a part of each conditional. Note that we implement assumptions introduced in Section 3.3 in this example. The list of experiments is shown in Table 4.

Configuration	Conditional distribution
Full	$\pi(Z_t \mid N, Z_c, \mathcal{A})$ $\pi(Z_c \mid N, Z_t, \mathcal{A})$ $\pi(X_{nl} \mid N, Z_t, Z_c, \mathcal{A} \setminus X_{nl})$
Simplified	$\pi(Z_t \mid N)$ $\pi(Z_c \mid N)$ $\pi(X_{nl} \mid N)$
Divide-and-conquer	$\pi(Z_t \mid N)$ $\pi(Z_c \mid N, Z_{td})$ $\pi(X_{nl} \mid N, Z_c)$

Table 4: The summary of conditional investigation experiments

We explain each of the previously defined terms in an example of generating the total number of cars. By using full conditionals we include all the variables so that guarantee that all relationships are included. To simplify, we could assume that the number of cars is mostly dependent on household size. Indeed, a household with more members has a higher probability to have more cars. Nevertheless, by only including the household size in the conditionals, we could generate unrealistic observations. For instance, for a household consisting only of children without driving licenses, we would draw multiple cars based solely on household

size. So the oversimplification of conditionals might cause a loss of accuracy.

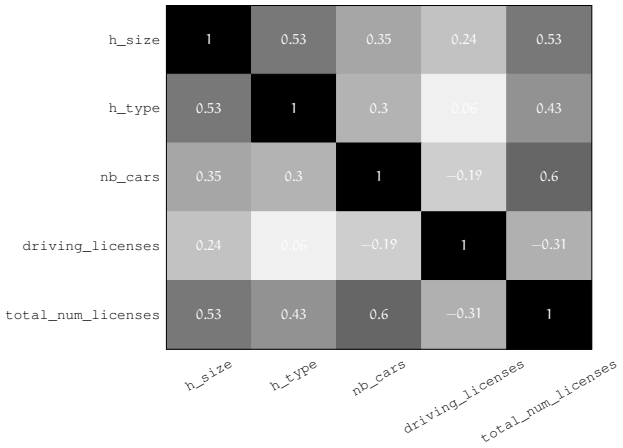

Figure 2: Correlation of household variables in real sample

Instead of including a set of variables $\mathcal{A}$ that represents if the n-th person in the household has a driving license or not, we can derive a new variable at the household level, called the total number of driving licenses Z_{td}. The value z_{td} of the variable Z_{td} can be derived as $\sum_{n=1}^{N} x_{nl}$, where x_{nl} is the value of a random variable X_{nl}. As shown in Figure 2, if we add the total number of driving licenses to the original dataset, we notice that it has a stronger correlation with the number of cars than all other attributes, which implies that represents a more informative attribute than the possession of a driving license itself. For instance, if the household is composed only of children, having zero driving licenses in households would enforce generating a smaller number of cars. Based on the results described in Section 4.2, we decide that the divide-and-conquer represents the best trade-off between accuracy and efficiency. Since generating the complete households involves $K + N \cdot L$ variables, we could not test full conditionals for each variable. However, similar to what we did for the total number of cars, we provide a modeling procedure for each variable described in Section 3.2.

3.2 Modeling conditionals for DAC one-step method

In this section, we explain how the conditional of each characteristic is constructed in the general case. The list of assumptions is presented in Section 3.3. Note that the household size is given as an input (see Section 3.4.3), and all conditionals are created using the contingency tables that are formed based on the subsets of a

specific household size N. This is equivalent to considering every variable conditional on the household size.

Household type: For the household sizes $N > 1$, let $\mathcal{A} = \{X_{na} | n \in \{1, ..., N\}\}$ be a set of random variables, where each represents the age of the n-th person in the household. Then, the value z_t of a discrete random variable Z_t is drawn from the conditional distribution $\pi(Z_t|\mathcal{A})$ for an a priori fixed realisation of the random variables defined by $\mathcal{A}$, i.e., for $X_{na} = x_{na}, \forall n \in \{1, ..., N\}$.

Number of cars: Let x_{nl} be the value of the binary variable X_{nl} that represents if the n-th person in the household has a driving license or not. For a household of size N, we can derive a value z_l of a discrete random variable Z_l denoting the total number of driving licenses as $\sum_{n=1}^{N} x_{nl}$. The value z_c of a discrete random variable Z_c is drawn from $\pi(Z_c|Z_l = z_l)$.

Age: For the household sizes $N > 1$, let $\mathcal{A} = \{X_{na} | n \in \{1, ..., N\}\}$ be a set of random variables that represent the age of the every n-th person in household and z_t the value of a discrete random variable Z_t that represents the household type. Then, the value x_{ja} of a discrete random variable $X_{ja} \in \mathcal{A}$ is drawn from the conditional distribution $\pi(X_{ja}|Z_t = z_t, \mathcal{A} \setminus \{X_{ja}\})$ for an a priori fixed realisation of the random variables defined by $\mathcal{A} \setminus \{X_{ja}\}$, i.e., for $X_{na} = x_{na}, \forall n \in \{1, ..., N\} \setminus \{j\}$.

Gender: For the household sizes $N > 1$, let $\mathcal{G} = \{X_{ng} | n \in \{1, ..., N\}\}$ be a set of random variables that represent the gender of the every n-th person in the household and z_t the value of a discrete random variable Z_t that represents the household type. Then, the value x_{jg} of a discrete random variable $X_{jg} \in \mathcal{G}$ is drawn from the conditional distribution $\pi(X_{jg}|Z_t = z_t, \mathcal{G} \setminus \{X_{jg}\})$ for an a priori fixed realisation of the random variables defined by $\mathcal{G} \setminus \{X_{jg}\}$, i.e., $X_{ng} = x_{ng}, \forall n \in \{1, ..., N\} \setminus \{j\}$.

Marital status: Let $\mathcal{A} = \{X_{na} | n \in \{1, ..., N\}\}$ be a set of random variables that represent the age of the n-th person in the household and z_t the value of a discrete random variable Z_t that represents the household type. The value x_{nm} of the random variable X_{nm} is drawn from $\pi(X_{nm}|Z_t = z_t, \mathcal{A})$ for an a priori fixed realisation of the random variables defined by $\mathcal{A}$, i.e., $X_{na} = x_{na}, \forall n \in \{1, ..., N\}$.

Employment: Let x_{na} be the value of a discrete random variable X_{na} that represents the age of the n-th person in the household. The value x_{ne} of the random variable X_{ne} is drawn from the conditional distribution $\pi(X_{ne}|X_{na} = x_{na})$.

Driving license: Let z_c be the value of a discrete variable Z_c that represents the number of cars in the household and x_{na} be the value of a discrete random variable

X_{na} that represents the age of the n-th person. The value x_{nl} of the random variable X_{nl} is drawn from the conditional distribution $\pi(X_{nl}|X_{na} = x_{na}, Z_c = z_c)$.

3.3 Assumptions

By investigating correlations before starting the generation process, we try to isolate the highly correlated areas by deterministic assignment of certain values which contributes to the improvement of the overall efficiency. The correlation identification process varies for different problems and it can be applied in different ways. We use the Pearson coefficient as shown in Figure 3.

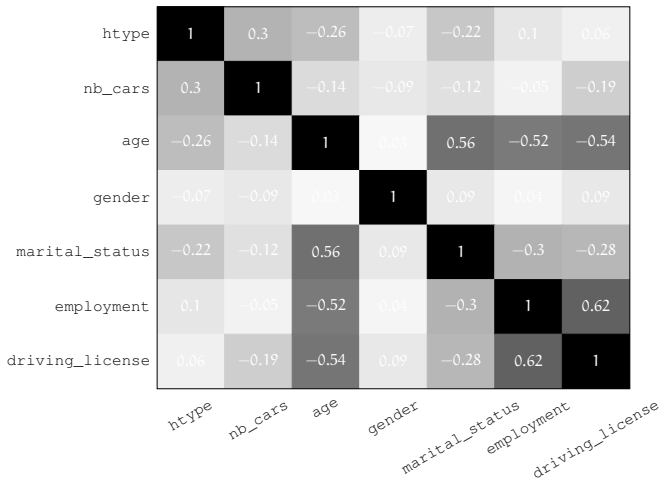

Figure 3: Correlation investigation of variables involved in Gibbs Sampler from the real sample

For correlated attributes, there might be some categories that produce most of the correlation. For example, as shown in Figure 3, there is a correlation between the age and the employment. This is expected given that people are going to school or getting retired at a specific age. Thus, if we know that the person is under 14 or above 65, we can automatically conclude that the employment status is education or, respectively, retired without taking into account any other information. On one hand, by assuming the value we sacrifice a bit of accuracy because we do not let a variable converge stochastically to the exact value (e.g. some retired people could still work). This might impact the representativity because some outliers might be omitted, but the realism is still preserved since there is no observation that does not comply with the expert knowledge constraints. On the other hand, we speed-up the algorithm since we exclude these categories from probability vectors used

for stochastic generation. The list of assumptions we make for the variables presented in Section 3.2 is as follows:

Household type: For a household of size $N = 1$, the value of z_t is set to $z_t =$ single.

Age: For a household of size $N = 1$, there is no need to capture relationships with other household members, so we can simplify the conditional and draw the value x_{ja} from $\pi(X_{ja}|Z_t = \text{single})$.

Gender: For a household of size $N = 1$, there is no need to capture relationships with other household members, so we can simplify the conditional and draw the value from $\pi(X_{jg}|Z_t = \text{single})$.

Marital status: If $x_{na} < 18$, then we set $x_{nm} =$ not married. On the other hand, for the households size $N = 2$ and z_t = couple, if the value of the variable x_{jm}, that denotes the marital status of the other household member, is married, then we also set x_{nm} = married.

Employment: If the age of the chosen person x_{na} is under 14, then x_{ne} is set to education, and if the age of the chosen person x_{na} is above 65, then x_{ne} is set to unemployed.

Driving license: If the age of the chosen person x_{na} is under 18, then x_{nl} is set to 0, which indicates that the person does not have a driving license. If $z_c > 1$, and no other person has a driving license, then x_{nl} is set to 1.

3.4 Implementation details

3.4.1 Convergence monitoring

For the simulation methods that we introduced, we simulate several chains of draws simultaneously. After a certain number of iterations, the sequences should converge to a common target joint distribution noted as $\pi(X)$. In order to assure that we draw from the common unique distribution we have to guarantee that all simulated chains mixed and reached a stationarity state with a defined number of draws. We monitor the convergence by comparing variation 'between' and 'within' simulated sequences. The variation of 'within' should be almost equal to the 'between' variation. To estimate this, we calculate two metrics: potential scale reduction factor and the effective sample size proposed by Gelman et al. (2013).

The potential scale reduction $\hat{R}$ indicates whether we should stop or continue the simulation runs, and it is calculated using the following Eq. 2:

$$\hat{R} = \sqrt{\frac{\frac{n-1}{n} \cdot W + \frac{1}{n} \cdot B}{W}} \tag{2}$$

where n is number of simulation draws, W is within-sequence variances, and B is between-sequence variance. The numerator estimates the marginal posterior variance for each chain.

Additional to $\hat{R}$, we calculate effective sample size n_{eff} to get an idea of the simulations precision using Eq. 3:

$$n_{eff} = \frac{m \cdot n}{1 + 2 \cdot \sum_{t=1}^{\infty} \rho_t} \tag{3}$$

where m is number of sequences, and ρ_t is autocorrelation of each sequence at lag t. Once we know that each simulated sequence is close to the distribution of all the sequences mixed together, we can treat the draws as a sample from the target distribution. Interested readers are referred to Gelman et al. (2013) for a complete discussion on these indicators. For each simulation method, we test efficiency by comparing the computational time needed to reach convergence.

3.4.2 Symmetry problem of Gibbs Sampler

In each iteration of the DAC one-step method, either one of the household characteristics or one characteristic of a member is picked and drawn from the corresponding conditionals that we present in Section 3.2. Compared to the two-step approach we do not label household members based on their roles and we construct the conditionals in a way that all necessary relationships between household members are captured. In order to avoid the generation of very similar households we sort individuals in decreasing order of age from the oldest to the youngest. Without age rank, two households that only differ in the order of the individuals would be regarded as different households, which could lead to a convergence problem. The ordering breaks the symmetry in the generated sample and it does not indicate the importance of the individual. Note that we discretize continuous age since we identified that there are the same trends for certain ranges of ages. The discretization of age is based on investigations of all other distributions given the age. Through this analysis, we identified age points where the life stage of a person usually changes. For example, children usually do not live alone, they are not married, they do not work, and do not have a driving license. After turning 18

people can get married and get a driving license, etc. Discretization of age helps us to reduce complexity while preserving the same accuracy.

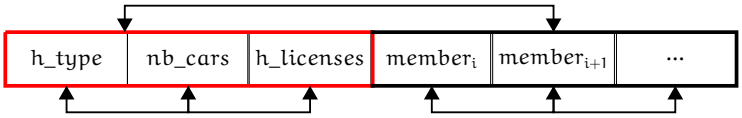

Figure 4: One-step methodology

3.4.3 Household size

Initially, we tried to involve household size in the generation process using the full conditionals. The algorithm failed to converge due to the presence of the total correlation between household size and household type for specific groups (i.e. one-member-single and two-member-couple) as shown in Table 5. Since GS is component-wise, if the algorithm picks up the illustrated vectors with total correlation it reaches a degenerative state since it always draws the same value.

	hs = 1	**hs = 2**	**hs $\geq$ 3**
Single	**1**	0	0
Couple	0	**1**	0
Couple with children	0	0	**1**

Table 5: The conditional distributions between household size (hs) and household type

Based on this analysis, we decide to exclude household size from the stochastic generation. However, since it defines the structure of the household, it must be involved in conditionals while generating other variables. Using the GS, we have to specify a desired number of generated draws. This allows us to specify the exact number of observations that we want to generate for each subset of a specific household size. The total number of draws per each subset is dictated by the marginal distribution of household size attribute in the real data sample. Following that, we divide the initial sample into subsets based on the household size and run several generation processes in parallel. Each subset contains only the observations of specified household size. At the end of the process, we merge these

subsets in order to compare the final synthetic sample with the real one. Consequently, the value of household size variable can be assigned to the dataset in the postprocessing phase. Note that household size and the total number of driving licenses are attributes in the output data sample, although they are not stochastically generated.

3.4.4 Marital status

Note that due to data unavailability, the generation of marital status does not fall under the same modeling procedure as age and gender. In other words, the best way to generate marital status would be to draw marital status conditional on the household type and marital status of other members, similarly as we do for age and gender. Instead of that, we generate marital status conditional on the household type and age of other members. As mentioned in Section 4.1, this is caused by the fact that we have information on the age and gender of each member of households, while for marital status we have information only on one household member. This is critical, since for some household types the marital status of one person depends on the marital of another. If we do not involve these interdependencies among individuals, we risk generating unrealistic observations (e.g. owner being married and the spouse being divorced). Therefore, we rely on assumptions as described in Section 3.3. By using assumptions we deterministically assign some values based on expert knowledge to make relationships among individuals realistic. This implies that we do not replicate the distribution of marital status of one individual that exists in the data, but rather the distribution of marital status of multiple individuals. Note that in Section 4.4 marital status is excluded from the comparison, since we do not have a real sample to validate our assumptions.

4 Case study

In this section, we want to show that by using our method it is possible to generate meaningful populations of households. That implies that we want to verify that we generate realistic relationships between individual and household layers. To do so, we use a methodology described in Section 2.3. We calculate SRMSE at three different aggregation levels (i.e. first, second, and third order). The first order shows a fit of marginal distributions, while the second and third order gives an insight into the replication of sub-distributions. We report one value that represents the mean and standard deviation of previously calculated SRMSE for each

combination of attributes at a specific aggregation level. The lower score indicates a better fit. All calculated values of statistics are confirmed by visualizing corresponding marginals and sub-distributions.

4.1 Data description and preprocessing

The demographic dataset used in this case study is the Swiss Mobility and Transport micro-census data (MTMC). The disaggregated sample contains information on 163,844 individuals living in 50,070 different households. In the first dataset, we have access to household attributes, age, gender, and driving license of each household member. On the other hand, in the second dataset, we have information on marital status and employment status only for one person per household. Compared to the original dataset, we delete missing values for household type, number of cars, employment, and driving license. With these changes, we lose 2.4% of data. Individuals below the legal driving age are presumed to not possess a driving license. For employment, we merge full and part-time workers into the group 'employed '. Each household is characterized by a unique identifier shared between the household members. This attribute defines a hierarchical structure between individuals and households. Since there are only 1% of households with more than 5 cars, we aggregate these categories into "more than 5" category. The list of possible values for each variable is listed in Table 3. In the following part, we describe and analyze the experiments conducted based on the processed data. All the experiments discussed in the following part are coded in Python and run on a server with Xeon(R) Gold 6140 CPU clocked at 2.30GHz and 36 processors.

4.2 The investigation of conditionals - divide-and-conquer

In this section, we discuss the influence of conditionals constructions on the efficiency and accuracy following the example presented in Section 3.1. The statistical tests illustrated in Table 6 show that the divide-and-conquer approach represents the best trade-off between accuracy and efficiency.

We see that by using full conditionals we obtain the longest computational time due to the increased complexity of adding more variables. Interestingly, the reduced complexity of simplified conditionals does not significantly affect the algorithm's convergence time. The potential reason for the slow convergence might be that we provide insufficient information with poorly constructed simplified conditionals. As expected, the divide-and-conquer approach has the lowest computational time. The main idea is to eliminate the variables that add up complexity without increasing the accuracy. This way of modeling speeds up the gen-

	First order		Second order		Third order		Computational time
	SRMSE	Difference [%]	SRMSE	Difference [%]	SRMSE	Difference [%]	Seconds
Full conditionals	4.41e-02 ± 6.19e-02	-	1.06e-01 ± 7.85e-02	-	1.84e-01 ± 7.65e-02	-	1011
Simplified conditionals	4.74e-02 ± 5.98e-02	+7.49	2.79e-01 ± 2.22e-01	+163.2	7.13e-01 ± 3.63e-01	+287.5	987 (÷1.02)
Divide-and-conquer	4.43e-02 ± 6.17e-02	+0.45	1.62e-01 ± 6.00e-02	+52.8	3.35e-01 ± 7.38e-02	+82.1	341 (÷2.97)

Table 6: Statistical tests for different conditional configurations

eration process because we decrease the number of variables, yet provide enough information to the algorithm to achieve a fast convergence to the unique joint distribution.

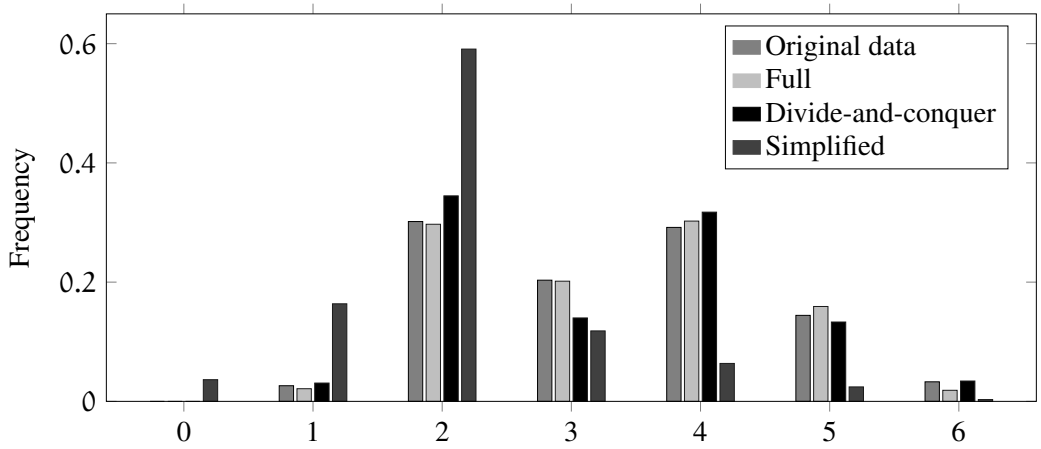

Figure 5: The sub-distribution of derived variable total number of driving licenses given that household has five cars

We observe that in all scenarios we achieve a good fit of marginals. However, we notice a more significant accuracy gap between the three scenarios, as we increase the statistical tests aggregation level. This implies that by using simplified conditionals, the sub-distributions are not well represented. If we oversimplify, we might omit some essential relationships, which leads to a lack of representativity and produces unrealistic households. In Figure 5, we illustrate the sub-distribution of the total number of driving licenses given that there are five cars in the household. We observe that considering only household size in simplified conditional is not sufficient for generating driving licenses of household members. For example, if the person does not have a driving license, she is less likely to have a

car. In Figure 5, we see that by using simplified conditionals, we obtain some categories that do not exist in the real data (e.g. household owning five cars, in which nobody has a driving license). In the case of applying full conditionals and a divide-and-conquer approach, it is impossible to generate illogical observations since we enforce some constraints (such as that the number of driving licenses cannot be bigger than the number of inhabitants). However, by using the divide-and-conquer approach we obtain the results of almost the same accuracy as using full conditionals three times faster.

4.3 Comparison of two-step and DAC one-step method

In this section, we compare two-step and DAC one-step methods. We use our implementation of a two-step method based on the description provided in Section 2.2.1. Based on the data presented in Table 3, in our experiments for the two-step method, we use only household size at the household level and age and gender at the individual level. In the DAC one-step method, we generate a full set of attributes. However, in comparisons, we focus only on comparing these three attributes. Note that household size is generated in a two-step method, while in the one-step method is given as an input parameter as explained in Section 3.4.3.

Generating individuals always in a specific order limits the two-step method since it causes the inability to generate diverse household types. For instance, if we assume that the second person is always a spouse, producing a single parent with a child is impossible. However, this household structure should be reproduced by algorithm since it exists in reality. In the one-step approach, this problem is solved by generating individuals conditional to the household type. This way all individuals are treated the same way without assigning roles, while the symmetry issue is solved by sorting individuals. As a consequence, in the two-step method the unrealistic age difference appears between partners in couples as shown in Figure 6. The age difference is expressed as a difference between age categories that are formed in pre-processing phase (Section 4.1).

In Figure 7 we show that the one-step method replicates better the age distribution of children. This is because, in the two-step method, the characteristics of the spouse and child are conditioned on all the previous individuals, whereas the characteristics of the latter members are not used to construct the conditionals to keep them simple.

Another assumption of the two-step method is that a spouse is a person with the minimum age distance from the owner among individuals of the opposite sex.

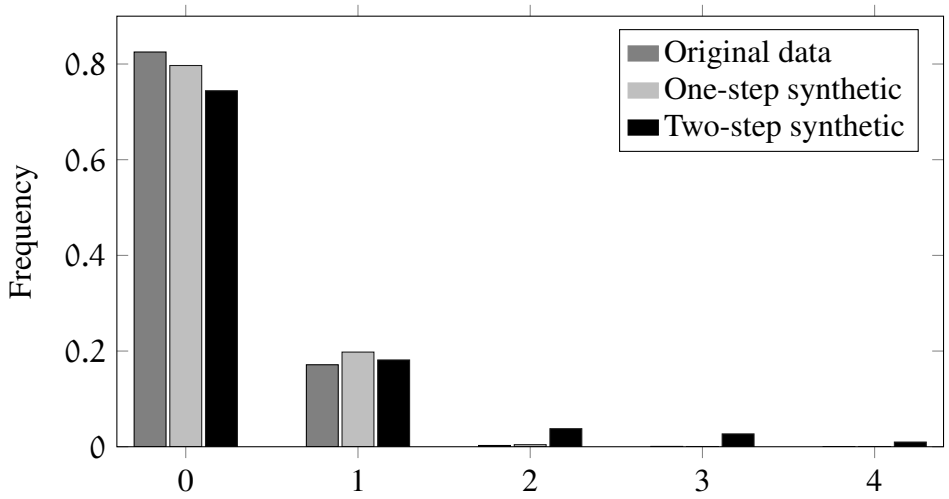

Figure 6: The distribution of age difference between the partners in household type "couple without children"

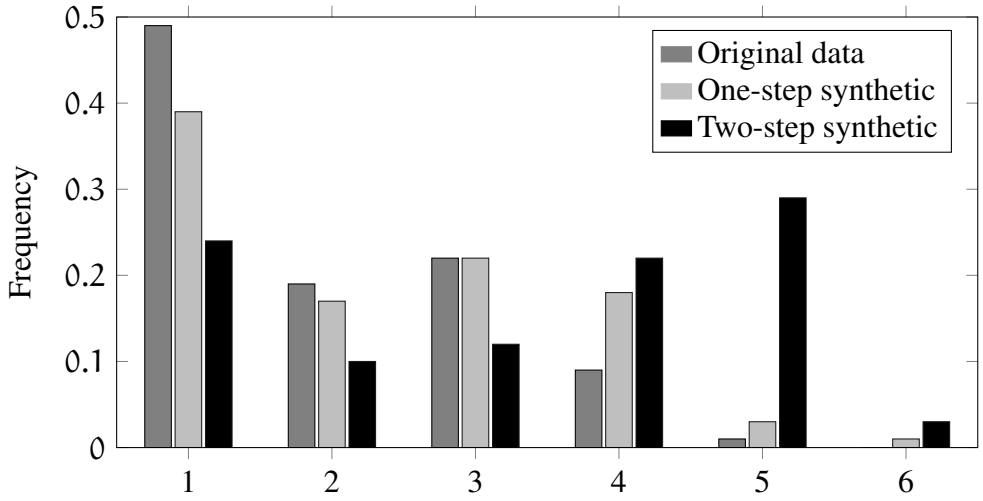

Figure 7: The marginal age distribution of children

This assumption decreases the probability of creating homosexual couples which result in under-representing or over-presenting of specific couple types. Since population trends change over time, the model should reflect the reality described by the data sample. In one-step, the gender of the spouse is drawn stochastically conditional to the household type and gender of the partner. However, as shown in Figure 8 two-step method enforces the generation of heterosexual couples more than the one-step method.

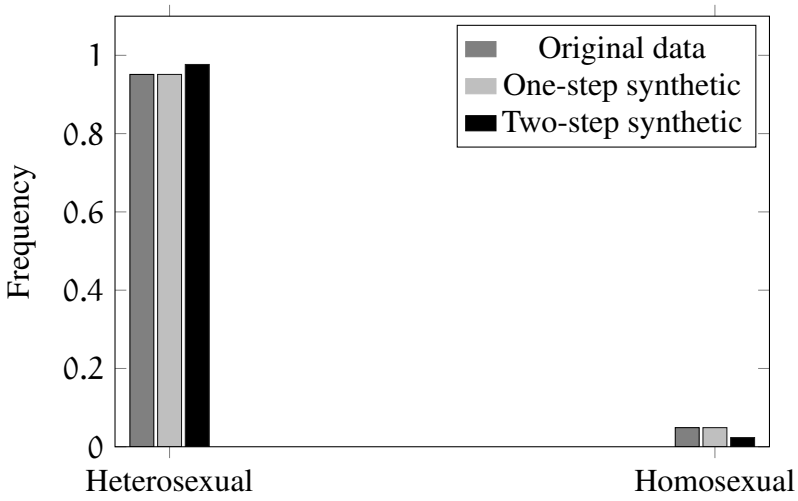

Figure 8: The representation of homosexual and heterosexual couples

In the two-step method, the individuals are not always sorted which leads to the generation of illogical observations with an unreasonable age difference. In Figure 9 we illustrate the age difference between mother and child in households labeled as "couples with one child". The negative age difference indicates that the two-step method might produce observations where the child is older than the parent. This behavior is a consequence of the imposed assumption that the spouse is the opposite gender compared to the owner. We illustrate one example of this phenomenon in Table 7. Suppose that we can find a homosexual male couple that adopted a daughter and son in the real dataset. Since the spouse is assumed to be an opposite gender compared to the owner, the two-step method labels the daughter as a spouse while the second partner is selected as a child. This is evidence that the role assignment to the household structures limits the method to be able to reproduce the new patterns that might appear in data over time (such as homosexual couples with adopted children).

Using a two-step method, in each iteration, the complete sequence of the individuals is generated which makes the method computationally heavy. Moreover, by enforcing the order in the generation process, we do not follow the definition of a traditional GS where the algorithm always randomly picks one dimension to draw. However, treating the whole sequence of individuals as one dimension in a two-step GS is necessary to prevent the generation of unrealistic observations. This phenomenon is explained through the example shown in Figure 10.

In this example, we are trying to generate the age attribute of the household

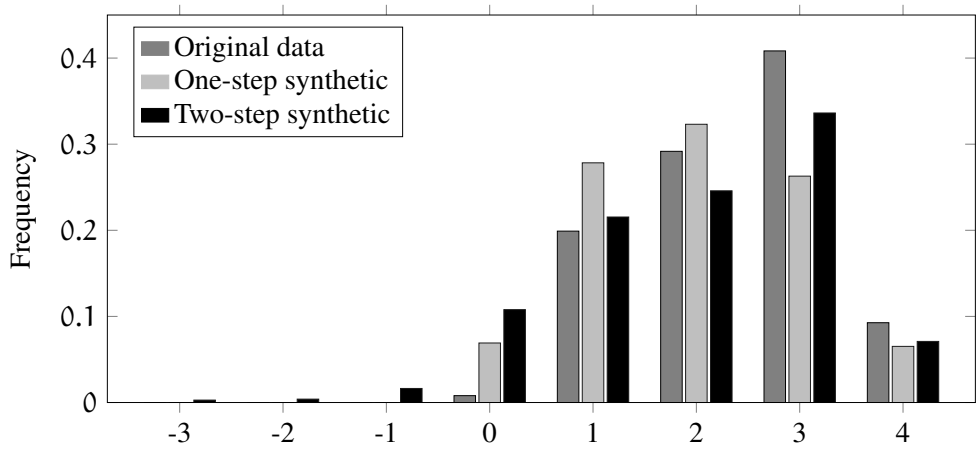

Figure 9: The distribution of age difference between a spouse and a child in "couples with one child" type of household

Age	Gender	Role
$\geq$ 65	male	owner
[35, 45]	male	spouse
[18, 25]	female	child
$\leq$ 18	male	child

Age	Gender	Role
$\geq$ 65	male	owner
[18, 25]	female	spouse
[35, 45]	male	child
$\leq$ 18	male	child

Table 7: The example of illogical synthetic household (right) generated using referent real sample (left)

composed of two parents and one child. Following the common practice of the GS, we should randomly pick and draw a value for one dimension in each iteration. In this case, we assume to draw the age of the second person. If we only use the information of the previous individual, we could draw the person who is much younger than expected. This would result in households of one parent and two children instead of two parents and one child. To overcome this issue, in the two-step, instead of picking the dimension randomly, they generate the whole sequence in one iteration. They guarantee that first, the owner's age is generated, then the spouse's age is drawn based on the owner's age, and finally, the children's age is drawn based on the spouse's age. In the one-step approach, we treat each individual equally, with the same importance, without assigning roles. Each iteration generates one individual by considering the relationships with all other

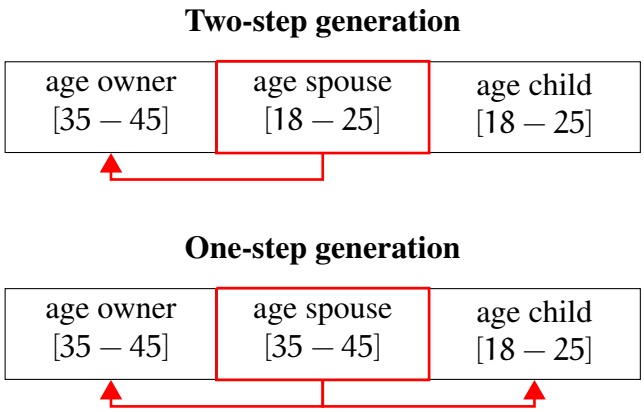

Figure 10: Comparison of differences between one-step and two-step approach

individuals. In the case of generating the age of the second person, the age difference between all the individuals is considered. By capturing all relations between individuals, we do not have a problem with the generation of unrealistic observations. The only requirement of the one-step approach is that the sequence of the individuals is always sorted. The sorting is necessary to prevent the symmetry issue of GS, which manifests through the generation of too similar households.

Treating all the individuals with the same importance makes the one-step approach more flexible because we can generate any type of household. Also, the generation process in a one-step procedure is faster than the two-step, because instead of creating the complete sequence of individuals per iteration, we only generate one individual per iteration. With the one-step method, we achieve better accuracy according to statistics while we reduce the computational time by half as shown in Table 8. Note that for the third value, we have only one value since we test only the combination of three attributes (i.e., household size, age, and gender).

	First order	Second order	Third order	Computational time
DAC one-step	**3.77e-02 ± 3.39e-01**	**2.45e-01 ± 2.46e-01**	**0.627**	**2.5h**
Two-step	3.50e-01 ± 2.93e-01	6.21e-01 ± 2.26e-01	0.836	5h

Table 8: Statistical tests between DAC one-step and two-step method

4.4 Comparison of DAC one-step method with state-of-the-art for the synthetic data generation

We also compare our approach with a state-of-the-art ML method Lederrey et al., 2022 which is publicly available on Github (`https://github.com/glederrey/SynthPop`). DATGAN does not generate hierarchical structures such as associated individuals and households. However, it can generate one person per household with a full set of attributes. In order to make these DATGAN comparable with our method, we randomly pick one individual per household from the synthetic dataset generated by the simulation method.

Based on the gaps between calculated statistics presented in Table 9 DATGAN generates a more representative sample than our method. To investigate results further, we analyze the marginal distribution of each attribute separately in Table 10.

	First order	Second order	Third order
DAC one-step	6.36e-02 ± 3.45e-02	1.58e-01 ± 5.91e-02	2.96e-01 ± 8.60e-02
DATGAN	**1.21e-02 ± 6.32e-03**	**3.66e-02 ± 1.23e-02**	**8.44e-02 ± 2.29e-02**

Table 9: Statistical tests between DAC one-step and DATGAN

	DAC one-step	DATGAN
Housheold type	0.04	0.002
Number of cars	0.05	0.01
Age	0.1	0.01
Gender	0.008	0.01
Employment	0.09	0.01
Driving license	0.07	0.002

Table 10: First order SRMSE per attribute

In order to identify what causes discrepancies between the calculated statistics of the two methods, we analyze in detail second and third-order sub-distributions. Second-order analysis implies that we test a marginal distribution of one attribute given the value of another one. On the other hand, third-order analysis means that we perform a second-order test on the subset in which we select only attributes with the specific attribute value. We choose sub-distributions based on the expert

knowledge rule we want to test. An example of second and third-order visualization is shown in Figure 11 and Figure 12, respectively.

In Figure 11 we show the relationship of age with other variables. On the left, we show the age distribution of workers. We can see that DATGAN generates children who are classified as workers, which does not appear in reality. The employment for this age category should be education. On the right, we show the distribution of marital status of people younger than fourteen. In these examples, we see that DATGAN generates some observations other than single, which also does not correspond to reality.

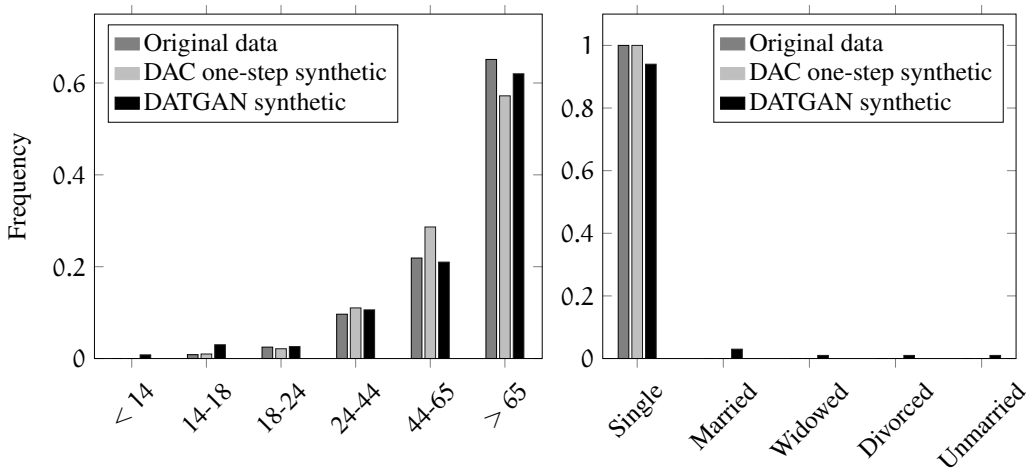

Figure 11: The second order sub-distribution comparison - (left) age distribution given the employment = employed, (right) marital distribution of people younger than fourteen

In Figure 12 we select only single households. On this subset, we analyze the sub-distribution of the number of cars of people without driving licenses. Since these people live alone in households, there is no possibility that someone else in the household has a driving license to drive a car. We would expect that it is less likely that people without a driving license possess a car. However, with DATGAN we generate more people without driving licenses having cars than exist in real data. We see that although DATGAN captures better the overall trend in data, it can generate illogical observations, while DAC one-step simulation does not generate any illogical observations. This is expected since the main advantage of our approach is that by designing the conditional distributions properly, we can enforce the generation of realistic households according to the expert knowledge

rules. One can argue that number of illogical observations generated by DATGAN is not significant, hence it could be removed in the postprocessing. In that case, the difficulty would be to identify all of the rules that should be tested. In the simulation method, all of these rules are embedded within the algorithm.

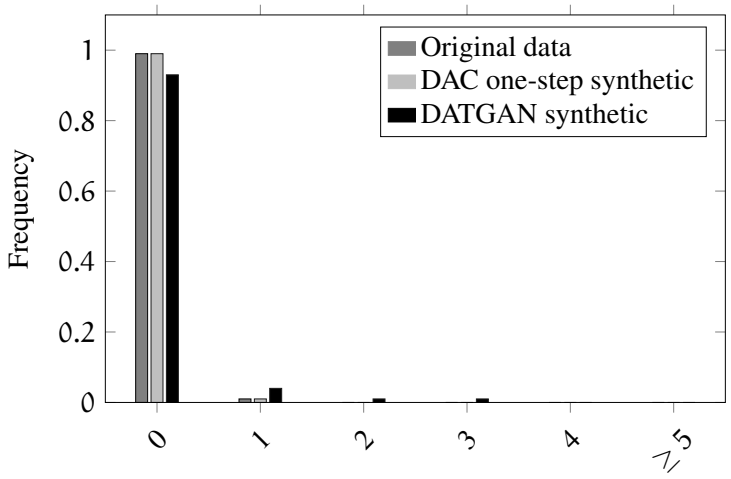

Figure 12: The third order sub-distribution comparison - the distribution of total number of cars for household type = single and driving license = no

As a result of this study, we can conclude that DATGAN cannot entirely control the generation process, since it generates some illogical observations. That means that with DATGAN we obtain a more representative sample, but we cannot guarantee realism and consistency of it. On the other hand, simulation guarantees realism and consistency, although some modeling assumptions may have an impact on representativity. It is also worth mentioning that the relevance of the representativity comparison between simulation and ML is debatable, given that the simulation method potentially could achieve the same accuracy with a sufficiently large number of draws. Also, the idea of the DAC one-step method is to generate hierarchies, meaning that normally it captures more relationships than DATGAN. The random choice of one individual per household in the sample generated sample might influence the results.

5 Conclusion

This paper introduces a new simulation-based approach for generating synthetic households. The approach utilizes modeling strategies to enhance the competitiveness of the simulation techniques in the big data era. By uniting the hierarchical generation process into a one-step procedure, this method offers greater flexibility and generality compared to other simulation population generation methods. The flexibility comes from the fact that the DAC one-step method generates the age attribute in decreasing order eliminating the need for labeling the household members based on their roles. Additionally, it incorporates household type as a variable which allows for generating diverse household structures (e.g., single parent with a child).

The obtained results show that the DAC one-step generates more representative samples than the two-step method more efficiently. The efficiency improvement stems from not requiring two separate GS to generate households. Moreover, we show that by designing the conditionals precisely, we can find a trade-off between accuracy and efficiency, which allows us to include more relationships and attributes in the generation process. The enhanced accuracy may steam from considering age and gender dependencies among all individuals, which ensures a better representation of multivariate distributions. Furthermore, by proposing the decomposition of the generation process based on the household size, we address the scalability issue of the simulation-based method. The results indicate that using model-based methods is superior to data-driven approaches in controlling the generation process. Although ML techniques better capture the correlations, they still might produce illogical observations.

The future direction is to investigate the possibilities of using the DAC one-step method to create additional hierarchical structures such as the individuals and their activity sequence. Research in the field of synthetic generation focuses on the generation of individuals and their sociodemographic characteristics, but there are fewer contributions when it comes to the generation to attributes related to the activities they perform. This lack of information hinders the use of synthetic populations in activity-travel applications, as it means that further assumptions must be taken to link activities to individuals. The DAC one-step method is a promising approach for an integrated multi-hierarchical framework (e.g. synthetic households with their activity sequence) for the full population generation.

References

Abraham, J. E., Stefan, K. J. and Hunt, J. D. (2012). Population synthesis using combinatorial optimization at multiple levels.

Arentze, T., Hofman, F. and Timmermans, H. (2007). Creating synthetic household populations: Problems and approach, *Transportation Research Record* p. 6.

Auld, J. and Mohammadian, A. (2010). Efficient methodology for generating synthetic populations with multiple control levels, *Transportation Research Record* **2175**(1): 138–147.
URL: *https://doi.org/10.3141/2175-16*

Axhausen, K. W. (2000). Activity-based modelling: Research directions and possibilities.

Badu-Marfo, G., Farooq, B. and Paterson, Z. (2020). Composite travel generative adversarial networks for tabular and sequential population synthesis.
URL: *https://arxiv.org/abs/2004.06838*

Barthelemy, J. and Toint, P. L. (2013). Synthetic population generation without a sample, *Transportation Science* **47**(2): 266–279.

Beckman, R. J., Baggerly, K. A. and McKay, M. D. (1996). Creating synthetic baseline populations, *Transportation Research Part A: Policy and Practice* **30**(6): 415–429.

Ben-Akiva, M. and Lerman, S. (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*.

Casati, D., Müller, K., Fourie, P. J., Erath, A. and Axhausen, K. W. (2015). Synthetic population generation by combining a hierarchical, simulation-based approach with reweighting by generalized ranking, *Transportation Research Record* **2493**(1): 107–116.
URL: *https://doi.org/10.3141/2493-12*

Castiglione, J., Bradley, M. and Gliebe, J. (2014). *Activity-Based Travel Demand Models: A Primer*, The National Academies Press, Washington, DC.

Choupani, A.-A. and Mamdoohi, A. R. (2016). Population synthesis using iterative proportional fitting (ipf): A review and future research, *Transportation Research Procedia* **17**: 223–233. International Conference on Transportation Planning and Implementation Methodologies for Developing Countries

(12th TPMDC) Selected Proceedings, IIT Bombay, Mumbai, India, 10-12 December 2014.

Farooq, B., Bierlaire, M., Hurtubia, R. and Flötteröd, G. (2013). Simulation based population synthesis, *Transportation Research Part B: Methodological* **58**.

Garrido, S., Borysov, S. S., Pereira, F. C. and Rich, J. (2019). Prediction of rare feature combinations in population synthesis: Application of deep generative modelling.
URL: *https://arxiv.org/abs/1909.07689*

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A. and Rubin, D. (2013). *Bayesian Data Analysis (3rd ed.)*, A Chapman and Hall Book,CRC Press, London.
URL: *https://doi.org/10.1201/b16018*

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial networks.

Guo, B. (2007). Population synthesis for microsimulating travel behavior, *Transportation Research Record* p. 9.

Lederrey, G., Hillel, T. and Bierlaire, M. (2022). Datgan: Integrating expert knowledge into deep learning for synthetic tabular data.
URL: *https://arxiv.org/abs/2203.03489*

Lenormand, M. and Deffuant, G. (2013). Generating a Synthetic Population of Individuals in Households: Sample-Free Vs Sample-Based Methods, *Journal of Artificial Societies and Social Simulation* **16**(4): 12.
URL: *http://jasss.soc.surrey.ac.uk/16/4/12.html*

Miranda, D. F. (2019). Reviewing synthetic population generation for transportation models over the decades.

Müller, K. and Axhausen, K. (2011). Population synthesis for microsimulation: State of the art, *in* TRB (ed.), *90th Annual Meeting of the Transportation Research Board*.

Olde Kalter, M.-J. and Geurs, K. (2016). Exploring the impact of household interactions on car use for home-based tours: a multilevel analysis of mode choice using data from the first two waves of the netherlands mobility panel, *European Journal of Transport and Infrastructure Research* **16**: 698–712.

Pougala, J., Hillel, T. and Bierlaire, M. (2022). Capturing trade-offs between daily scheduling choices, *Journal of Choice Modelling* **43**: 100354.

Pritchard and Miller (2012). Advances in population synthesis: fitting many attributes per agent and fitting to household and person margins simultaneously., *Transportation 39* p. 685–704.

Saadi, I., Mustafa, A., Teller, J., Farooq, B. and Cools, M. (2016). Hidden Markov Model-based population synthesis, *Transportation Research Part B: Methodological* **90**: 1–21.

Templ, M., Meindl, B., Kowarik, A. and Dupriez, O. (2017). Simulation of synthetic complex data: The r package simpop, *Journal of Statistical Software* **79**(10): 1–38.

Xu, L. and Veeramachaneni, K. (2018). Synthesizing Tabular Data using Generative Adversarial Networks, *arXiv:1811.11264 [cs, stat]* . arXiv: 1811.11264.
URL: *http://arxiv.org/abs/1811.11264*

Yaméogo, B. F., Gastineau, P., Hankach, P. and Vandanjon, P.-O. (2021). Comparing methods for generating a two-layered synthetic population, *Transportation Research Record* **2675**(1): 136–147.
URL: *https://doi.org/10.1177/0361198120964734*

Ye, X., Konduri, K., Pendyala, R., Sana, B. and Waddell, P. (2009). Methodology to match distributions of both household and person attributes in generation of synthetic populations.

Zhu, Y. and Ferreira, J. (2014). Synthetic Population Generation at Disaggregated Spatial Scales for Land Use and Transportation Microsimulation, *Transportation Research Record: Journal of the Transportation Research Board* **2429**(1): 168–177.
URL: *http://journals.sagepub.com/doi/10.3141/2429-18*