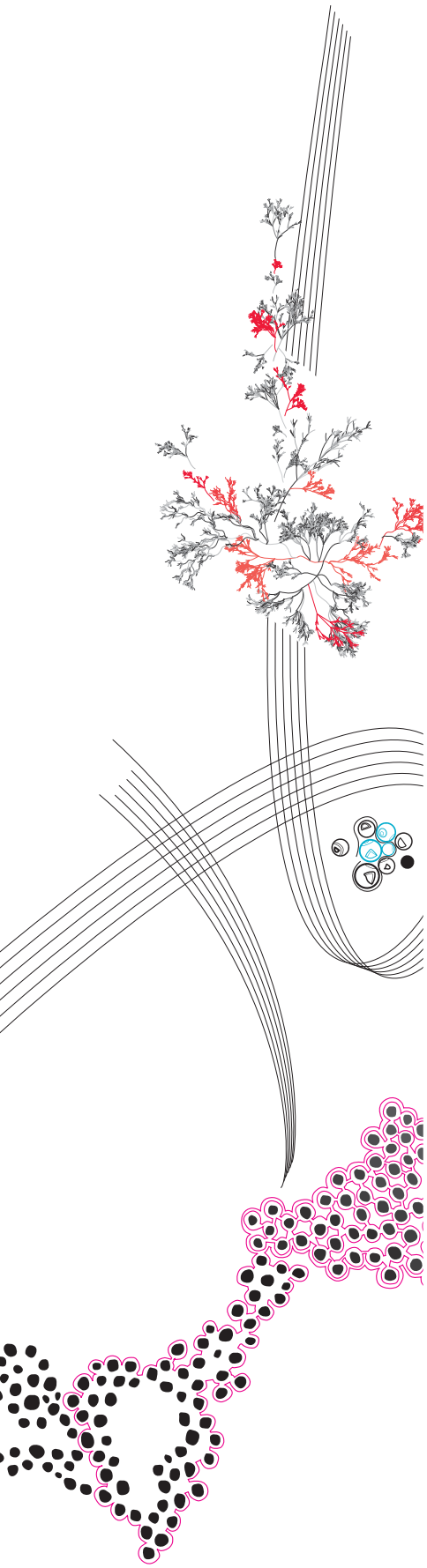


Master Thesis Applied Mathematics

Maintaining household composition in synthetic populations for microscopic travel demand models

Aline de Jong

January, 2025

Stochastic Operations Research
Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

Supervisors:
J.C.W. van Ommeren (University of Twente)
W.R.W. Scheinhardt (University of Twente)
L.J.N. Brederode (Dat.mobility)
P.W. Klein Kranenbarg (Dat.mobility)

UNIVERSITY OF TWENTE.

Preface

With this thesis I complete my master degree in Applied Mathematics at the University of Twente. During this research I had the opportunity to apply my mathematical knowledge to a practical problem,

First of all, I want to thank Luuk and Peter for giving me the opportunity to do my thesis at Dat.mobility, and for their guidance during this process. Thank you for your insights and ideas, which helped me during my thesis. Together with the other people at Dat.mobility, you made it a pleasure to be in the office in Deventer. I would also like to thank my supervisors from the University of Twente, Jan-Kees and Werner. Thank you for our nice discussions in which I sometime had to explain concepts a couple of times, as Werner joined the supervision later in the process.

Furthermore, I want to thank everyone that has made my time at the University of Twente a great time. From playing games with my mathematics friends, to all that I experienced at D.S.V. de Skeuvel. All the fun trips and activities I will remember. Over the years, I became more fanatical with marathon skating and that included the cosy hours in the car with fellow Skeuvelaars. I will cherish there moments for a very long time. Also a big thank you to my family and boyfriend for their support and encouragement during my studies, and always having you by my side.

Aline de Jong
Enschede, January 2025

Abstract

In microscopic travel demand models, all residents are modelled as individuals that each make their own choices based on their characteristics and circumstances. Modelling individual behaviour offers detailed insights into mobility patterns and the effects of policies on specific target groups. The travel demand modelling begins with the generation of a synthetic population, representing either a reference population or a future scenario. The current population synthesizer used by Dat.mobility and Goudappel has its limitations. During the last step, household compositions are not preserved, leading to inconsistencies, where characteristics of individuals do not reliably represent complete households. To ensure the new population synthesizer addresses these limitations, and generates consistent households, the population synthesis problem is formulated as a mixed-integer linear program that enforces both household consistency and compliance with marginal totals. Additional requirements are incorporated in extensions of the mixed-integer linear program. The proposed formulations are evaluated on a full-scale travel demand model using the open source solver HiGHS.

Contents

1 Introduction

Goudappel is a Dutch consultancy firm in mobility planning. Together with Dat.mobility (part of Goudappel Group), they develop methods to collect and analyse mobility data. Dat.mobility converts data into information that leads to insight and knowledge. To be able to advise governments and private parties with developing and testing their mobility policy, Dat.mobility developed a microscopic tour-based travel demand modelling framework, called Octavius [1]. This provides reliable insights into the effects of mobility policy on different target groups.

Octavius differs from classical trip- and tour-based models. In trip-based models, such as the gravity model [2], separate trips are modelled and all inhabitants have the same average behaviour (aggregated population). Tour-based models go a step further by generating consecutive trips that form a tour. The inhabitants do not all have the same average behaviour, but they are split in subgroups, which each has its own average behaviour (disaggregated population). Also with Octavius, tours are generated, but the inhabitants are now individually modelled and make choices based on their characteristics (microscopic population). Microscopic travel demand modelling focuses on simulating individual travel behaviours rather than aggregated averages. The relative position of Octavius in comparison to other models is illustrated in Figure 1.1.

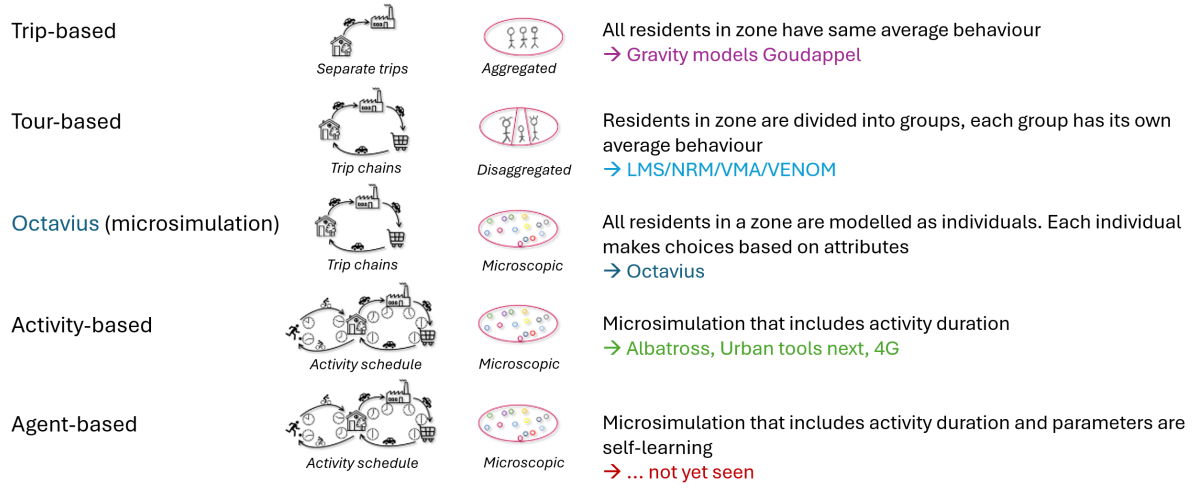


FIGURE 1.1: *Where does Octavius stand?*

Besides Octavius, Goudappel also uses the gravity model and has been using it for a long time. They still use their gravity model, because until about three years ago, it was their only travel demand model. Also, since Octavius only exists for a couple of years, it is at present set up for a few cities/study areas. Furthermore, gravity models can give a good approximation when modelling traffic and the effect on for example emissions, or noise levels.

Octavius is developed because mobility varies greatly by target group. Young people travel differently than seniors, and two working parents travel differently than a single person in their thirties. Effective mobility policies should take these differences into account as much as possible. It is most important that traffic models can more accurately calculate the effects of policy measures, by taking the characteristics of individuals into account. On top of that, they should be able to measure and visualise the effects for different target groups. In addition,

we see on the streets how our behaviour is increasingly influenced by new mobility forms and concepts such as shared scooters and e-bikes, MaaS (Mobility as a Service) and hubs, and eventually self-driving cars. A future-proof traffic model must be prepared to include their effects in calculations.

Microscopic travel demand modelling with Octavius has various advantages. With Octavius, it is possible to test the effect of policy measures on specific target groups. The modelling takes place at the individual level instead of using average behaviour, therefore it provides a detailed insight into the mobility behaviour of different target groups and the effect of a policy on them. In Octavius, both household and personal characteristics are taken into account. Furthermore, microscopic travel demand modelling with Octavius gives a better simulation of reality, as it employs a tour-based approach. Therefore, the complete daily travel behaviour of individuals is used instead of individual trips without coherence.

The current choice models implemented in Octavius derive daily activity patterns for all persons and households within a predefined study area using the four stages as shown in Figure 1.2. In the first stage, the population synthesizer generates a synthetic population. A population synthesizer creates a virtual representation of a population by combining individual and household attributes in a statistically valid manner. In the second stage, tours are generated for each agent (an individual in the synthetic population). A tour is defined as a round trip that starts and ends at the home of a synthetic person. During a tour, the person does one or more activities, such as working or shopping. For each synthetic person, the tour generator produces the number of tours and the number, type, and order of the activities within each tour. An example of a tour is home $\rightarrow$ work $\rightarrow$ store $\rightarrow$ home. This tour consists of three trips: home $\rightarrow$ work, work $\rightarrow$ store, and store $\rightarrow$ home. The third stage is the destination choice for each activity in the tour. For example, where a synthetic person works and where he/she shops on an average day. Finally, in the fourth stage, the modes are selected for all trips in all tours. A different mode can be selected for each trip in the tour, constrained by a set of modes that are interchangeable. A person may use public transport to get from home to work but decide to walk from work to the store, and then travel by public transport again to go from the store to their home.

FIGURE 1.2: *Schematic overview of the four stages in the current travel demand modelling framework in Octavius. The blue boxes represent the population synthesizer and the choice models, whereas the in- and outputs to the stages are represented as blue text without background.*

Being a micro-simulator, the starting point for any application of Octavius is deriving a synthetic population, representing either a reference population of the study area, or some scenario. A reference population represents the current population that aligns with the known marginal totals. In a scenario, these marginal totals are adjusted to observe how the population changesfor instance, in response to an increase in the number of individuals or households.

For privacy reasons, there are no complete datasets on the socio-demographic characteristics of individuals on a small geographic scale that can be used directly, since this is not publicly

available and may not be used in such form. Therefore, a population synthesizer is necessary to generate a synthetic population representative of the actual population. The population synthesizer thus lies at the foundation of the microscopic travel demand modelling framework Octavius. A population synthesizer also allows us to do forecasting and what-if scenarios on the population. This would not be possible with only an observed population.

1.1 Problem description

The primary objective of the population synthesizer is to create a stable synthetic population for each zone while ensuring consistency with both individual and household marginal totals on attributes. A zone is a predefined area for which the marginal totals are known, specifically, the number of individuals and households with certain attributes.

The population synthesizer in Octavius combines personal information (e.g. gender and age) and household information (e.g. number of people and cars in the household), see Figure 1.3. Before we discuss the population synthesizer and its limitations in more detail, we explain the relation between attribute categories, attributes, and segments. The attribute categories are listed in Figure 1.3, for example "gender" is an attribute category. Under note [1] we find the attribute categories on the personal level, and under note [2] we find the attribute categories on the household level. Here, we also find the possible values for each attribute category. The combination of an attribute category and a corresponding value is called an attribute, for example, "gender = female" is an attribute. Finally, a segment is a combination of attributes, for example [social participation = student, age = 18 – 29, gender = female, driver's license = yes, ethnicity = Dutch] is a segment.

The first step of the current population synthesizer is determining the number of synthetic inhabitants and households in each segment per zone using Iterative Proportional Fitting (IPF). IPF finds the most likely distribution of inhabitants and households across segments, given an observed sample distribution, that satisfies the zonal marginal totals.

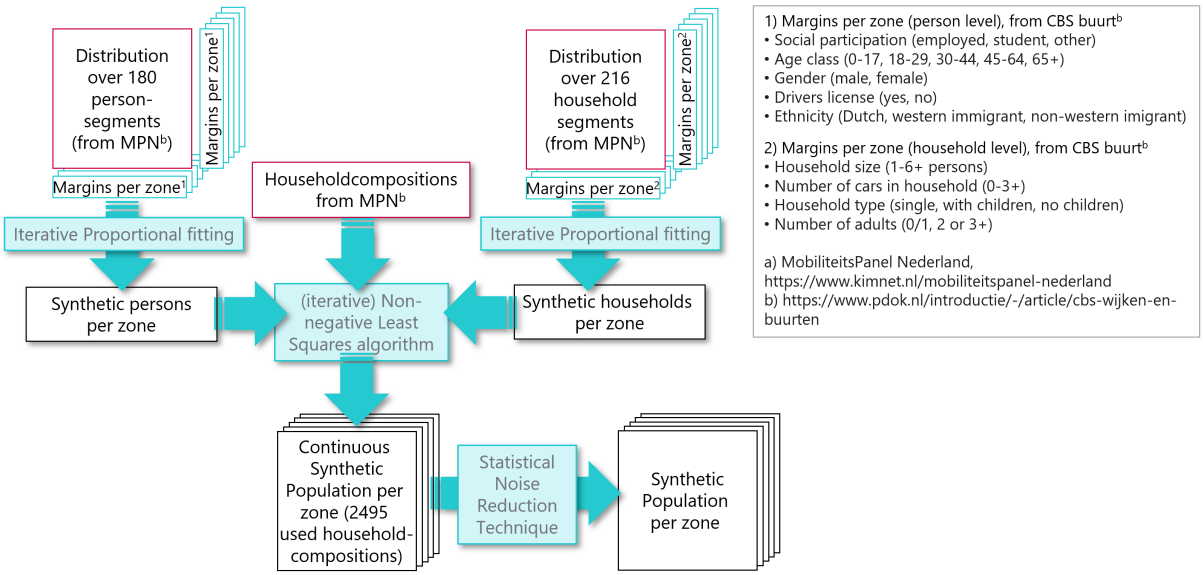


FIGURE 1.3: *Schematic overview of the current population synthesizer in Octavius. Details are discussed in Section 2.2.*

Furthermore, a representative sample of Dutch household compositions is known from the mobility panel Netherlands (MPN). The household composition data defines the set of valid combinations of personal and household segments. With the iterative Non-Negative Least Squares (iNNLS) algorithm, the households in the MPN data are weighted such that the resulting continuous synthetic population corresponds to the personal and household segment totals from the two IPF results.

The continuous population is integerised using the Statistical Noise Elimination Technique (SNET) [3]. In this step of discretising the continuous population, the household compositions cannot be maintained using SNET. Based on the continuous household weights, SNET selects individuals from households that have the highest value until the exact number of inhabitants in a zone is reached. In this integerisation process, it sacrifices household consistency. Resulting problems are, for example, that in a zone there could be only one person that is in a three-person household, but no others in a three-person household; or two persons with a driving licence can both use one car at the same time, while in reality they would have to share the car.

As an individual's decision depends on both the personal characteristics and the household situation, it is important to generate synthetic populations that take into account both personal and household information [4, 5]. Personal and household information are both used for choice modelling in Octavius. With the current population synthesizer there is a consistency issue in the fact that we do not have complete households. The household information should not only be seen as attributes of a person, but it is also important to know which persons belong together in a household and that these households are consistent.

The inter-dependencies among households remain present in the continuous synthetic population obtained after iNNLS. However, the inter-dependencies are not preserved in the final (integerised) synthetic population using SNET in the current population synthesizer. Generating a two-layered population (i.e. linking persons and consistent households) remains one of the most challenging problems in population synthesis [6].

In addition to aligning with the marginal totals, it is important for the synthetic population to remain stable. Stability ensures that the model produces the same solution each time it is run with the same input. Achieving this requires a specific condition. The most reliable approach is to introduce a condition that guarantees a unique solution. A condition with behavioural significance, such as identifying the most likely solution, is particularly appropriate. This can be achieved by minimising relative entropy. Furthermore, we want the solution to remain robust to small changes, avoiding drastic shifts in outcomes. Identifying the most likely solution helps address these requirements effectively.

1.2 Goal and outline of the report

The goal of this thesis is to develop a population synthesis method that keeps household composition intact and matches both personal- and household-level marginal totals. This means generating a synthetic population where individuals are linked to the same household if they come from the same household, and the attributes of both people and households add up to the known marginal totals. Based on this objective, the following research question can be formulated:

How can a discrete synthetic population be generated such that it is consistent with per-

sonal and household marginal totals, contains information about household composition, and is the most likely synthetic population?

To generate a reference population, it is desired to obtain a unique and most likely synthetic population (unicity). For the scenarios, the resulting synthetic population must be stable in order to be able to make a good comparison with the reference population (stability).

This thesis focuses on the use of a synthetic population in the microscopic travel demand models implemented in Octavius. However, synthetic populations are also used in models with other fields of application such as for the spread of infectious diseases [7], but also in urban planning and public health modelling in general [8]. Therefore, generalisations of the methods developed in this thesis may also be used in population synthesizers with other applications. The various uses for synthetic populations have different attributes of interest [7]. Therefore, the construction of the population synthesizer may vary across fields of application.

The remainder of this report is structured as follows. In Section 2, literature on population synthesizers and integerisation algorithms is discussed, as well as the details of the current population synthesizer used in Octavius. In Section 3, we present the mathematical problem formulation, that meets the main requirements, and propose some extensions to incorporate additional requirements. In Section 4, results of the implemented methods are presented and analysed. In Section 5, limitations and possibilities for further research are discussed. Also, in this section, concluding and highlights of the research are presented.

2 Background

This section contains a literature search on population synthesis techniques (Section 2.1), and a detailed description of the population synthesizer currently used in Octavius (Section 2.2). We conclude this section with an overview of all methods mentioned in the previous subsections (Section 2.3).

2.1 Literature research

Synthetic populations are a simplified representation of the actual population as only those variables of interest are to be reproduced [9]. The primary objective of population synthesis can be summarised as generating an individual dataset in full compliance with the statistical characteristics of various input data [10]. For synthetic populations in microscopic travel demand models, we are looking for a discrete synthetic population, as opposed to a continuous synthetic population, that contains consistent households with individuals that each have their own set of characteristics.

There are three different categories of approaches in generating synthetic populations [11], namely synthetic reconstruction (SR), combinatorial optimisation (CO), and statistical learning (SL). SR and CO methods produce synthetic populations by means of replicating individuals, whereas SL methods draw a population following a joint probability estimation. SR methods are deterministic, whereas CO an SL are both stochastic. [4]

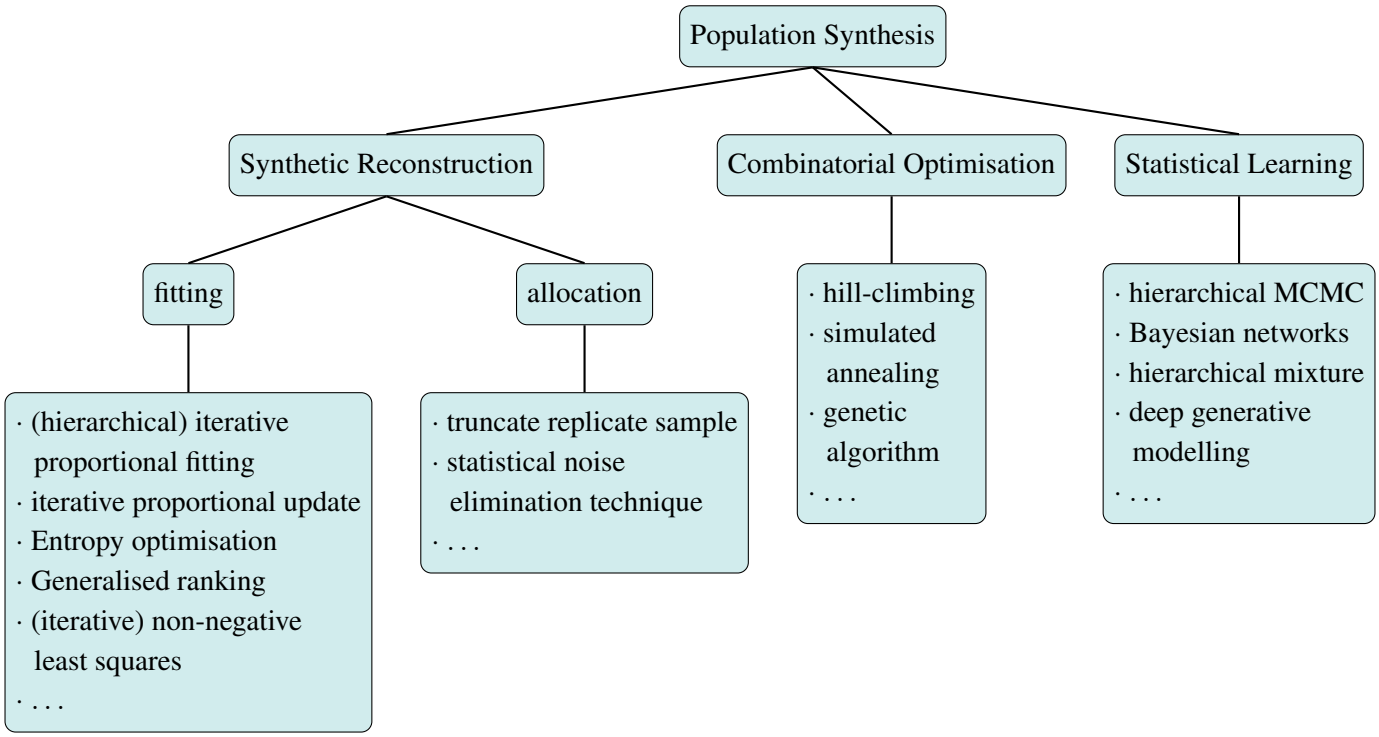


FIGURE 2.1: *Schematic overview of population synthesis methods.*

2.1.1 Synthetic reconstruction (SR)

SR methods follow a two-step procedure: fitting and allocation. The fitting step assigns positive weights to the individuals and households. In the allocation step, these non-integer weights are converted to integer weights, and individuals are replicated in proportion to their weights [**?**].

Iterative Proportional Fitting (IPF) is the most widely used algorithm for the fitting step within SR to generate a synthetic population. For IPF it is important to have an initial, representative sample of the true population at hand [12] since the result depends on the input sample. The advantages of IPF are that it is simple, accurate, fast, and it finds the most likely population. That is, the population adhering to the minimum relative entropy. However, it is not able to generate synthetic populations linking persons with households [12, 4, 5]. It is a clear shortcoming when a method is not able to generate a two-layered synthetic population, as an individual's decision depends on both his/her personal characteristics and household situation [13, 14, 6]. Therefore, it is important to have the individuals linked to households.

Another shortcoming of IPF (and also other SR methods) is the zero-cell problem [4]. It occurs when one or more cells in the contingency table have a zero value, indicating that a specific combination of attributes (a segment) does not exist in the sample population that gives the distribution over the segments (see Figure 1.3). If a cell starts with a zero, no matter how much adjustment is made, the cell remains zero. A zero-cell implies that the corresponding segment will not be represented in the synthetic population, even if the real population could have a small but non-zero count for that combination due to sampling or rounding errors.

Besides IPF, there are more methods that fall under the SR category, also ones that can generate a two-layered population (e.g. persons and households). Yamogo et al. [4] name Iterative Proportional Update (IPU) and Hierarchical Iterative Proportional Fitting (HIPF) as methods that simultaneously estimate both household and individual-level attributes. They respectively compute the factors and weights linked to individual and household records using an iterative approach with a cross-categorisation of individual types into household types. HIPF constantly switches between the household and person domain. Whereas in IPU, weights are adjusted to satisfy household level constraints first and then updated to satisfy person-level constraints.

When the population synthesis problem is formulated as a constrained optimisation problem, it can be solved using one of two approaches: entropy-optimisation (ENT) and generalised ranking (GR). Entropy-optimisation directly optimises an information-based similarity metric of the weights and therefore introduces the least possible amount of new information. GR directly adjusts weights to satisfy both individual and household-level constraints. The aim of GR is to weight a sample of individuals using information available on calibration variables, which are the marginals obtained from the aggregated data. These methods generalise the classical statistical ranking ratio method. This is done by minimising a distance measure between initial weights and final weights, subject to calibration equations.

All mentioned SR methods generate fractional weights that satisfy the marginals [5]. Therefore, the weights need to be integerised to construct the final population. This integerisation process is the allocation step in SR methods. Lovelace et al. [15] describe in total five integerisation methods: simple rounding, threshold approach, counter-weight method, proportional

probabilities, and truncate, replicate, sample (TRS). The first three methods are deterministic, the last two are probabilistic methods. They found that their developed method TRS has the best performance of the mentioned methods.

Albiston et al. [16], Rahman et al. [17], and Tuccillo et al. [18] use the TRS method developed by Lovelace et al. [15] for integerisation. Also, Zhou et al. [19] use TRS as their integerisation procedure. They also mention that the errors increase due to the integerisation process, and suggest that an improved integer programming algorithm might find a better solution.

2.1.2 Combinatorial optimisation (CO)

CO methods, also called fitness-based-sampling, directly generate a list of households and persons. Entire individuals or households being allocated to each zone. An advantage of using a CO method is that the data requirements are less restrictive than those for SR methods. Disadvantages are the computational complexity for large population sizes, and the possibility of getting stuck in a local optimum, global optimum solution can therefore not be guaranteed. Furthermore, also the zero-cell problem as mentioned in Section 2.1.1 is also present in CO methods. The CO method, as described by Templ et al. [12] and Voas et al. [20], can be summarised as follows:

1. Initial step: randomly chose a number of households from the sample to form an appropriately sized group.
2. Estimate the fitness of the selected population to the marginals (or relative entropy) for the group of households.
3. Add, remove, or swap a household with another from the sample, and recalculate the goodness of fit.
   (a) If the replacement improves the fit, the new group of households is accepted.
   (b) Otherwise the new group of households is rejected.
4. Repeat step 3 until a certain goodness-of-fit threshold or an arbitrarily fixed number of iterations has been reached.

Various optimisation procedures have been proposed to determine which household to add/remove/swapp next, such as hill-climbing (HC) [21], simulated annealing (SA) [22], and genetic algorithm (GA) [23].

With HC, steps are taken into the direction that improves the goodness-of-fit, as a result, the algorithm may get stuck in a local optimum. SA works very similar to HC, however, in step 3a, it is also possible to accept the new group of households even though the goodness-of-fit is worse. This allows the algorithm to find solutions that are eventually better. Therefore SA is less prone to getting stuck in a local optimum and in practice, finds the global optimum. This may take a long time depending on the initial sample.

The GA method is based on natural selection, survival of the fittest. In step 3a, parents are selected from the population, then a new population is generated, and finally, mutations are possible.

2.1.3 Statistical learning (SL)

SL methods, also known as simulation-based methods, focus on the joint distribution of all attributes in the sample by directly estimating a probability for each combination, including those not observed in the sample [11]. A major drawback of SL methods is that they fail to satisfy the zonal margins while satisfying the marginal distributions of all variables simultaneously. During the population synthesis process, when margins are available, it is important to match the generated synthetic population to the observed margins as closely as desired.

Many SL-based methods for generating synthetic populations have been proposed, such as: hierarchical Markov Chain Monte Carlo (hMCMC), Bayesian Network (BN), hierarchical mixture (HM), and deep generative modelling (DGM) based on a variational auto-encoder.

Since the margins are not precisely matched using SL methods, Sun and Erath [24] and Sun et al. [11] both recommend applying SR methods as a post-processing step (when marginals are available) on a synthetic sample of the population generated from SL methods so as to create a population matching the marginals. The downside of using an SR method as a post-processing step is that also an integerisation method is needed.

From this literature research, it is found that generating a two-layered population (i.e. linking persons and households) is one of the most challenging problems in population synthesis [6]. For a comprehensive overview of population synthesis methods, we refer to the review by Yamogo et al. [4].

2.2 Current population synthesizer

The current implementation of the population synthesizer falls in the category SR mentioned in Section 2.1. Here, the fitting part consists of IPF and the iNNLS method. After this part, we know the continuous weights of each household composition in the data from the mobility panel Netherlands (MPN). That means, we have the fractional number of times each household composition in MPN should be in the synthetic population. The allocation part of the population synthesizer is the use of SNET to integerise the outcome of iNNLS.

2.2.1 Input data

The data that serves as input for the IPF consists of dis-aggregated and aggregated data for both individuals and households. The aggregated data tells us the total number of individuals and households with a certain attribute, obtained from the *CBS wijken en buurten* data.

Furthermore, there is also data available from MPN, this contains observed household compositions, thereby defining the feasible solution space. This data links personal information with household information. The household composition gives us the information about the household (e.g. number of persons, number of cars) and information of each of the individuals in the household. The MPN data also gives a representative distribution over the segments.

An overview of the input data used in the current population synthesizer in Octavius can be found in Table 2.1.

TABLE 2.1: *Overview of available input datasets.*

Data	Description
CBS wijken (districts) en buurten (neighbourhoods)	The district and neighbourhood map data contains the geometry of all municipalities, neighbourhoods, and districts in the Netherlands with several key statistical figures as attributes. The boundaries of neighbourhoods and districts are largely based on what municipalities report to CBS. From this data, we obtain the zonal marginal totals for both the personal and the household attributes. Size: $\pm$ 13000 neighbourhoods.
Mobiliteitspanel Nederland (MPN)	The Netherlands Mobility Panel (MPN) studies trends in the travel behaviour of a fixed group of individuals and households over a long period (since 2013). It can be used to determine how changes in personal and household characteristics and in other travel-related factors correlate with changes in travel behaviour. The research is carried out by a team of researchers from KiM Netherlands Institute for Transport Policy Analysis (Ministry of Infrastructure and Water Management). From this data, we obtain the household compositions of the respondents. Size: 16487 persons in 7243 households, of which 2495 usable household compositions for population synthesis.

2.2.2 Iterative proportional fitting (IPF)

IPF is used to find the margins of how many individuals and households there are with each combination of personal and household attributes. These margins are obtained separately for individuals and households.

Since IPF minimises the relative entropy, it finds the solution with a distribution that is closest to the initial sample distribution given by the MPN data.

A detailed description of the IPF procedure is given based on the formulation by Reffel [25]. IPF takes a non-negative input matrix $A \in \mathbb{R}^{k\times l}_{\geq 0}$ (from MPN) with positive row sums $a_{i+}\ \forall\ i$ and positive column sums $a_{+j}\ \forall\ j$, and two positive vectors $r \in \mathbb{R}^{k}_{>0}$ and $c \in \mathbb{R}^{l}_{>0}$ that represent the known row and column marginals from the ***CBS wijken en buurten*** data. The input problem is formed by the triple (A, r, c). The procedure is initialised by setting $A(0) := A$. Let $A(t)$ be the matrix after step t. The IPF sequence A(t) is calculated by iteratively repeating the following two steps:

1. Odd steps $t + 1$ fit row sums to row margins. All entries in the same row are multiplied by the same multiplier

$$a_{i,j}(t+1) := \frac{r_i}{a_{i+}(t)} a_{ij}(t) \qquad \forall \text{ entries } (i,j). \tag{2.1}$$

2. Even steps $t + 2$ fit column sums to column margins. All entries in the same column are multiplied by the same multiplier

$$a_{i,j}(t+2) := \frac{c_j}{a_{+j}(t+1)} a_{ij}(t+1) \qquad \forall \text{ entries } (i,j). \tag{2.2}$$

These steps are repeated until the selected level of convergence is reached. Example 1 demonstrates one iteration of the IPF procedure.

Example 1. The IPF procedure is demonstrated by means of a small example where we have $r = \begin{bmatrix}1000 & 1050\end{bmatrix}^{\mathsf{T}}$, $c = \begin{bmatrix}400 & 1000 & 650\end{bmatrix}$, and $A = \begin{bmatrix}200 & 450 & 350\\ 200 & 550 & 300\end{bmatrix}$. For the initialisation we have the following table:

	age 0-17	age 18-64	age 65+	sum	margin
male	200	450	350	1000	70
female	200	550	300	1050	80
sum	400	1000	650	2050	
margin	30	80	40		150

In the first step, we find $a_{1j}(1) = \frac{70}{1000}a_{1j}(0)\ \forall\ j$ and $a_{2j}(1) = \frac{80}{1050}a_{1j}(0)\ \forall\ j$. This gives the following table:

	age 0-17	age 18-64	age 65+	sum	margin
male	14	31.5	24.5	70	70
female	15.2	41.9	22.9	80	80
sum	29.2	73.4	47.4	150	
margin	30	80	40		150

In the second step, we find $a_{i1}(2) = \frac{30}{29.2}a_{i1}(1)\ \forall\ i$, $a_{i2}(2) = \frac{80}{73.4}a_{i2}(1)\ \forall\ i$, and $a_{i3}(2) = \frac{40}{47.4}a_{i3}(1)\ \forall\ i$. This gives the following table:

	age 0-17	age 18-64	age 65+	sum	margin
male	14.4	34.3	20.7	69.4	70
female	15.6	45.7	19.3	80.6	80
sum	30	80	40	150	
margin	30	80	40		150

In the above example, we completed one iteration of the IPF procedure in two dimensions. IPF is demonstrated here in two dimensions. However, it can easily be extended to multiple dimensions by performing a step with multipliers for each dimension in one repeat loop of the procedure. Furthermore, a multidimensional table can be reformulated as a two-dimensional table, see Example 2.

Example 2. When given the three dimensions gender (male, female), age (0-17, 18-64, 65+), and participation (working, student). We can transform this into two dimensions by combining the attributes of two attribute categories. Then the columns could be for the age: [0-17], [18-64], and [65+]; and the rows for the combinations of gender and participation: [male, working], [male, student], [female, working], and [female, student].

If all input data is strictly positive, IPF is proven to converge [26, 27].

Theorem 1. *If all entries of* (A, c, r) *are strictly positive, then IPF converges to a unique solution.*

In our literature search (Section 2.1), we found that one of the drawbacks of IPF is the zero-cell problem. This problem is explained in Section 2.1.1. The current population synthesizer deals with this by replacing the zeros by a very small value: 10^{-5}.

2.2.3 Iterative non-negative least squares (iNNLS)

As mentioned before in Section 2.1.1, IPF in itself cannot simultaneously estimate both household and individual-level attributes. Therefore, another step is needed to link individuals to households, creating the household compositions. In the current population synthesizer, this is done using the iterative Non-Negative Least Squares (iNNLS) procedure, self-developed by Dat.mobility. This is an extension of the Non-Negative Least Squares (NNLS) procedure [28] to enforce uniqueness of the solution. It ensures that we obtain the same unique solution each time the model is run. The margins obtained from IPF and the MPN data form the input for iNNLS. With iNNLS, we find how many households there should be of each household composition. NNLS yields the following optimisation problem:

$$\begin{aligned} \min \quad & \|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{b}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{w} \geq 0, \end{aligned} \tag{2.3}$$

where $\|\cdot\|_2$ represents the Euclidean norm. The matrix $\boldsymbol{A}$ represents the households in the MPN data. The vector $\boldsymbol{b}$ contains the segment totals obtained from the IPF step. The vector $\boldsymbol{w}$ gives the weight for each household composition.

Finding linearly independent blocks of households Since we have a lot more households in MPN than there are segments, there must be linearly dependent households. Having linearly dependent households may have as a result that there is no unique solution to NNLS. Therefore, the first step of iNNLS is to find blocks of linearly independent households. To do this, we follow the following steps:

1. Start with the first household not yet in a linearly independent block.
2. Loop through all other households not yet in a linearly independent block.
   (a) If the household is not linearly dependent on the households already in the block, add it to the block.
3. If no more households can be added to the block, then start a new block with the next household not yet in a linearly independent block.
4. Repeat until all households are in a linearly independent block.

Iteratively solve NNLS for linearly independent blocks Now that we have the linearly independent blocks, we iteratively solve NNSL for each linearly independent block.

1. Initialise weights by setting $\boldsymbol{w}$ equal to the frequency of the households in the MPN data.
2. For each linearly independent block:
   (a) Correct the vector $\boldsymbol{b}$ by subtracting the totals contributed by the other blocks given the previous weights.
   (b) Solve the NNLS problem with the block and the corrected values of the vector $\boldsymbol{b}$.
   (c) Update the weights for the household in the block.
3. The weights are now updated for all households, so check the convergence criterion.
   (a) If $\frac{||\text{updated weights} - \text{previous weights}||}{(\text{number of updated weights} > 0)} < \epsilon$, then iNNLS is finished.
   (b) Otherwise, repeat step 2 with previous weights = updated weights

The found weights are continuous values. As mentioned before, we require a discrete population, therefore the weights need to be integerised to get a synthetic population that represents an actual population. How this is done in the current population synthesizer is discussed in the next section.

In an earlier version of the population synthesizer in Octavius, Iterative Proportional Updating (IPU) [29] was used before iNNLS was developed. This proved to be a too strict approach (as it implicitly adheres as much as possible to the distribution across households in the MPN data). NNLS lets go of that restriction and can use any household in the MPN data; whatever makes it fit to the margins.

2.2.4 Statistical noise elimination technique (SNET)

The method used to integerise the outcome of iNNLS to create individuals is SNET. The drawback of SNET is that it cannot be applied to a two-layered population. Therefore the population is collapsed into one layer by interpreting the household properties as personal attributes. Each created agent thus has characteristics in such a way that it not only has personal information but also information about the agents' household, such as the number of cars in the household. SNET is applied to this one-layered population. Therefore, the household compositions are not maintained and there is no household consistency. Details about SNET are described by Klein Kranenbarg [3].

Instead of SNET, TRS method was considered in the past for the allocation step to integerise the outcome of iNNLS. However, it was found that due to the small number of households per zone, the TRS method results in large deviations because there is not enough sampling to average out well.

2.3 Summary of population synthesis methods

As shown in the previous paragraphs, there are several population synthesis methods, both in literature and used in the current population synthesizer in Octavius. Table 2.2 summarises the pros and cons of these methods.

All statistical learning (SL) methods lack the ability to match zonal margins and do not support unicity. The combinatorial optimisation (CO) methods are able to match the zonal margins, but lack unicity and require a lot of computation time. Most of the synthetic reconstruction (SR) methods have all the requirements except the ability to generate two-layered populations. Only IPU/HIPF and iNNLS can deal with this, however, the allocation methods break the result down to one-layered again.

TABLE 2.2: *Summary of pros and cons of each method.*

Method	Matching zonal margins	Two-layered population	Most-likely result	Unique result	Computation time
IPF (SR fitting)	+	-	+	+	+
iNNLS (SR fitting)	+/-	+	+	+	+/-
IPU/HIPF (SR fitting)	+	+/-	+	+	+
ENT/GR (SR fitting)	+	-	+	+	+
TRS (SR allocation)	+/-	-		-	+
SNET (SR allocation)	+/-	-		+	+
HC/SA (CO)	+	+	+/-	-	-
GA (CO)	+	+	+/-	-	-
hMCMC (SL)	-	+	+	-	+/-
BN (SL)	-	+	+	-	+/-
HM (SL)	-	+	+	-	+/-
DGM (SL)	-	+	+	-	+/-

3 Methodology

Based on the problem description, research questions, and background, this section first defines a concise mathematical problem formulation that meets the basis requirements in Section 3.1. Furthermore, Section 3.2 discusses additional requirements. In Section 3.3, the basis mathematical program is extended in various ways to incorporate the additional requirements. From there, we discuss the evaluation framework and define performance metrics in Section 3.4.

3.1 Mathematical problem formulation

In this section, the population synthesis problem is formulated as a mathematical optimisation problem. The two main requirements for our base model are maintaining household consistency and complying to zonal margins. To achieve this, the mathematical program needs a well-defined objective. In the context of population synthesis, this objective can focus either on the marginal totals or on the frequency of each segment within the synthetic population. Since our primary goal is to create a synthetic population that aligns with the known marginal totals, we use these as the basis for our objective function. The two main requirements are described as follows, the additional requirements are discussed in Section 3.2.

- **Household consistency**
  The synthetic population should be comprised of consistent households. Individuals in the synthetic population are grouped into households that reflect the composition of real-world households, maintaining relationships and attributes. Details can be found in Section 3.1.1.
- **Zonal margins**
  The synthetic population should match all zonal marginal totals. Zonal margins represent the known marginal totals of attributes within predefined zones. These totals are incorporated in population synthesis to ensure the synthetic population reflects the demographic characteristics of each zone. In doing so, the population synthesizer should allow for prioritisation of attributes. Details can be found in Section 3.1.2.

Before we dive into the mathematical formulations of these requirements and the optimisation problem, we define the sets, parameters, and variables that will be used from here on. The notation and descriptions of these can be found in Tables 3.1 to 3.3. The sets and parameters are all obtained from available input data described in Section 2.2.1.

TABLE 3.1: *Description of sets.*

Set	Description
S^P	Set of personal segments.
S^H	Set of household segments.
A^P	Set of personal attributes.
A^H	Set of household attributes.
A^Z	Set of zonal attributes.
S	Set of all personal and household segments, $S := S^P \cup S^H$.
A	Set of all personal, household, and zonal attributes $A := A^P \cup A^H \cup A^Z$.
B	Set of attribute categories.
C	Set of possible household compositions known from the MPN data.

TABLE 3.2: *Description of parameters.*

Parameters	Description
M_s	The values of the known distribution of segments $s \in S$, from MPN.
T_a	The known margins for $a \in A$, from *CBS wijken en buurten*.
$n_{c,s}$	number of times the segment $s \in S$ is in a household composition c, is 0 or 1 for $s \in S^H$ and $\mathbb{N}$ for $s \in S^P$.

TABLE 3.3: *Description of variables.*

Variables	Description
w_c	Weights determining how many households there should be of composition $c \in C$.
m_s	Variable for the number of times segment $s \in S$ is in the synthetic population.
t_a	Variable for the marginal total of each attribute $a \in A$.
x_a	Absolute error between known marginal totals and marginal totals in synthetic population.

The weights w_c are the decision variables, the remaining variables are auxiliary variables.

3.1.1 Household consistency

Since the synthetic population must include household composition, we need to know which agents are in which household. We know this when we have a solution to the population synthesis problem that tells us how many (integer) times we should have each possible household composition from MPN (Section 2.2.1) in our synthetic population. Each household composition includes in which segment the individuals in that household are. That way agents can directly be extracted from the household, including the household identification. Household consistency is thus maintained when the household weights $w_c \in \mathbb{N}$.

3.1.2 Zonal margins

Ideally, the synthetic population should perfectly match all marginal totals. However, this is often not feasible for every attribute because we are limited to the possible household compositions in the MPN data. Therefore, there may not be a combination of households that adds up to the given attribute totals. To address this, the mathematical program minimises the absolute differences between the known marginal totals and those derived from the synthetic population.

As the synthetic population should match the zonal margins, we need a measure for determining the difference between the margins in the synthetic population and the known marginal totals. Let t_a be the margins in the synthetic population, and let T_a be the known marginal totals for $a \in A^P \cup A^H$. There are several error-based measures where a lower value is preferred. For example, the total absolute error (TAE):

$$\mathit{TAE} = \sum_{a \in A} |T_a - t_a| \,. \tag{3.1}$$

A drawback of the TAE is that it does not consider the size of the population or the number of attributes and attribute-values. Based on the TAE, we have the standardised absolute error (SAE), which scales with the actual population size. This enables the comparison between the goodness-of-fit of populations of zones with different population size:

$$\mathit{SAE} = \frac{\mathit{TAE}}{T_{\text{zone\_inh}}} = \frac{\sum_{a \in A} |T_a - t_a|}{T_{\text{zone\_inh}}}. \tag{3.2}$$

Furthermore, there are also error measures based on the sum squared error (SSE):

$$\mathit{SSE} = \sum_{a \in A} (T_a - t_a)^2 \,. \tag{3.3}$$

Similar to TAE, SSE does not consider the size of the population or the number of attributes and attribute-values. The SSE differs from the TAE by giving a heavier penalty to larger errors. Based on the SSE, we have the root mean squared error (RMSE), which considers the number of attributes:

$$\mathit{RMSE} = \sqrt{\frac{\text{SSE}}{|A|}} = \sqrt{\frac{\sum_{a \in A} (T_a - t_a)^2}{|A|}}. \tag{3.4}$$

A drawback is that smaller populations can be reported as closer to the actual population, while the absolute number of errors might be the same. The standardised RMSE compensates for this:

$$\mathit{SRMSE} = \frac{\mathit{RMSE}}{\frac{\sum_{a \in A} T_a}{|A|}} = \frac{\sqrt{\frac{\sum_{a \in A} (T_a - t_a)^2}{|A|}}}{\frac{\sum_{a \in A} T_a}{|A|}}. \tag{3.5}$$

Any of these error-based measures can be used in the objective function of the mathematical program. If all margins in the synthetic population are equal to the known margins, then $T_a - t_a = 0$, and thus any mentioned error-based measure will be zero.

3.1.3 Mixed-integer program

A mathematical formulation of the population synthesis problem is formulated for generating a two-layered synthetic population using the SAE (eq. (3.2)) in the objective function. We choose this error measure, because we do not want to assign a higher penalty to one large error than to multiple smaller errors. The total of absolute errors over personal attributes are standardised with the known total number of inhabitants per zone, and the total of absolute errors over household attributes are standardised with the known total number of households per zone. We formulate the following mixed-integer linear program (MILP) to find the weights of each household composition.

$$\min \quad \frac{\sum_{a \in A^P} x_a}{T_{\text{zone\_inh}}} + \frac{\sum_{a \in A^H} x_a}{T_{\text{zone\_hh}}} \tag{3.6a}$$

$$\sum_{\{s \in S \text{ where } a \in s\}} \sum_{c \in C} w_c n_{c,s} = t_a \qquad \forall\, a \in A^P \cup A^H \tag{3.6b}$$

$$\sum_{c \in C} w_c = t_{\text{zone\_hh}} \tag{3.6c}$$

$$\sum_{s \in S^P} \sum_{c \in C} w_c n_{c,s} = t_{\text{zone\_inh}} \tag{3.6d}$$

$$T_a - t_a \leq x_a \qquad \forall\, a \in A \tag{3.6e}$$

$$-T_a + t_a \leq x_a \qquad \forall\, a \in A \tag{3.6f}$$

$$T_{\text{zone\_inh}} - t_{\text{zone\_inh}} = 0 \tag{3.6g}$$

$$w_c \in \mathbb{N} \qquad \forall\, c \in C \tag{3.6h}$$

$$t_a \in \mathbb{R} \qquad \forall\, a \in A \tag{3.6i}$$

$$x_a \in \mathbb{R} \qquad \forall\, a \in A \tag{3.6j}$$

The objective (3.6a) minimises the standardised absolute error. Constraints (3.6b) sets the marginal totals for each attribute $a \in A^P \cup A^H$. Constraints (3.6c) and (3.6d) determine the marginal totals for the total number of households and inhabitants in the synthetic population. Constraints (3.6e) and (3.6f) determine the absolute errors x_a. Constraint (3.6g) set the absolute error for the number of inhabitants to zero. The known marginal total of inhabitants is the most important explanatory variable for the overall use of the mobility system, therefore it is considered the most important margin. As a result, the number of individuals in the synthetic population must equal the known marginal total of inhabitants. Constraints (3.6h) make sure the variables w_c are natural numbers (integer and non-negative). Constraints (3.6i) and (3.6j) make sure the variables t_a and x_a are real numbers. Since the weights w_c are integer, the totals t_a and absolute errors x_a are by definition also integer without setting additional integer constraints.

This MILP includes the household consistency requirement as it finds weights $w_c \in \mathbb{N}$ on household level, where households are constrained to feasible compositions through the set of possible household compositions from MPN data. It also complies to the total number of inhabitants exactly, the other marginal attribute totals in the synthetic population are as close as possible to the known totals by minimising the SAE over personal and household attributes.

3.2 Additional requirements

Synthetic populations will be generated for each zone individually, following a zone-by-zone approach. This means that the population synthesis process will be conducted separately for each predefined area or zone, ensuring that the characteristics of each zone are accurately represented in the synthetic population. By treating zones independently, we can better account for the unique attributes, demographics, and marginal totals associated with each area, creating a synthetic population that reflects the specific conditions of each zone.

To achieve this, the population synthesizer must meet several requirements. These requirements ensure that the synthetic populations do not only comply to the marginal totals, but also stable and consistent. The synthesizer must align with the given marginal totals at both the household and personal levels, maintain realistic household compositions, and produce stable outputs. Furthermore, the synthesizer should be robust enough to handle variations in zone characteristics and support scalability to accommodate a large number of zones effectively. To address these aspects, we outline the following requirements.

- **Stability**
  The resulting synthetic population from a scenario run should, compared to the reference output, only have changes that are related and proportional to the changes in input. E.g. adding a few more inhabitants should not change the composition of the already existing population in a zone. Details can be found in Section 3.2.1.
- **Unicity**
  The result should be unique. This may be achieved by finding the most likely solution. Therefore, we want the solution that is the most likely given the input data. The most important herein is that the distribution over the segments in the solution should be similar to the distribution over the segments in the OViN/MPN data. This can be achieved by minimising the relative entropy. Details can be found in Section 3.2.2.
- **Performance**
  The method should not be complex in both space and time. The model should be able to generate synthetic populations without needing an extensive amount of memory and computation time. The implemented methods will be tested and compared with each other using a small instance. However, the final method should also work for real-world instances. Details can be found in Section 3.2.3.

The household consistency and the zonal margin on total number of inhabitants (discussed in Section 3.1.3) are the most important, followed by the stability and matching the remaining zonal margins as closely as possible. The least important requirements are unicity and performance. However, these requirements do have some bound to make sure the methods are usable in practice.

3.2.1 Stability

The stability requirement consists of two parts: comparability and sensitivity. Comparability is required when running a scenario with changes in the attribute totals of the population. For example, when a newly build street is added to a zone, or when the distribution over segments

and/or attributes changes because of demographic changes, we want to be able to compare the results of the reference with the scenario. Therefore, changes in the input of the population synthesizer should result in proportional and related changes in the output. For example, when adding agents in a scenario, we do not want the agents in the reference population to change characteristics.

Sensitivity is when we are talking about the effect of very small changes (noise) to the totals on the households that are selected. Such small changes or noise should ideally not change the household weights very much.

Since we are optimising the weights for each household composition in the MPN data, our goal is to minimise changes in the weight vector. To measure the distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, we can use either the Manhattan distance

$$\sum_{i \in I} |x_i - y_i| \,, \tag{3.7}$$

or the Euclidean distance:

$$\sqrt{\sum_{i \in I} (x_i - y_i)^2}. \tag{3.8}$$

We opt for the Manhattan distance because, in our context, one large deviation is not necessarily worse than multiple small errors.

The objective is to ensure that differences in input data do not result in unnecessary re-allocation of individuals between household categories. When the margins for one attribute category are adjusted, we aim to minimize the impact on other attributes. While we recognize that strong correlations between attributes mean some changes are inevitable, our goal is to limit these changes as much as possible.

Metric using number of times each segment is in a household

With this metric, we compare the two households sets (which contains individuals as well) by calculating the distance between the segments in each household in the reference set with the segments each household in the scenario set. The number of times segments are in a household is represented by a vector. To determine the distance between a household vector in the reference set and a household vector in the scenario set, we use the Manhattan distance as mentioned in Section 3.2.1.

Let HH^R be the set of households in the reference solution and HH^S the set of households in the scenario solution. Then the distance on the segment level between household $u \in HH^R$ and $v \in HH^S$ is calculated as follows:

$$\sum_{s \in S} |u_s - v_s| \,. \tag{3.9}$$

With these distances, we find the best assignment that minimises the total distance between the two sets using a distance matrix. In this matrix, each row corresponds to a household in the reference set and each column corresponds to a household in the scenario set. If one set contains more households than the other, then the households that are not assigned to another household are not considered in calculating the distance. The best assignment is found using the Hungarian algorithm.

Hungarian algorithm The Hungarian algorithm works for square matrices. For us, the two household sets we are comparing, are not necessary of the same size. If this is the case, then we add extra rows or columns containing zeros to make the matrix square. In the end, the assignment of households to added rows of zeros are discarded.

1. Row reduction: subtract the smallest value of each row from all values in the row.
2. Column reduction: subtract the smallest value of each column from all values in the column.
3. Test for an optimal assignment: minimum number of lines required to cover all zeros in the matrix. If this number is equal to the number of rows (and columns), then an optimal assignment can be found, and go to step 5.
4. Shift zeros:
   (a) Find smallest uncovered value.
   (b) Subtract this value from all uncovered values, and add it to each value that is intersected by two lines.
   (c) Remove lines and go back to step 3.
5. Making the final assignment: choose zeros such that only one zero is selected per row and column. The indices of the chosen zeros are the assignment. There may be multiple options, in that case the total costs (in our case distance) is the same.

In order to compare runs with different noise added, we standardise the total distance of the best assignment by dividing it by the sum of absolute change of attribute totals. Furthermore, to compare between zones, we can also divide the total distance between two household sets by the known number of households in the zone.

A similar metric was also considered, but based on the number of times each attribute is in a household. For details, see Appendix A. We choose not to use this metric as it does not give a representative idea of what changes in a synthetic population when attribute totals are changed.

Metric for changes in marginal totals per attribute category

We also look at a metric that focuses on the changes in marginal totals per attribute category. Per attribute category, we determine how much change there is in the attribute totals, both in the input totals and in the totals of the resulting population, compared to the reference totals. For each attribute category $b \in B$, where B is the set of attribute categories, we use the following equations to determine the difference between the reference case and the scenario case. Let $c_{b,\text{in}}$ be the change in input of attribute category b, and let $c_{b,\text{out}}$ be the change in output of attribute category b.

$$c_{b,\text{in}} = \sum_{a \in b} |T_a - Q_a| \tag{3.10}$$

$$c_{b,\text{out}} = \sum_{a \in b} |t_a - q_a| \,, \tag{3.11}$$

where T_a are the input attribute totals of the reference, Q_a are the input attribute totals of the scenario, t_a are the attribute totals in the solution to the reference case, and q_a are the attribute totals in the solution for the scenario.

3.2.2 Unicity

The unicity requirement consists of two parts, namely, the most likely solution and a unique solution. Finding the most likely solution does not imply that the solution is unique as there may be multiple most likely solutions that are evenly likely. First, let us discuss the requirements to find the most likely solution. A most likely solution can be found by minimising the relative entropy, also known as the Kullback-Leibler divergence:

$$D_{\text{KL}}(P||Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right), \tag{3.12}$$

where P represents the distribution in the outcome of the model, and Q represents the distribution in the known sample. Since P and Q represent distributions, it must be that $\sum_{x \in X} P(x) = 1$ and $\sum_{x \in X} Q(x) = 1$.

We want to compare the distribution over the segments in our synthetic population with the known distribution over the segments from MPN. Since the segment totals do not sum up to one, we need to normalise the values. Therefore, we split the relative entropy into two parts, one for the household segments and one for the personal segments. The values of household segments are divided by the total number of households in the set. The values of personal segments are divided by the total number of individuals in the set. Let m_s^* and M_s^* be the normalised values for $s \in S$. Then the relative entropy for household and personal segments is:

$$D_{\text{KL}}^H(m^*||M^*) = \sum_{s \in S^H} m_s^* \log \left(\frac{m_s^*}{M_s^*} \right) \qquad \text{for household segments,} \tag{3.13}$$

$$D_{\text{KL}}^P(m^*||M^*) = \sum_{s \in S^P} m_s^* \log \left(\frac{m_s^*}{M_s^*} \right) \qquad \text{for personal segments.} \tag{3.14}$$

Adding the relative entropy minimisation objective to the MILP (eq. (3.6)) does not need to result in a unique solution. There can still be differences in the outcome of two separate runs of the model when they have identical objective values.

While unicity is one of the requirements because of practical application for users, it should be mentioned that a stochastic component allows the model to consider and produce alternative and potentially more realistic populations [22].

3.2.3 Performance

The performance of the software implementations of the algorithms is important in both time and space. Time performance concerns the computational time the population synthesizer requires given a certain input size. The memory or space required by the population synthesizer may play an important part in the performance of the algorithm, especially in cases when the memory is limited.

The prototype implementation should give results within 16 hours on a high-end machine. Whereas memory requirements should be lower than 64GB.

3.3 Extensions of the mixed-integer program

The MILP (eq. (3.6)) in Section 3.1.3 meets the requirement of household consistency and zonal marginal totals. This MILP is used as a basis and can be extended in various ways to improve the performance on the metrics mentioned in Section 3.4 by taking into account relative entropy, and a suggestion to add stability when a scenario is run and compared to the reference.

To incorporate unicity, we add the relative entropy component. There are two options for this. In the first option (Section 3.3.1), we determine the segment totals in the synthetic population found by solving the mathematical program and add the relative entropy to the objective function to minimise. In the second option (Section 3.3.2), the segment totals determined during the IPF step of the current population synthesizer in Octavius are used, then the SAE between those segment totals and the segment totals in the synthetic population found when solving the mathematical program is minimised. Finally, in Section 3.3.3, we include an extension that improves the stability when a scenario is run.

3.3.1 Adding relative entropy to the objective function

When the relative entropy is added to the objective, the objective function (3.6a) is replaced by:

$$\begin{aligned}\min \quad & \frac{\sum_{a\in A^P} x_a}{T_{\text{zone\_inh}}} + \frac{\sum_{a\in A^H} x_a}{T_{\text{zone\_hh}}} \\ & + \sum_{s\in S^P} m_s^* \log\left(\frac{m_s^*}{M_s^*}\right) + \sum_{s\in S^H} m_s^* \log\left(\frac{m_s^*}{M_s^*}\right),\end{aligned} \tag{3.15a}$$

where

$$\begin{aligned} m_s^* &= m_s/t_{\text{zone\_inh}} && \forall\, s \in S^P, \\ m_s^* &= m_s/t_{\text{zone\_hh}} && \forall\, s \in S^H, \\ M_s^* &= M_s/T_{\text{zone\_inh}} && \forall\, s \in S^P, \\ M_s^* &= M_s/T_{\text{zone\_hh}} && \forall\, s \in S^H. \end{aligned}$$

The constraints to add to determine the distribution over segments are:

$$\sum_{c\in C} w_c n_{c,s} = m_s \qquad \forall\, s \in S. \tag{3.15b}$$

The function for relative entropy adds non-linearity to the objective function. Not only because of the $\log$-function, but also because the variables $m_s^*\ \forall s \in S$ are determined by dividing one variable by another. To solve this problem with a MIP, to objective needs to be linearised.

The first thing we can do is to linearise the m_s^* variables by replacing $t_{\text{zone\_inh}}$ by $T_{\text{zone\_inh}}$. This will not change the values of $m_s^*\ \forall s \in S^P$, because constraint (3.6g) ensures that $t_{\text{zone\_inh}} = T_{\text{zone\_inh}}$. Similarly, for $m_s^*\ \forall s \in S^H$ we can replace $t_{\text{zone\_hh}}$ by $T_{\text{zone\_hh}}$. However, it should be mentioned that for the total number of households there is no constraint that sets $t_{\text{zone\_hh}} = T_{\text{zone\_hh}}$. Therefore, the values of $m_s/t_{\text{zone\_hh}}$ need not be equal to $m_s/T_{\text{zone\_hh}}$. But since we

minimise the SAE, the values of $t_{\text{zone\_hh}}$ and $T_{\text{zone\_hh}}$ are as close as possible. So $m_s/T_{\text{zone\_hh}}$ will give a good approximation. (The larger $T_{\text{zone\_hh}}$, the better the approximation since an absolute error on the total number of households has less effect on larger values than on smaller values.)

Since we know what values the m_s variables are possible beforehand, we can create indicator variables $z_{i,s} \in \{0, 1\}$ for each possible value of m_s, where i is the value of m_s. For the $z_{i,s}$ variables we have $z_{i,s} = 1$ if $m_s = i$, else $z_{i,s} = 0$. To ensure this, we need to add some constraints for the consistency of the $z_{i,s}$ variables.

$$\sum_{i \in I^H} i z_{i,s} = m_s \qquad \forall\, s \in S^H \tag{3.16a}$$

$$\sum_{i \in I^H} z_{i,s} = 1 \qquad \forall\, s \in S^H \tag{3.16b}$$

$$\sum_{i \in I^P} i z_{i,s} = m_s \qquad \forall\, s \in S^P \tag{3.16c}$$

$$\sum_{i \in I^P} z_{i,s} = 1 \qquad \forall\, s \in S^P \tag{3.16d}$$

where $I^H = \{0, \ldots, T_{\text{zone\_hh}}\}$ and $I^P = \{0, \ldots, T_{\text{zone\_inh}}\}$.

Furthermore, to remove the non-linearity caused by the $\log$-function from the objective, we can calculate the values of $m_s^* \log\left(\frac{m_s^*}{M_s^*}\right)$ for both $s \in S^H$ and $s \in S^P$ beforehand. The objective function then becomes

$$\begin{aligned} \min \quad & \frac{\sum_{a \in A^P} x_a}{T_{\text{zone\_inh}}} + \frac{\sum_{a \in A^H} x_a}{T_{\text{zone\_hh}}} \\ & + \sum_{s \in S^P} \sum_{i \in I^P} i^* \log\left(\frac{i^*}{M_s^*}\right) z_{i,s} + \sum_{s \in S^H} \sum_{i \in I^H} i^* \log\left(\frac{i^*}{M_s^*}\right) z_{i,s}, \end{aligned} \tag{3.16e}$$

where

$$\begin{aligned} i^* &= i/T_{\text{zone\_inh}} && \forall\, i \in I^P, \\ i^* &= i/T_{\text{zone\_hh}} && \forall\, i \in I^H, \\ M_s^* &= M_s/T_{\text{zone\_inh}} && \forall\, s \in S^P, \\ M_s^* &= M_s/T_{\text{zone\_hh}} && \forall\, s \in S^H. \end{aligned}$$

The $i^* \log\left(\frac{i^*}{M_s^*}\right)$ terms are constants and thus the objective is now linear.

3.3.2 Using segment totals from the IPF step in Octavius

The second option would be to use the IPF step of the current population synthesizer in Octavius and then minimise the SAE between the segment totals from IPF and the segment totals in the synthetic population found when solving the mathematical program. As mentioned in Section 2.2.2, IPF minimises the relative entropy and meets the requirement of marginal totals. Therefore, we would not need to include relative entropy in the mathematical program and thus avoid non-linearity. Furthermore, the objective function can be changed to minimise the SAE over segment totals instead of marginal totals on attributes. In this case, the objective

function (3.6a) changes slightly to minimise the absolute error between the segment totals obtained from IPF and the segment totals in the synthetic population. The absolute error variables $x_a \; \forall \; a \in A^P \cup A^H$ are replaced by absolute error variables $y_s \; \forall \; s \in S$ for the absolute error between the segment totals from IPF and the segment totals in the synthetic population. The objective function (3.6a) is replaced by:

$$\min \quad \frac{\sum_{s \in S^P} y_s}{T_{\text{zone\_inh}}} + \frac{\sum_{s \in S^H} y_s}{T_{\text{zone\_hh}}}. \tag{3.17a}$$

Again, we need to add a constraint to determine the segment totals:

$$\sum_{c \in C} w_c n_{c,s} = m_s \qquad \forall \; s \in S. \tag{3.17b}$$

The constraints (3.6e) and (3.6f) for determining the absolute errors are replaced by:

$$M_s - m_s \leq y_s \qquad \forall \; s \in S \tag{3.17c}$$
$$-M_s + m_s \leq y_s \qquad \forall \; s \in S \tag{3.17d}$$
$$T_a - t_a \leq x_a \qquad \forall \; a \in A^Z \tag{3.17e}$$
$$-T_a + t_a \leq x_a \qquad \forall \; a \in A^Z \tag{3.17f}$$

3.3.3 Adding stability for scenarios

In order to add stability to compare a scenario to the reference, the Manhattan distance (eq. (3.7)) between the known weights in the reference and the weights in the synthetic population of the scenario can be added to the objective function. The weights of the reference are additional input to the model, and thus constant. The objective function for the basis MILP (3.6) then becomes

$$\min \quad \frac{\sum_{a \in A^P} x_a}{T_{\text{zone\_inh}}} + \frac{\sum_{a \in A^H} x_a}{T_{\text{zone\_hh}}} + \frac{\sum_{c \in C} r_c}{T_{\text{zone\_hh}}} \tag{3.18a}$$

where r_c is the absolute difference between the weights of household composition $c \in C$ with the following additional constraints for the absolute values:

$$W_c^r - w_c \leq r_c \qquad \forall \; c \in C \tag{3.18b}$$
$$-W_c^r + w_c \leq r_c \qquad \forall \; c \in C \tag{3.18c}$$

We add this to each of the models, not only to the basis model (3.6) in Section 3.1.2, but also to the extensions mentioned above.

The added term $\frac{\sum_{c \in C} r_c}{T_{\text{zone\_hh}}}$ can also be added to the objective functions of the extensions in Sections 3.3.1 and 3.3.2, where then the additional constraints are also added to the MILPs.

3.4 Evaluation framework

In this study, we solve the MILPs proposed above using the open source HiGHS solver [30] with the JuMP package [31] in Julia programming language. The requirements for our population synthesis method and the resulting synthetic population are mentioned in Sections 3.1 and 3.2. Based on these requirements, we formulate the metrics below that are used in our evaluation framework. All metrics except the stability performance metrics are evaluated with the reference population. The stability performance metrics are evaluated in the sensitivity and stability analysis.

Household consistency

The main goal of the proposed methods is to get household consistency. However, it is quite difficult to formulate something neatly, because there is no household consistency in the current population synthesizer. Therefore, in the evaluation of the methods, we just check whether there is household consistency or not.

Metric for fit on marginal attribute totals

Of the generated synthetic population, we calculate the standardised absolute error of the marginal attribute totals. The SAE is calculated for personal and household attributes separately, using the following equations:

$$SAE^P = \frac{\sum_{a \in A^P} |t_a - T_a|}{T_{\text{zone\_inh}}} \tag{3.19}$$

$$SAE^H = \frac{\sum_{a \in A^H} |t_a - T_a|}{T_{\text{zone\_hh}}} \tag{3.20}$$

Stability performance metric

The stability performance metrics are used to evaluate the performance of the algorithm when applied to scenario studies. In Section 3.2.1, we discussed two metrics that give a good idea whether two household sets are similar. As the changes in the output should relate to the changes in input, we standardise the total distance between two household sets by dividing by the sum of absolute differences in input. The metric will thus give us how many segments are different per one household or person change in attribute totals.

$$\frac{\sum_{j=1}^{\max\{|HH^R|,|HH^S|\}} \sum_{s \in S} |u_{j,s} - v_{j,s}|}{\sum_{b \in B} c_{b,\text{in}}} \tag{3.21}$$

where HH^R is the set of households in the reference solution, HH^S is the set of households in the scenario solution, and $b \in B$ are the attribute categories.

Also, the metric for changes in marginal totals per attribute category is standardised in the same way, to be able to compare between different runs with different noise.

$$\frac{\sum_{b \in B} |c_{b,\text{in}} - c_{b,\text{out}}|}{\sum_{b \in B} c_{b,\text{in}}} \tag{3.22}$$

Unicity metric

The relative entropy is calculated for personal and household attributes separately, using the following equations (these are eqs. (3.13) and (3.14) in Section 3.2.2):

$$D_{\mathrm{KL}}^{P}(m^*||M^*) = \sum_{s \in S^P} m_s^* \log\left(\frac{m_s^*}{M_s^*}\right) \tag{3.23}$$

$$D_{\mathrm{KL}}^{H}(m^*||M^*) = \sum_{s \in S^H} m_s^* \log\left(\frac{m_s^*}{M_s^*}\right) \tag{3.24}$$

Performance

The requirements for performance are upper bounds. Therefore, we look at whether they are exceeded or not. The run time will be measured in seconds or minutes, whichever is most appropriate. For the space requirement, we look at the memory usage of the machine.

Car availability metric

In the current population synthesizer, there are several rules to determine whether a person can drive a car and/or be a car passenger. The rules for car usage are as follows:

- Car driver: a person must have a drivers license, and there must be at least one car in the household.
- Car passenger: besides the person itself, there must be at least one (other) adult and at least one car in the household.

The rule for car passengers is incomplete, because an adult would need to have a drivers license in order to be able to drive a passenger. In the current population synthesizer there is no household consistency and this cannot be check. When household compositions are maintained, it is possible to incorporate the requirement of a drivers license for the adult that may drive a passenger. By incorporating this, we expect the car availability for passengers to decrease. This metric is used to test whether the car availability decreases when household consistency is maintained.

3.5 Solution approaches

In the overview of population synthesis methods (Section 2.3), we mention pros and cons of the methods. Based on those findings and the problem formulation, we discuss suitable solution possibilities. In general, there are three approaches:

1. Solve the MILP directly.
2. Use parts of the current population synthesizer and use a different or adjusted method for the allocation (integerisation) step.
3. Use another population synthesis method for the entire population synthesizer.

The approaches that we consider in this report are solving the MILPs proposed in this section, and using the IPF step of the current population synthesizer and then solve a MILP instead of using iNNLS and SNET.

The statistical learning (SL) methods were not considered as they do not match zonal marginal totals and this is one of the main requirements. Combinatorial optimisation methods were considered, but due to the time complexity found to not perform well on a small test instance. Therefore, also these methods were not explored further.

The methods that are evaluated are as follows:

- Current population synthesizer, Section 2.2
- MILP SAE attributes, eq. (3.6)
- MILP SAE segments, eq. (3.17)
- MILP SAE attributes + relative entropy, eq. (3.16).

4 Case study

To test the performance of the methods, they are applied to the strategic transport model of Zwolle. In this project, the city of Zwolle is the main study area. The zones in and near Zwolle are small (area wise), and get larger the further away from Zwolle. In total, there are 1380 zones in this project, of which 515 in the main study area, 487 in the influence area, and 343 in the outer area. Zones outside the Netherlands are not included in the case study. The areas are visualised in Figure 4.1. Between the study area and the influence area, there are 35 zones without any inhabitants. Also within the three areas there may be zones without any inhabitants.

The mathematical programs formulated in Section 3 are implemented in Julia programming language using the JuMP package [31], and are solved using the open source HiGHS solver [30] on a 2.6 GHz Intel Core i7 Notebook with 32GB RAM.

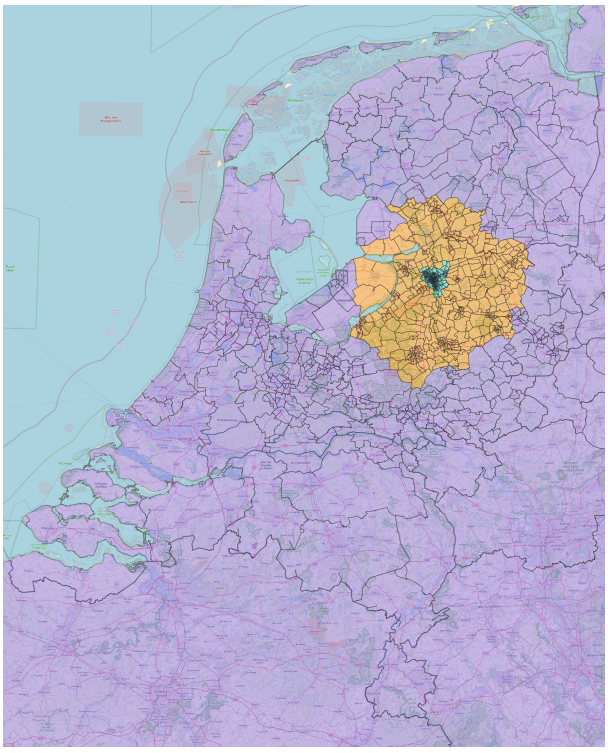


FIGURE 4.1: *Map of the three areas in the transport model of Zwolle. The purple zones belong to the outer area. The orange zones belong to the influence area. The blue zones belong to the main study area.*

Box-plots For showing the results, we use box-plots. A box-plot is a graphical representation of data distribution that summarizes key descriptive statistics. It is constructed using five key statistics, often referred to as the five-number summary:

1. Minimum: The smallest data point, excluding any outliers.
2. First Quartile (Q1): The value below which 25% of the data fall.
3. Median (Q2): The middle value of the data, splitting the data into two equal halves.
4. Third Quartile (Q3): The value below which 75% of the data fall.
5. Maximum: The largest data point, excluding any outliers.

These components are used to draw a box-plot as follows:

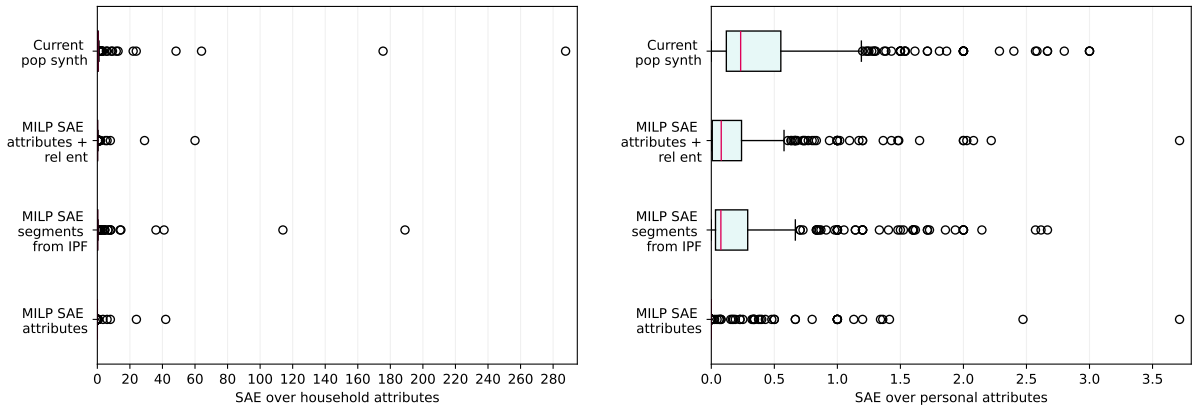
- Box: Encloses the interquartile range (IQR), which spans from Q1 to Q3. This represents the middle 50% of the data. In our plots represented by the light blue box.
- Median: A line inside the box marks the median (Q2), providing a measure of central tendency. In our plots represented by a pinkish red line.
- Whiskers: Extend from the box to the smallest and largest data points within a range of $1.5 \cdot \text{IQR}$ from Q1 and Q3, respectively.
- Outliers: Represented as individual points beyond the whiskers.

4.1 Results for reference population

In this results section, we discuss the results from the models for the reference population. We split this in two or three parts: results of the main study area, results of the influence area, and results of the outer area. This approach is used because the zones increase in size as their distance from Zwolle grows. With larger zones, the runtimes are expected to increase; and the metrics regarding the standardised absolute errors and the relative entropy will likely give lower values, as deviations from the known margins and distribution over segments is averaged out more.

4.1.1 Main study area

In this section, we look at the results for the zones in the main study area of the Zwolle project using the current population synthesizer, solving the MILP minimising SAE over attribute totals and relative entropy, solving the MILP minimising SAE over segment totals obtained using IPF, and solving the MILP minimising the SAE over attribute totals.



(A) *Box-plot of the SAE's of household attributes with different methods.*

(B) *Box-plot of the SAE's of personal attributes with different methods.*

FIGURE 4.2: *Box-plots of the fit on margins (SAE of attributes) of different population synthesis methods when applied to the zones in the main study area in the Zwolle project.*

In Figure 4.2a, we see that there are some outliers for which the fit on marginal totals is (very) poor. Figure 4.3 identifies these outliers.

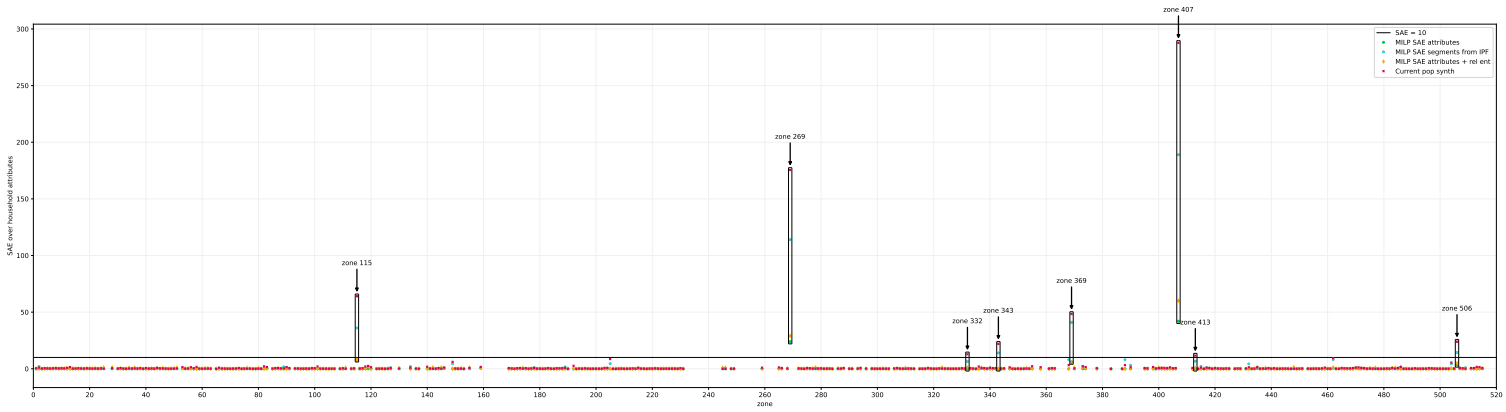


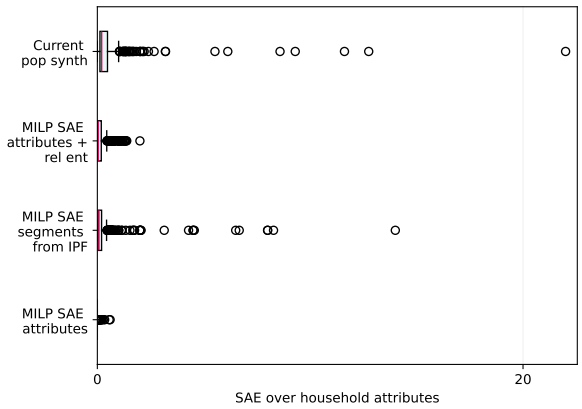
FIGURE 4.3: *SAE's of household attributes per zone in the main study area in the Zwolle project. The horizontal black line represents SAE = 10.*

The SAE of household attributes with the different methods is large in the same zones, as can be seen in Figure 4.3. In Table 4.1, we see that in these "difficult" zones, the average number of persons per household is very high. In the first five zones listed, this average is higher than the maximum household size found in MPN data (9 persons, which only occurs five times in the MPN sample).

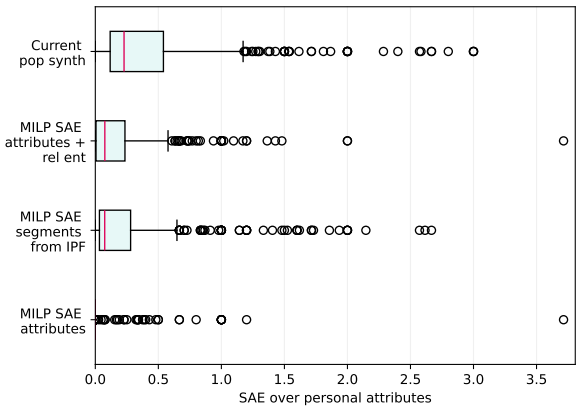
TABLE 4.1: *Highest SAE's of household attributes per zone, and the number of persons per household on average.*

Zone	SAE hh attributes				persons	households	avg. # persons/hh
	MILP att	MILP seg	MILP att + rel ent	current pop synth			
407	42	189	60	287.667	127	1	127
269	24	114	29	175.533	75	1	75
115	8	36	8	64	25	1	25
369	6	41	6	48.333	23	1	23
506	3.462	14.462	5	23.908	152	13	11.69
343	0	14	0	22	7	1	7
332	0	6.5	1	12.75	10	2	5
413	0	6.667	0.667	11.611	15	3	5

These zones with on average more than 9 persons per household are clearly consistency errors for which the method is cannot provide a good solution. Therefore, in the remainder of the results, these zones are excluded from further results. It was found that the five zones listed above are the only ones with on average more than 9 persons per household.



(A) *Box-plot of the SAE's of household attributes with different methods.*



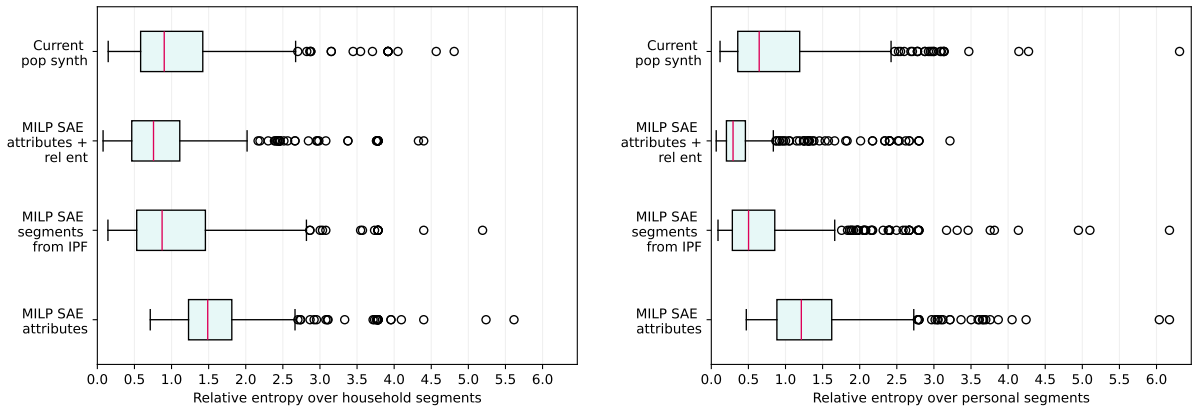
(B) *Box-plot of the SAE's of personal attributes with different methods.*

FIGURE 4.4: *Box-plots of the fit on margins (SAE of attributes) of different population synthesis methods when applied to the zones in the main study area in the Zwolle project. Here, the results of the zones where the average number of persons per households is too high are removed.*

What is interesting to see in Figure 4.4, is that the current population synthesizer performs the worst of the four methods on the SAE for both personal and household attributes. This may be, because in the SNET step, individuals are selected based on probabilities and until the exact number of inhabitants is reached. It does not consider any other attributes.

Furthermore, we see that solving the MILP that minimises the SAE over attribute totals performs best for the metrics on the SAE over attributes, as is to be expected since this is the only goal of this MILP. The MILPs that minimise the SAE over attribute totals and relative entropy, and minimise the SAE over segment totals from IPF score in between the current population synthesizer and the basis MILP that only minimises the SAE over attribute totals.

Even though the MILP that minimises the SAE over attribute totals performs best for the metrics on the SAE over attributes, it performs the worst for the metrics on the relative entropy, see Figure 4.5. This can also be expected, as this MILP does nothing to minimise the relative entropy.



(A) *Box-plot of the relative entropy of household segments with different methods.*

(B) *Box-plot of the relative entropy of personal segments with different methods.*

FIGURE 4.5: *Box-plots of the results on the unicity metric of different population synthesis methods when applied to the zones in the main study area in the Zwolle project.*

When looking at the relative entropy for household segments in Figure 4.5a, we see that the MILP SAE attributes & relative entropy appears to perform a bit better than the current population syntheziser and the MILP SAE segments. We see more distinction in Figure 4.5b, where the MILP SAE attributes & relative entropy clearly performs the best over the four methods. Here, the MILP SAE segments appears to perform a bit better than the current population syntheziser as its box is closer to zero than the box of the current population syntheziser. That the current population syntheziser and the MILP SAE segments perform similar on the relative entropy can easily be explained, as they both use IPF, which minimises the relative entropy. The other steps (iNNLS and SNET in the current population synthesizer and solving the MILP) do nothing to minimise the relative entropy further.

From the results above, one might say that the MILP SAE attributes & relative entropy performs quite well and may be a good method to consider for population synthesis. However, due to the complexity of the MILP, the run times are quite high for this model, as can be seen in Figures 4.6 and 4.7. Given the relative large run times, in these figures, a logarithmic scale is used on the vertical axis.

Finally, we have a look at the car availability in Figure 4.8. We expect the number of car passengers to decrease in the synthetic populations resulting from the MILPs. There are a lot of zones where this is the case, however not for all zones. This may be because the synthetic populations found using each method could be very different from each other when looking at the composition of the selected households.

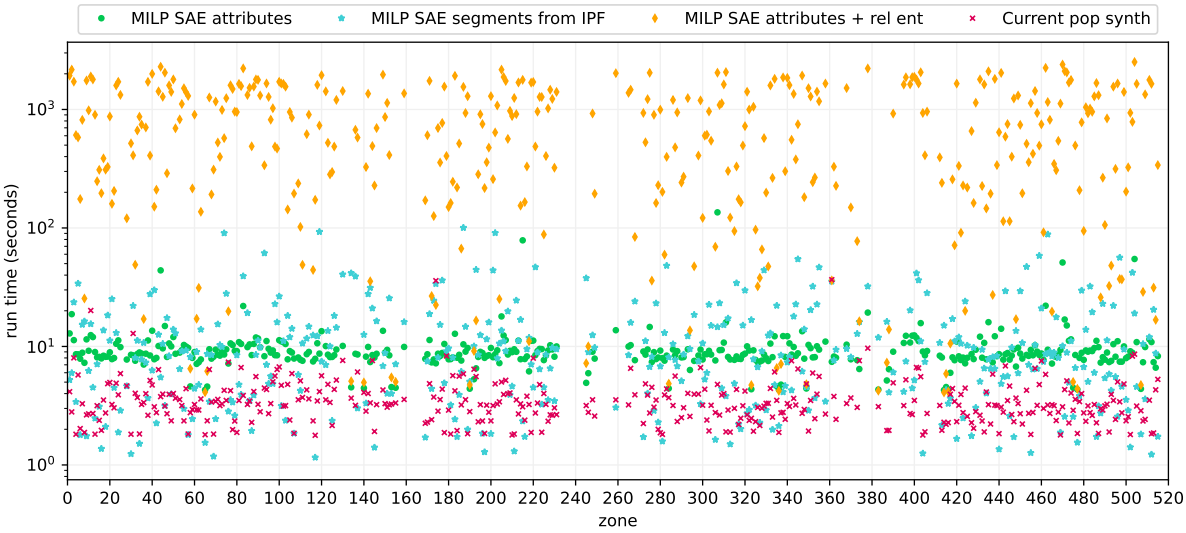


FIGURE 4.6: *Plot of the run times for each zone with different methods on a logarithmic scale.*

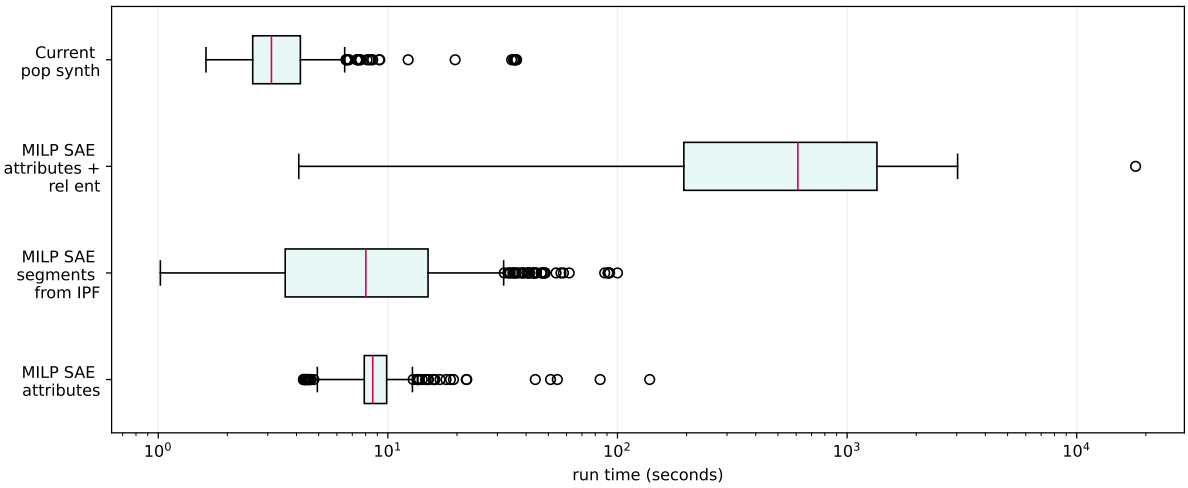
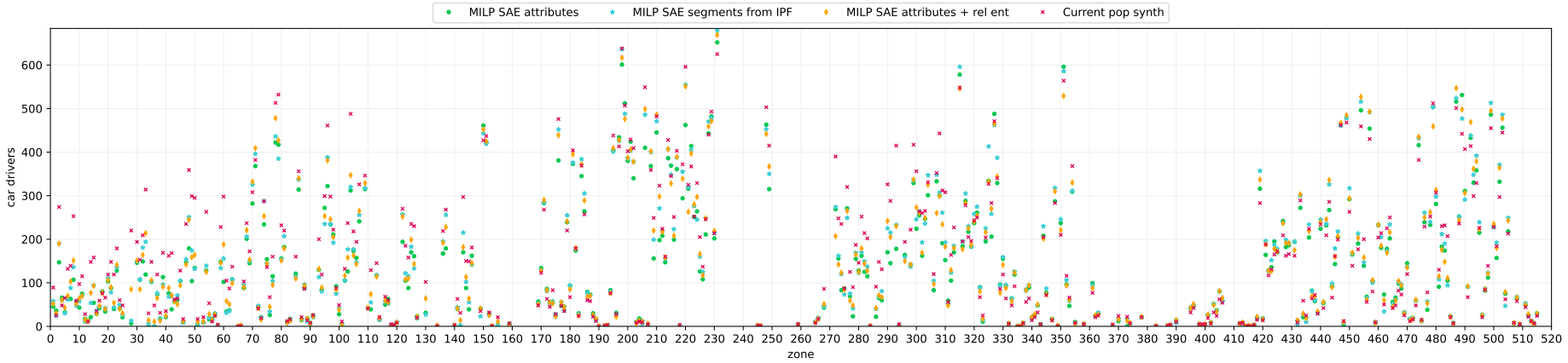
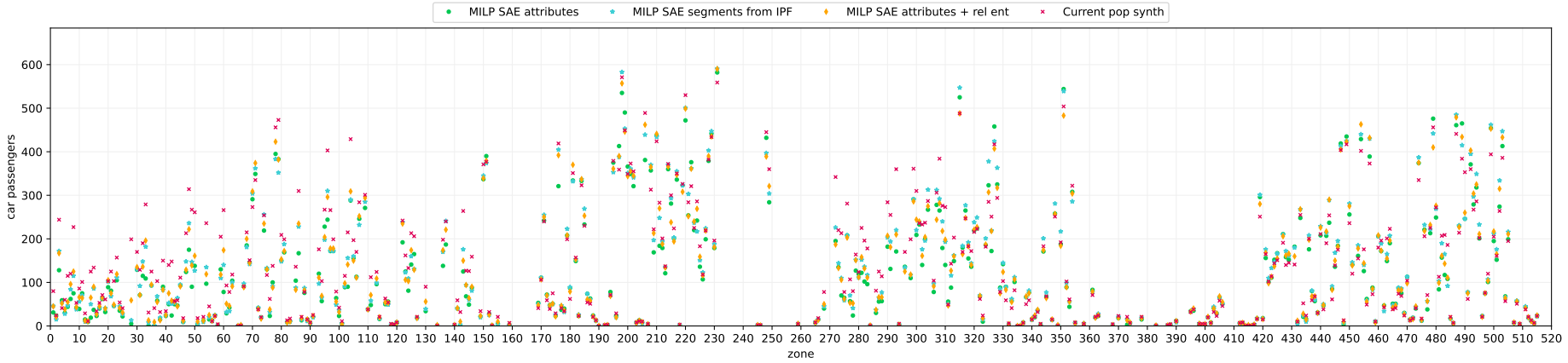


FIGURE 4.7: *Box-plot of the run times using different methods for population synthesis applied to the zones in the main study area in the Zwolle project on a logarithmic scale.*



(A) *Number of possible car drivers.*



(B) *Number of possible car passengers.*

FIGURE 4.8: *Number of possible car drivers and passengers in the zones in the main study area in the synthetic populations resulting from the different methods.*

4.1.2 Influence and outer area

When the methods are applied on zones that are part of the influence area of the Zwolle model, the methods behave very similar compared to the main study area. The same holds for the outer area. We see that, compared to the zones in the main study area, the run time for the larger zones in the influence and outer area are generally larger. This is to be expected, as the size (number of inhabitants and households) of the zones increase as we get further away from Zwolle. Also the standardised fit on margins is smaller with the larger zones, that is because the absolute errors are averaged out more. This also yields for the relative entropy of the unicity metric The plots corresponding to the synthetic populations in the influence and outer area can be found in Appendix B.

4.2 Sensitivity analysis

To test the sensitivity of the model, we add noise to certain attribute categories. The attribute categories that we (separately) add noise to are: number of cars in households, and social participation. The categories are selected based on (1) the importance of the attributes in the decisions individuals make in the rest of the model, and (2) the reliability of the input data/data gathering of the attribute totals. The number of cars has high importance and does not have the desired reliability. The social participation is the most important personal attribute category in the choice models that follow the population synthesizer, and also has low reliability.

For each attribute, a standard normally distributed noise component was added and scaled to 10% of the actual attribute total. We do this so that the noise is proportional to the original values of the attribute totals. Any negative attribute totals are truncated to zero, as we cannot have a negative number of persons or households. The resulting attribute totals are not integerised, because in determining the reference synthetic population with MILP SAE segments we also used continuous values (since the IPF outcome in continuous). Therefore, it is not appropriate to round the values here if they were not rounded in previous calculations.

We conduct the sensitivity analysis on only a couple of zones to allow for a detailed discussion of the results, rather than performing a superficial analysis across all zones. The zones chosen for this analysis are located within the main study area, as these zones have the greatest influence on travel behaviour within the region. The selection criteria for the zones include their size (the number of individuals and households) and the runtime required for calculations. Since the analysis involves adding random noise multiple times to assess variability, it is important to select zones where the model can quickly converge to a solution for each run. The zones that we apply the discussed noise to, are listed in Table 4.2.

The MILP SAE attributes + relative entropy exhibits a considerable runtime for zone 198. This zone has been included due to its larger size and comparatively lower runtime with the other MILP models.

For each zone, attribute category and MILP, we conduct 100 sensitivity analysis runs using different draws for the noise for each combination of method and zone (except MILP SAE attributes + relative entropy for zone 198), and determine the distance between the synthetic population with the noisy totals and the synthetic population of the reference. Values closer to zero are desired as we model the effect of noise, and thus do not want large changes in the

TABLE 4.2: *Zones selected for analysing sensitivity.*

Zone	Inhabitants	Households	Runtime (seconds) for reference population		
			MILP SAE attributes	MILP SAE segments	MILP SAE attributes + relative entropy
16	73	38	7.92	2.50	5.88
20	216	119	8.93	5.54	25.42
198	1258	427	8.30	5.34	2394.06

resulting synthetic population.

The results for the different zones turn out to be very similar, also both noise categories give comparable results. Therefore, in this results section, we focus on the first run with noise added to the number of cars in the households in zone 16.

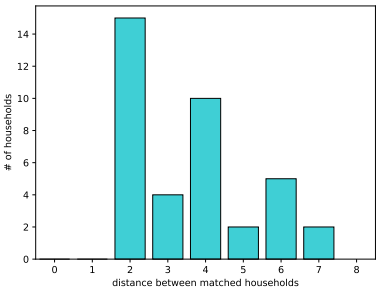
The result we get from one run with noise is which households (from MPN) are in the synthetic population of the zone. We compare this to the households in the reference synthetic population. Each household in the reference set is assigned to a household in the run with noise, such that the total distance between the two household sets is minimised.

For example, for the MILP SAE attributes, the first run with noise added to the number of cars in households in zone 16 gives the following assignment of households between the reference and 1st run with noise:

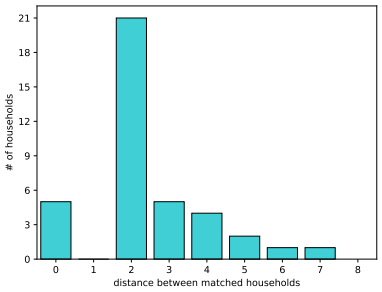
- Household 0 in reference $\rightarrow$ household 2 in the run with noise (distance 2)
- Household 1 in reference $\rightarrow$ household 7 in the run with noise (distance 2)
- Household 2 in reference $\rightarrow$ household 8 in the run with noise (distance 2)
- Household 3 in reference $\rightarrow$ household 9 in the run with noise (distance 2)
- Household 4 in reference $\rightarrow$ household 10 in the run with noise (distance 2)
- ...
- Household 37 in reference $\rightarrow$ household 25 in the run with noise (distance 7)

The total distance of the assignment standardised by the change in input (eq. (3.21)) is 95.96. On average, this comes down to a change of 2.5 segments per one household change in the input.

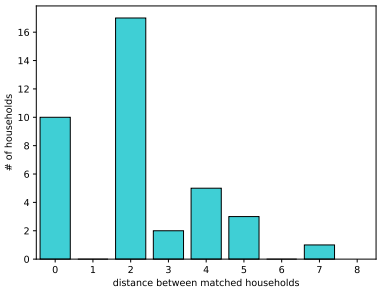
The distance between assigned households for the first run with noise added to the number of cars in households is visualised for each MILP in Figure 4.9. We see that the distribution of the distances shifts closer to zero when looking at the histograms from left to right. This indicates that incorporating relative entropy makes that there are more households that are both in the reference and the run with noise. This effect is bigger with the MILP SAE attributes + relative entropy compared to the MILP SAE segments.



(A) *MILP SAE attributes.*



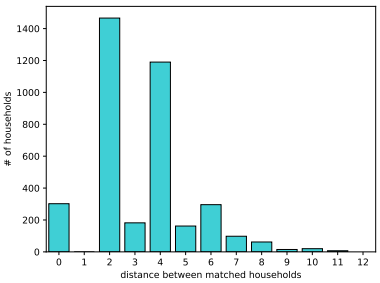
(B) *MILP SAE segments.*



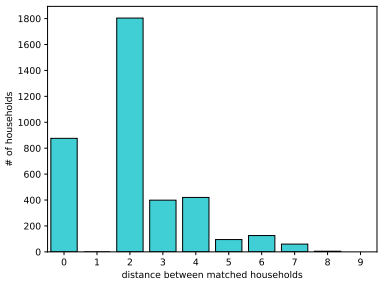
(C) *MILP SAE attributes + relative entropy.*

FIGURE 4.9: *Histograms of the distance between matched households of the 1st run with noise on the number of cars in households for zone 16.*

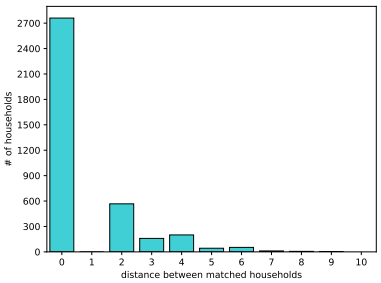
The above results are from the distances between assigned households for one run with noise. These results are also obtained for all other runs with noise. In Figure 4.10, we see the distance between assigned households in all 100 runs cumulatively. We see similar results here as we had for the first run. With MILP SAE attributes there is the least overlap in households between the reference and the run with noise. With MILP SAE segments, this is already better, as is to be expected since relative entropy is incorporated via the IPF step that is done first. When the relative entropy is minimised in the MILP directly (with MILP SAE attributes + relative entropy), there are even more households that are both in the reference and in the runs with noise.



(A) *MILP SAE attributes.*



(B) *MILP SAE segments.*



(C) *MILP SAE attributes + relative entropy.*

FIGURE 4.10: *Histograms of the distance between matched households in all 100 runs with noise on the number of cars in households in zone 16.*

Also the standardised total distance for all 100 runs can be visualised. In Figure 4.11, we do this again for zone 16 with noise added to the attribute totals of the number of cars in the households. We see that the distances are smaller for the MILP SAE attributes + relative entropy compared to the MILP SAE segments. The MILP SAE attributes results in the largest standardised distances. This is as expected since there is nothing in this MILP that tries to get the results of the run with noise close to the reference. Adding relative entropy to the objective of the MILP results in a synthetic population for the run with noise, that overlaps the most with the reference population compared to the other MILPs.

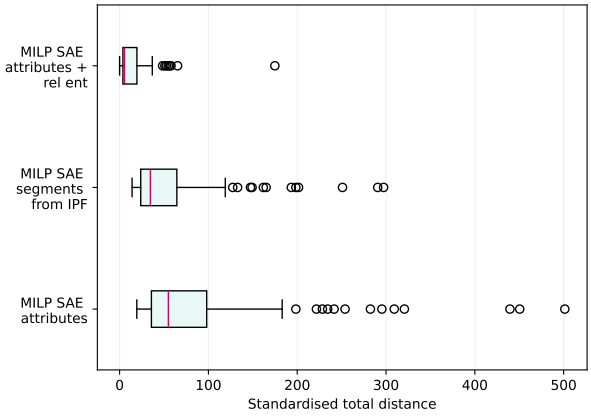


FIGURE 4.11: *Box-plot of the standardised distance between the households in the reference and the scenario with noise added to the attribute totals of the number of cars in the households in zone 16.*

Next to the distance between the households in the reference and scenario, we also look at the change in the input of attribute totals and the change in the attribute totals of the synthetic populations. Again for zone 16, we plot the total absolute change per attribute category in the 100 runs in a box-plot, see Figure 4.12. We see that indeed only the input attribute totals regarding the number of cars in households are changed.

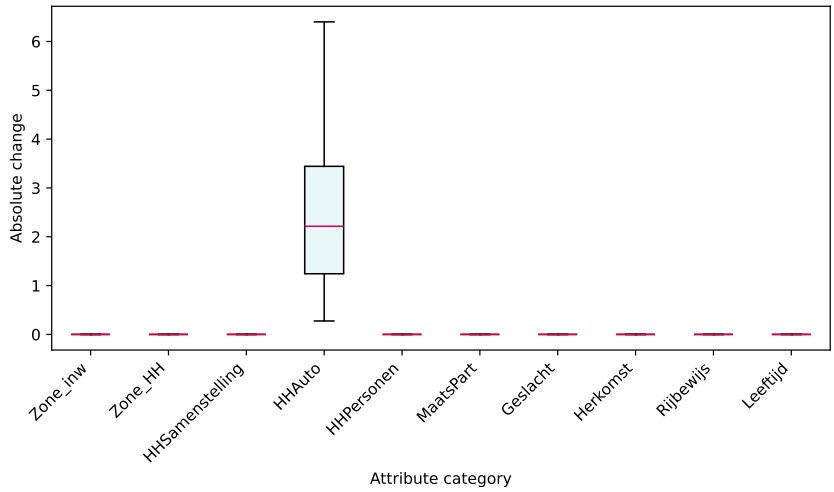
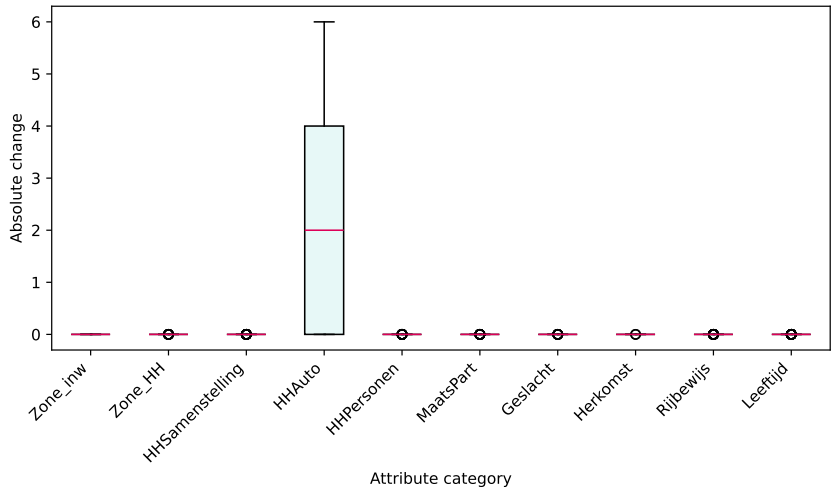
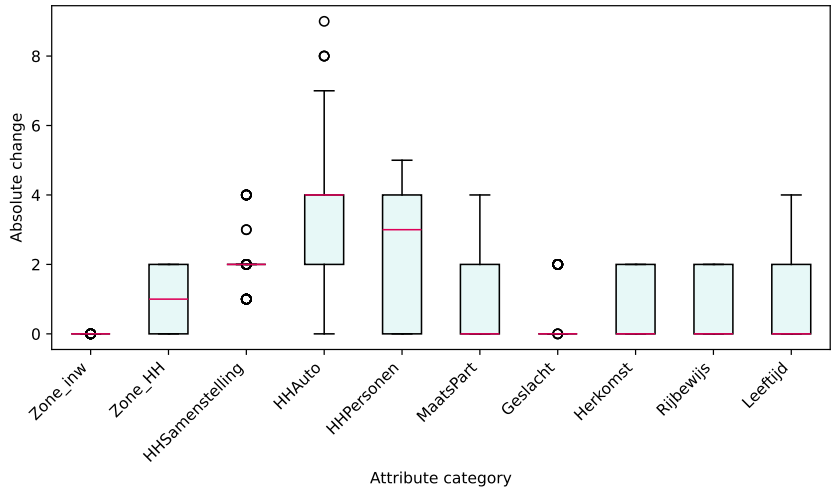


FIGURE 4.12: *Total absolute change in the input attribute totals of zone 16 when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*

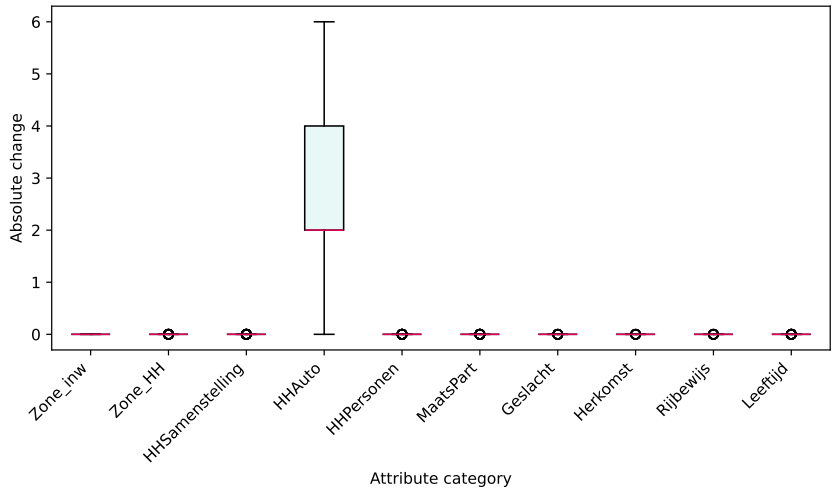
Looking at the change in attribute totals in the resulting synthetic populations gives Figure 4.13. We see that with the MILP SAE attributes and the MILP SAE attributes + relative entropy there are no changes in the output attribute totals in attribute categories other than the one we applied noise to. With MILP SAE segments, we do see changes in other attribute categories. This may be explained by the fact that we first do the IPF step in which the segment totals are calculated before we use the MILP. The additional step may cause more variation in output attribute totals.



(A) *MILP SAE attributes.*



(B) *MILP SAE segments.*



(C) *MILP SAE attributes + relative entropy.*

FIGURE 4.13: *Total absolute change in the output attribute totals of zone 16 with different MILPs when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*

The summarising results for the other zones and noise category can be found in Appendix C. Here, we see similar results as we see here with noise added to the attribute totals of the number of cars in households in zone 16.

4.3 Stability analysis

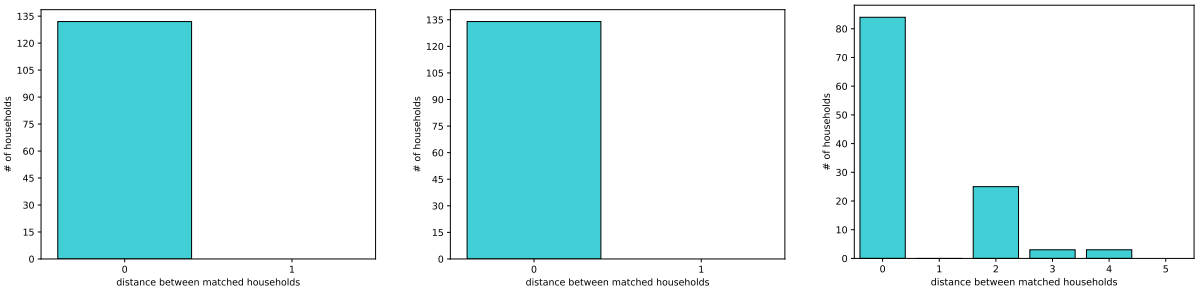
To analyse the stability when running a scenario, we apply the scenario of 180 additional individuals in each zone, regardless of the current size of the zones. The 180 individuals are added to the attribute total using the distribution over the attributes in the reference, rounded to the nearest integer.

We analyse the results for the same three zones as were used in the sensitivity analysis in the previous section. When running a scenario, we use the methods with the proposed in Section 3.3.3.

With the added term to keep the weights of the reference and scenario the same as much as possible, we see that indeed the households that were already in the reference are also in the scenario, as all distances are zero. This is also because the added households and persons are distributed over the attributes according to the distribution of the reference. In these results, the distances of households in the scenario that are not matched to a household in the reference are zero.

Only for the MILP SAE attributes + relative entropy we see that we do not get the households that were already in the reference synthetic population. There is no clear explanation for this, except that the relative entropy term in the objective may cause the MILP to find solutions that are closer to the "most likely" solution. We see this as well for zone 20 and 198, see Appendix D. This is unusual, as the sensitivity analysis showed that incorporating relative entropy into the objective produces scenario solutions that partially include the same households as in the reference, despite not constraining the weights to be as close as possible to the reference weights.

Just as in the previous section, we look at the results of zone 16 here. The figures regarding zone 20 and 198 can be found in Appendix D



(A) *MILP SAE attributes.*

(B) *MILP SAE segments.*

(C) *MILP SAE attributes + relative entropy.*

FIGURE 4.14: *Histograms of the distance between matched households for the scenario in zone 16.*

5 Concluding remarks

This study aimed to develop method(s) to create synthetic populations that adhere to the household consistency requirement. Household consistency is an important aspect in synthetic populations, because decisions in the following steps of a travel demand model may depend on decisions made by others in the household. Having household consistency in the synthetic population give us the option to not only determine car availability better but also include other household decision, such as MaaS plans, the need for a babysitter for the kids, or working from home.

We have looked at population synthesis methods across three categories: synthetic reconstruction, combinatorial optimisation, and statistical learning. And we looked closely at the current population synthesizer (which falls in the synthetic reconstruction category) to see where the issues with household consistency arise. Next to household consistency, we also had the requirements of complying with the marginal attribute totals. Furthermore, compatibility (all changes in outputs are related to changes in input) was important, this was achieved by incorporating relative entropy to find the most likely synthetic population.

The current population synthesizer has household consistency until the iNNLS step, but with continuous weights. The household consistency is not maintained in the final integerisation step. Just rounding the weights would preserve household consistency, but gives very bad results looking at the marginal attribute totals, because there are many weights close to zero. To gain a clear understanding of the population synthesis problem, we formulated it as a mathematical program. Adding the constraint that the weights must be integer and non-negative ensures household consistency, the main goal. By minimising the standardised absolute error of the attribute totals, the resulting synthetic population complies with the known totals as much as possible.

We extended the mathematical program to find the most likely synthetic population by also minimising the relative entropy in the objective of the mathematical program. This caused non-linearity in the mathematical program, for which we found a way to linearise it. This added a lot of variables to the mathematical program. An alternative method to incorporate relative entropy minimisation is by using the results from the first step of the current population synthesizer: iterative proportional fitting. It is known that IPF minimises the relative entropy, while complying to the marginal attribute totals. So the idea is that by minimising the absolute difference to the segment totals obtained from IPF, we both minimise the relative entropy and the absolute error to the attribute totals.

It turned out that solving the (basis) MILP performed better then expected with respect to computational requirements. Therefore we decided to continue the path of solving the MILP directly and incorporated the additional requirements in the MILPs as well. Adding relative entropy to the objective function made it more difficult to solve because we added a lot of variables to make it linear. Indirectly minimising the relative entropy, by using the segment totals obtained in the IPF step, performed a lot better looking at the computational requirements. The other performance metrics for these two MILPs were similar. In the sensitivity analysis, the MILP SAE attributes + relative entropy performed the best. However, in the stability analysis, it produced some unexpected results for which we could not identify a clear explanation. Although the MILP SAE segments approach performed slightly worse in the sensitivity analysis,

it generates synthetic populations significantly faster than the MILP SAE attributes combined with relative entropy. Therefore, the MILP minimising the SAE of the segment totals is advised to include household consistency in the synthetic populations.

Discussion

For all population synthesizer applications in this report, we used data from MPN. The MPN data contains household and the persons in these household. However, there are only 2193 unique complete household compositions in the data, while there most likely are a lot more feasible combinations of segments to create complete households. We know that the MPN questionnaire is mostly filled in by "standard" household and does not contain a lot of households that are more unusual. For example, there are very few student houses in the data. These can be households with many similar persons, looking at the segment of these persons. This means that in zones (certain neighbourhoods of a city) where there live many students, the synthetic population is not very representative of the actual population.

Zones containing facilities such as elderly homes, for example, pose challenges in generating a representative synthetic population and make it difficult, if not impossible, to align with the marginal attribute totals. In the marginal data, all the persons living in the elderly home are in the same household. The same is the case with jails. This give situations as found in Table 4.1, where in one zone there are 127 persons living in one household. There are of course more zones with elderly homes or jails, but in larger zones the average number of persons per household is weighted out by the more normal household sizes. Also, the marginal totals are (manually) checked beforehand to remove impossibilities such as a zone having more households than persons. In future implementation, there should be an input validation step that checks for obvious input consistency errors such as on average too many persons per household in a zone.

The sensitivity and stability analysis were not done in comparison to the current population synthesizer. The reason behind this is that in the current synthetic populations, there is no household consistency and therefore the reference and scenario populations cannot be compared to each other in the same way as suggested in this report. Because of this it is also difficult to say what the desired bounds for the performance metrics are.

References

[1] L. Brederode, T. Hardt, and B. Rijksen, "Development of a microscopic tour based demand model without statistical noise," European Transport Conference, 2020.

[2] A. G. Wilson, "The Use of Entropy Maximising Models, in the Theory of Trip Distribution, Mode Split and Route Split," *Journal of Transport Economics and Policy*, vol. 3, no. 1, pp. 108–126, 1960.

[3] P. K. Kranenbarg, "Reducing the statistical noise in a microscopic tour-based travel demand model," 2021.

[4] B. F. Yamogo, P. Gastineau, P. Hankach, and P. O. Vandanjon, "Comparing Methods for Generating a Two-Layered Synthetic Population," *Transportation Research Record*, vol. 2675, pp. 136–147, Jan. 2021.

[5] B. F. Yamogo, P. O. Vandanjon, P. Gastineau, and P. Hankach, "Generating a Two-Layered Synthetic Population for French Municipalities: Results and Evaluation of Four Synthetic Reconstruction Methods," *Journal of Artificial Societies and Social Simulation*, vol. 24, Mar. 2021.

[6] N. Fournier, E. Christofa, . Arun, P. Akkinepally, . Carlos, and L. Azevedo, "Integrated population synthesis and workplace assignment using an efficient optimization-based person-household matching method," *Transportation*, vol. 48, no. 2, pp. 1061–1087, 2021.

[7] K. Zhu, K. Liu, J. Liu, Y. Shi, X. Li, H. Zou, H. Du, and L. Yin, "Generating synthetic population for simulating the spatiotemporal dynamics of epidemics," *PLoS Computational Biology*, vol. 20, Feb. 2024.

[8] I. Mahmood, N. Bishop, A. Calinescu, M. Wooldridge, and I. Zachos, "A multi-objective combinatorial optimisation framework for large scale hierarchical population synthesis," European Simulation and Modelling Conference 2023, 2023.

[9] K. Chapuis and P. Taillandier, "A brief review of synthetic population generation practices in agent-based social simulation," in *SSC 2019, Social Simmulation Conference*, (Mainz, Germany), 2019.

[10] P. Ye and F.-Y. Wang, "Basic Population Synthesis," *Parallel Population and Parallel Human*, pp. 17–54, June 2023.

[11] L. Sun, A. Erath, and M. Cai, "A hierarchical mixture modeling framework for population synthesis," *Transportation Research Part B*, vol. 114, pp. 199–212, 2018.

[12] M. Templ, B. Meindl, A. Kowarik, and O. Dupriez, "Simulation of Synthetic Complex Data: The R Package simPop," *Journal of Statistical Software*, vol. 79, no. 10, 2017.

[13] B. P. Y. Loo and W. W. Y. Lam, "A multilevel investigation of differential individual mobility of working couples with children: a case study of Hong Kong," *Transportmetrica A: Transport Science*, vol. 9, no. 7, pp. 629–652, 2013.

[14] M. J. Olde Kalter and K. T. Geurs, "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*, vol. 16, no. 4, pp. 698–712, 2016.

[15] R. Lovelace and D. Ballas, "Truncate, replicate, sample: A method for creating integer weights for spatial microsimulation," *Computers, Environment and Urban Systems*, vol. 41, pp. 1–11, Sept. 2013.

[16] G. Albiston, T. Osman, and D. Brown, "A neural network approach for population synthesis," *Simulation*, 2024.

[17] M. N. Rahman and M. R. Fatmi, "Population Synthesis Accommodating Heterogeneity: A Bayesian Network and Generalized Raking Technique," *Transportation Research Record*, vol. 2677, pp. 41–57, June 2023.

[18] J. Tuccillo, R. Stewart, A. Rose, N. Trombley, J. Moehl, N. Nagle, and B. Bhaduri, "UrbanPop: A spatial microsimulation framework for exploring demographic influences on human dynamics," *Applied Geography*, vol. 151, p. 102844, Feb. 2023.

[19] M. Zhou, J. Li, R. Basu, and J. Ferreira, "Creating spatially-detailed heterogeneous synthetic populations for agent-based microsimulation," *Computers, Environment and Urban Systems*, vol. 91, p. 101717, Jan. 2022.

[20] D. Voas and P. Williamson, "An Evaluation of the Combinatorial Optimisation Approach to the Creation of Synthetic Microdata," *International Journal of Population Geography*, vol. 6, no. 5, pp. 349–366, 2000.

[21] J. E. Abraham, K. J. Stefan, and J. D. Hunt, "Population Synthesis Using Combinatiorial Optimization at Multiple Levels," in *Transportation Research Board 91st Annual Meeting*, (Washington DC, United States), 2012.

[22] K. Harland, A. Heppenstall, D. Smith, and M. Birkin, "Creating realistic synthetic populations at varying spatial scales: A comparative critique of population synthesis techniques," *JASSS*, vol. 15, no. 1, 2012.

[23] M. Birkin, A. Turner, and B. Wu, "A Synthetic Demographic Model of the UK Population: Methods, Progress and Problems," in *36th Annual Conference Regional Science Association International British and Irish Section*, (Jersey, Channel Islands), 2006.

[24] L. Sun and A. Erath, "A Bayesian network approach for population synthesis," *Transportation Research Part C: Emerging Technologies*, vol. 61, pp. 49–62, Dec. 2015.

[25] F. P. Reffel, "Termination of the iterative proportional fitting procedure," *Statistics and Probability Letters*, vol. 92, pp. 59–64, 2014.

[26] I. Csiszár, "I-divergence geometry of probability distributions and minimization problems," *The annals of probability*, vol. 3, no. 1, pp. 146–158, 1975.

[27] F. Willekens, A. Por, and R. Raquillet, "Entropy, multiproportional and quadratic techniques for inferring patterns of migration from aggregate data,[w:] a. rogers (red.) advances in multiregional demography," tech. rep., Research Report RR-81-06, IIASA, Laxenburg, 1981.

[28] D. Chen and R. J. Plemmons, "Nonnegativity constraints in numerical analysis," in *The birth of numerical analysis*, pp. 109–139, World Scientific, 2010.

[29] X. Ye, K. Konduri, R. M. Pendyala, B. Sana, and P. Waddell, "A methodology to match distributions of both household and person attributes in the generation of synthetic populations," in *Transportation Research Board 88th Annual Meeting*, 2009.

[30] Q. Huangfu and J. Hall, "Parallelizing the dual revised simplex method," *Mathematical Programming Computation*, vol. 10, no. 1, pp. 119–142, 2018.

[31] M. Lubin, O. Dowson, J. D. G. Vielma, J. Huchette, B. Legat, and J. Pablo, "JuMP 1.0: Recent improvements to a modeling language for mathematical optimization," *Mathematical Programming Computation*, 2023.

A Metric using number of times each attributes is in a household

The second metric is an alternative for the aforementioned metric. Instead of using the number of times each segment is in a household as a vector, we create a vector containing the number of times each attribute is in a household. Again, let HH^R be the set of households in the reference solution and HH^S the set of households in the scenario solution. Then the distance on the attribute level between household $u \in HH^R$ and $v \in HH^S$ is calculated as follows:

$$\sum_{a \in A^H \cup A^P} |u_a - v_a| \,. \tag{A.1}$$

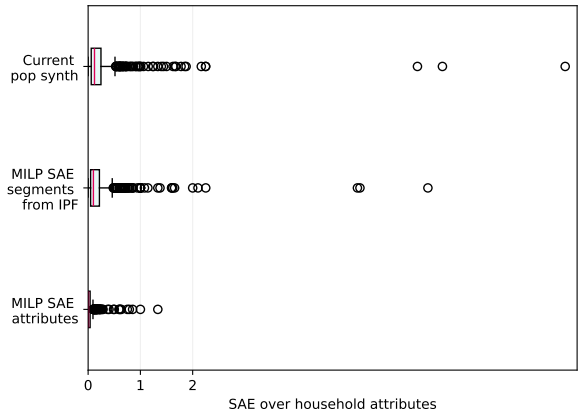
With these distances, we find the best assignment that minimises the total distance between the two sets. If one set contains more households than the other, then the households that are not assigned to another household are not considered in calculating the distance. The best assignment can again be found using the Hungarian algorithm.

This metric is not a good metric for our purpose. It could be that some persons change segment, but the total number of attributes stays the same, see Example 3. Therefore, this metric is not used in our evaluation framework.

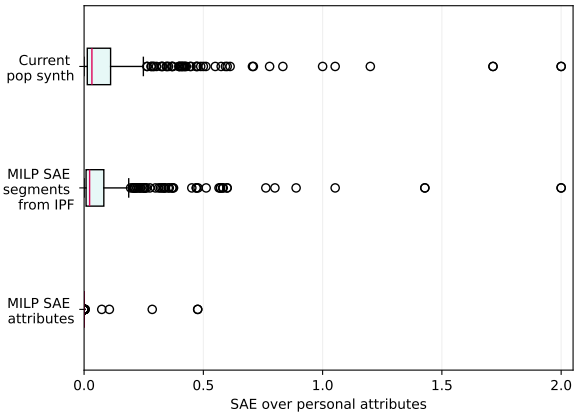
Example 3. In the reference there is a household containing 2 persons: a man with drivers licence and a woman without a drivers licence. In the scenario solution, there could be a similar household with a man without a drivers licence and a woman with a drivers licence. The attribute totals is both these households is the same, so the metric will return 0 difference. But the households are not actually the same.

B Results of reference populations in the influence and outer area

B.1 Influence area

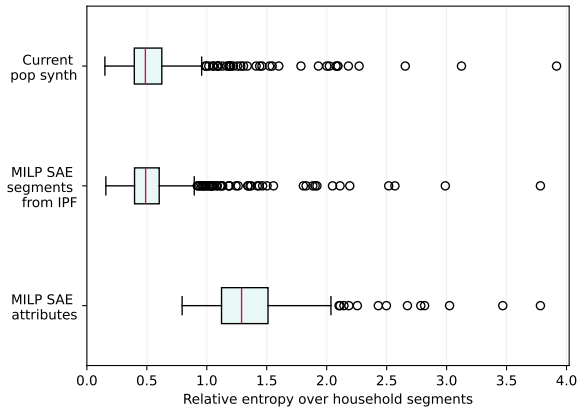


(A) *Box-plot of the SAE's of household attributes with different methods.*

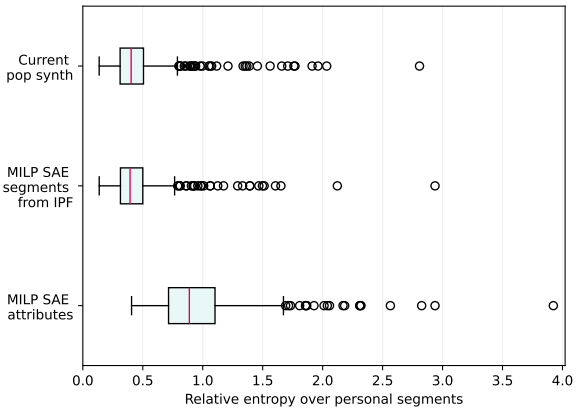


(B) *Box-plot of the SAE's of personal attributes with different methods.*

FIGURE B.1: *Box-plots of the fit on margins (SAE of attributes) of different population synthesis methods when applied to the zones in the influence area in the Zwolle project.*



(A) *Box-plot of the relative entropy of household segments with different methods.*



(B) *Box-plot of the relative entropy of personal segments with different methods.*

FIGURE B.2: *Box-plots of the results on the unicity metric of different population synthesis methods when applied to the zones in the influence area in the Zwolle project.*

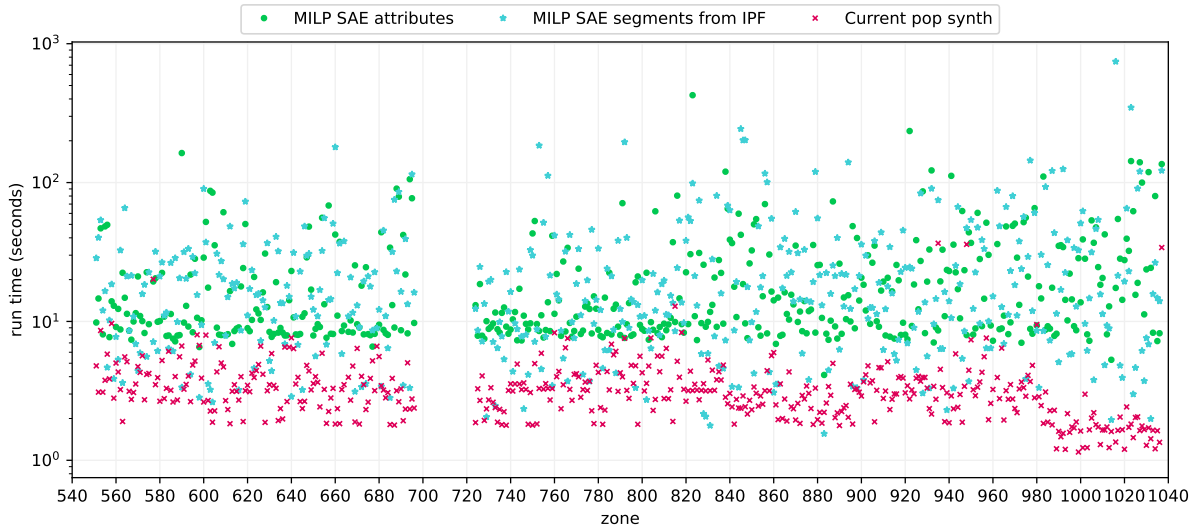
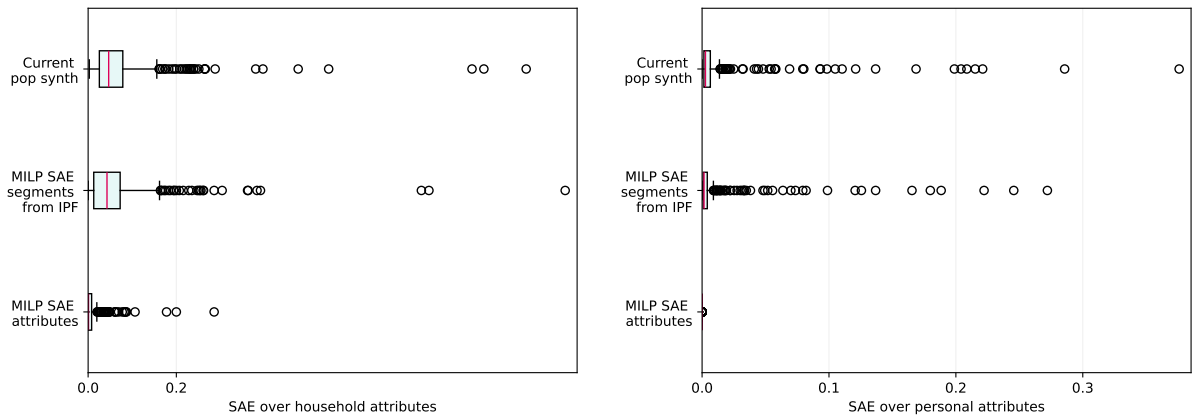


FIGURE B.3: *Plot of the run times for each zone with different methods on a logarithmic scale.*

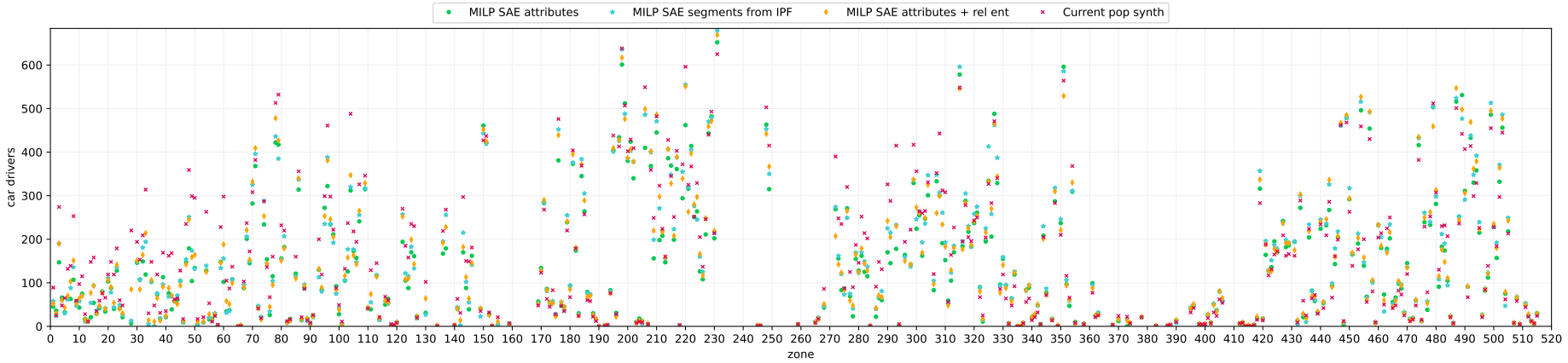
B.2 Outer area



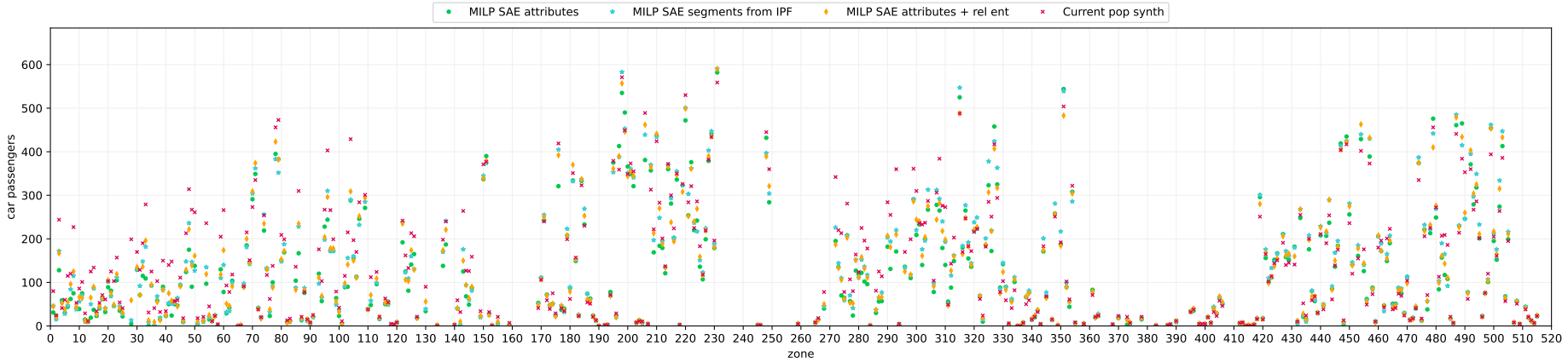
(A) *Box-plot of the SAE's of household attributes with different methods.*

(B) *Box-plot of the SAE's of personal attributes with different methods.*

FIGURE B.5: *Box-plots of the fit on margins (SAE of attributes) of different population synthesis methods when applied to the zones in the outer area in the Zwolle project.*

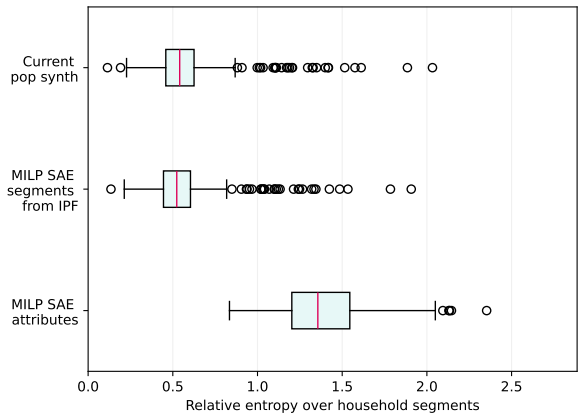


(A) *Number of possible car drivers.*

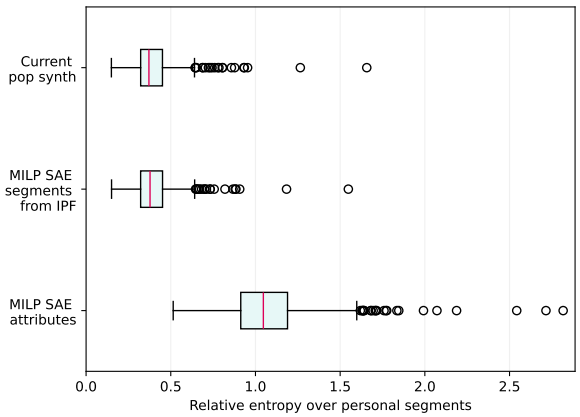


(B) *Number of possible car passengers.*

FIGURE B.4: *Number of possible car drivers and passengers in the zones in the main study area in the synthetic populations resulting from the different methods.*



(A) *Box-plot of the relative entropy of household segments with different methods.*



(B) *Box-plot of the relative entropy of personal segments with different methods.*

FIGURE B.6: *Box-plots of the results on the unicity metric of different population synthesis methods when applied to the zones in the outer area in the Zwolle project.*

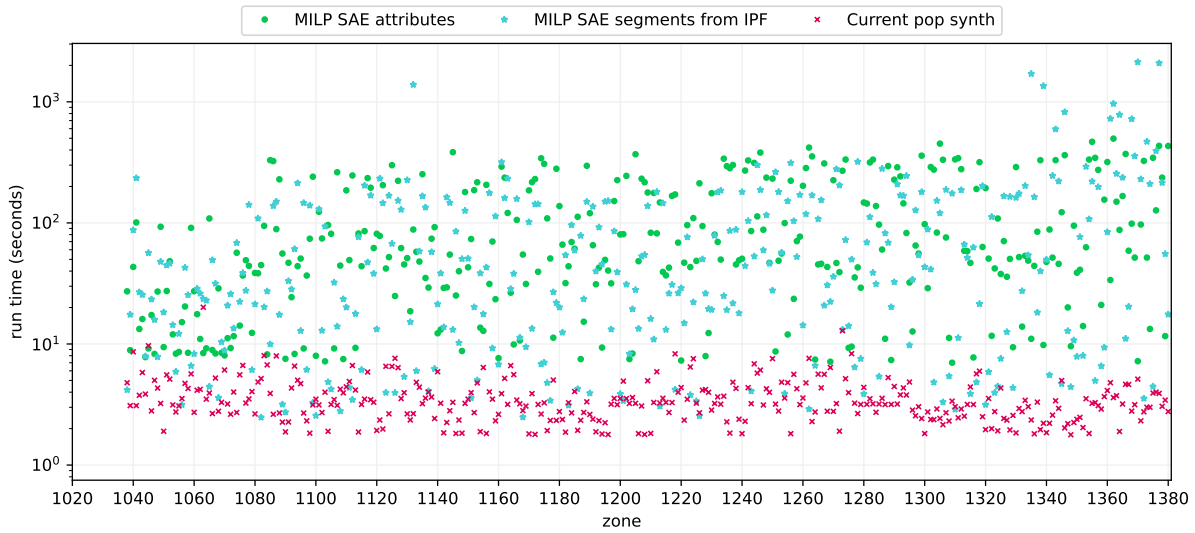
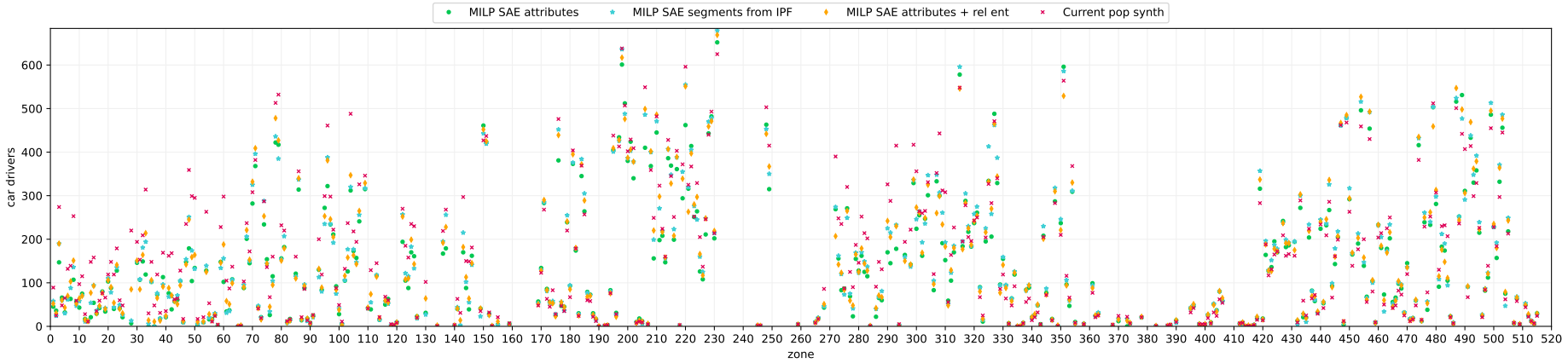
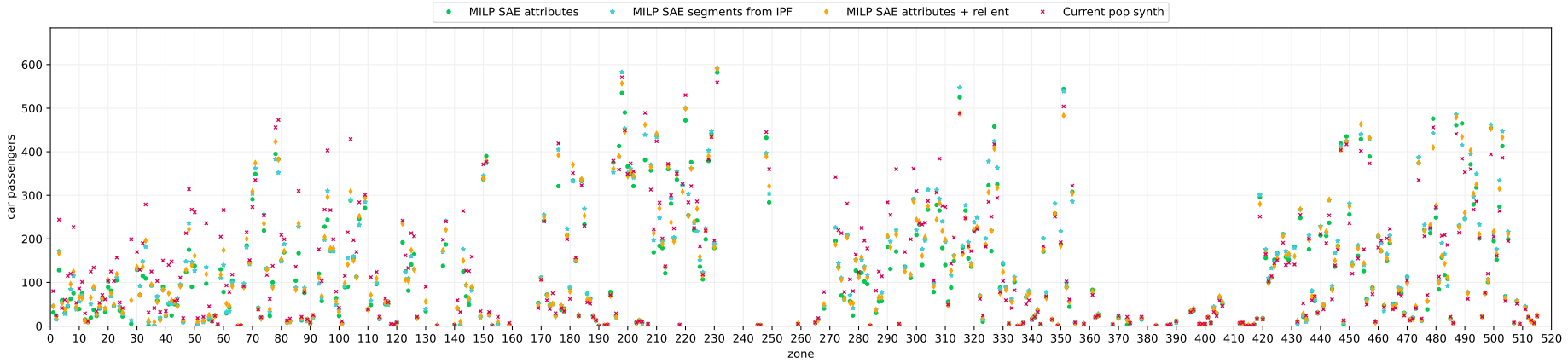


FIGURE B.7: *Plot of the run times for each zone with different methods on a logarithmic scale.*



(A) *Number of possible car drivers.*

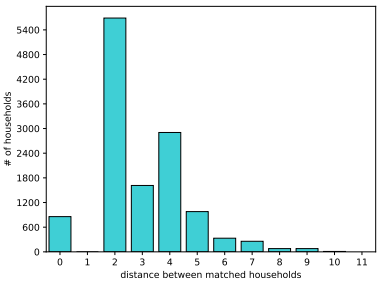


(B) *Number of possible car passengers.*

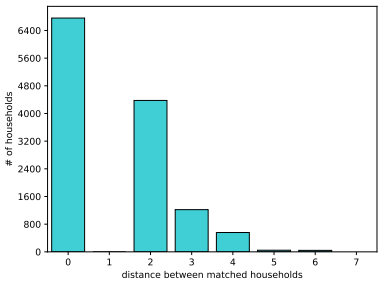
FIGURE B.8: *Number of possible car drivers and passengers in the zones in the main study area in the synthetic populations resulting from the different methods.*

C Additional results of sensitivity analysis

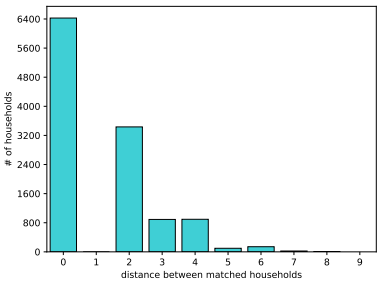
C.1 Distance between households in reference and noise scenarios



(A) *MILP SAE attributes.*

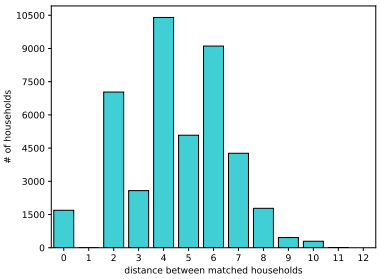


(B) *MILP SAE segments.*

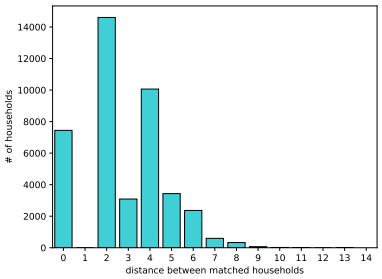


(C) *MILP SAE attributes + relative entropy.*

FIGURE C.1: *Histograms of the distance between matched households in all 100 runs with noise on the number of cars in households in zone 20.*

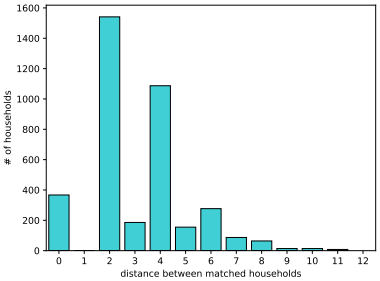


(A) *MILP SAE attributes.*

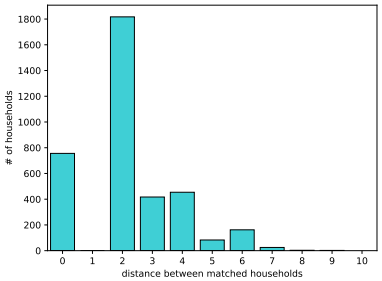


(B) *MILP SAE segments.*

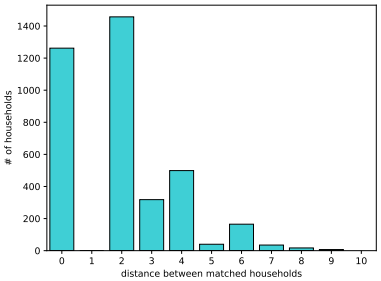
FIGURE C.2: *Histograms of the distance between matched households in all 100 runs with noise on the number of cars in households in zone 198.*



(A) *MILP SAE attributes.*

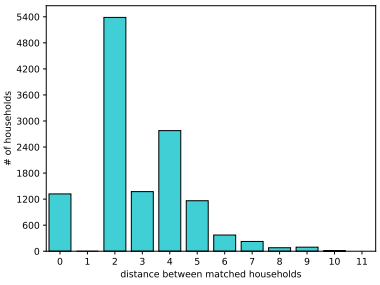


(B) *MILP SAE segments.*

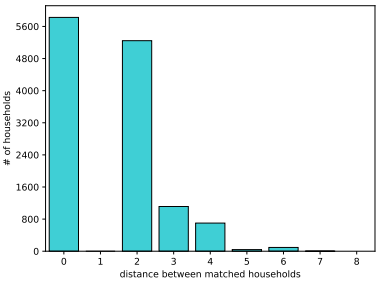


(C) *MILP SAE attributes + relative entropy.*

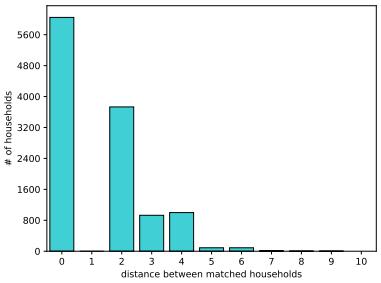
FIGURE C.3: *Histograms of the distance between matched households in all 100 runs with noise on social participation of inhabitants in zone 16.*



(A) *MILP SAE attributes.*

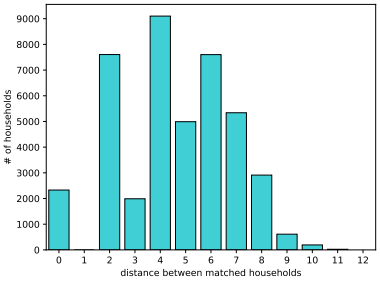


(B) *MILP SAE segments.*

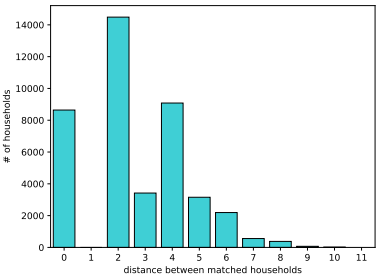


(C) *MILP SAE attributes + relative entropy.*

FIGURE C.4: *Histograms of the distance between matched households in all 100 runs with noise on social participation of inhabitants in zone 20.*



(A) *MILP SAE attributes.*



(B) *MILP SAE segments.*

FIGURE C.5: *Histograms of the distance between matched households in all 100 runs with noise on social participation of inhabitants in zone 198.*

C.2 Standardised total distance between households in reference and noise scenarios

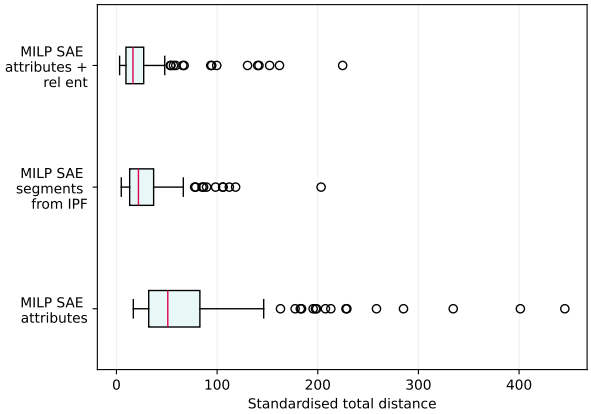


FIGURE C.6: *Box-plot of the standardised distance between the households in the reference and the scenario with noise added to the attribute totals of the number of cars in the households in zone 20.*

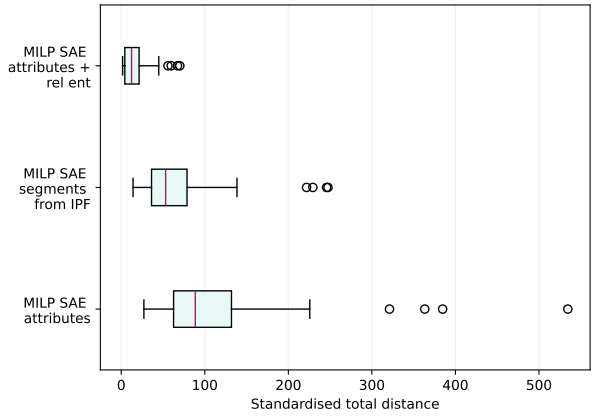


FIGURE C.7: *Box-plot of the standardised distance between the households in the reference and the scenario with noise added to the attribute totals of the number of cars in the households in zone 198.*

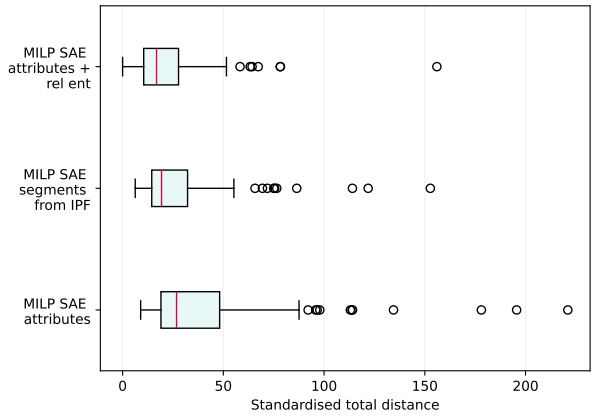


FIGURE C.8: *Box-plot of the standardised distance between the households in the reference and the scenario with noise added to the attribute totals of social participation in zone 16.*

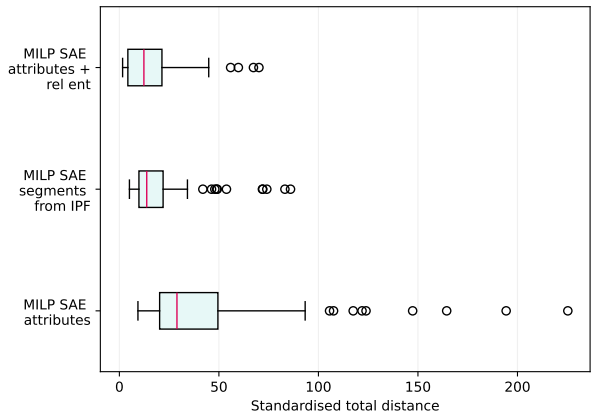


FIGURE C.9: *Box-plot of the standardised distance between the households in the reference and the scenario with noise added to the attribute totals of social participation in zone 20.*

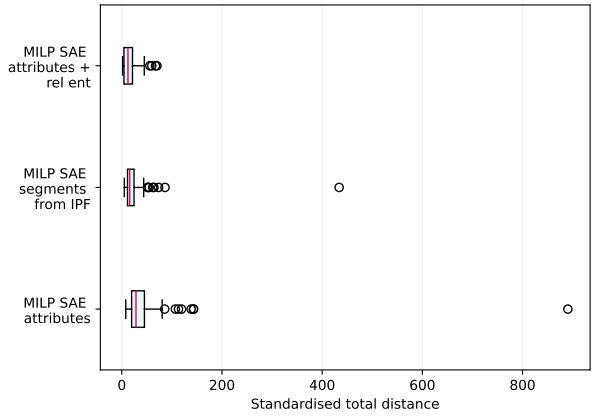


FIGURE C.10: *Box-plot of the standardised distance between the households in the reference and the scenario with noise added to the attribute totals of social participation in zone 198.*

C.3 Input change compared to output change

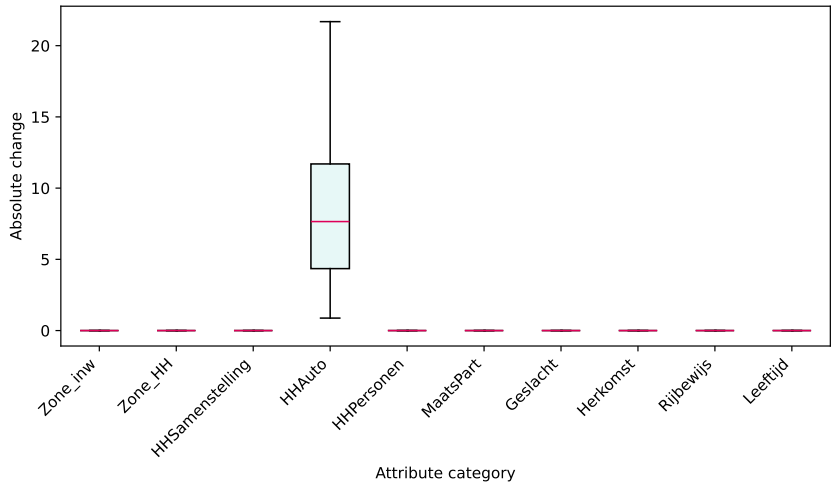
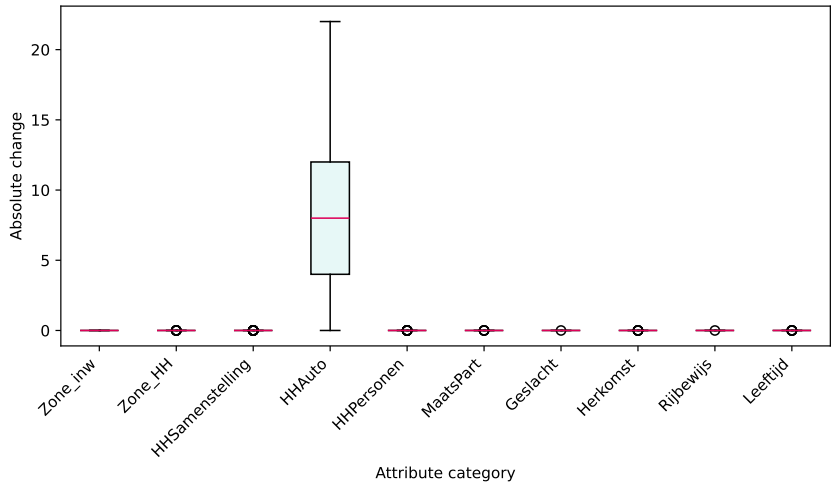
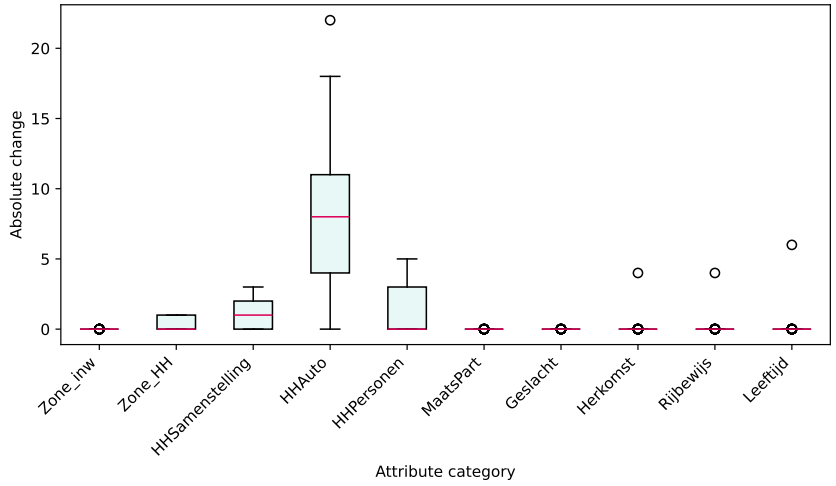


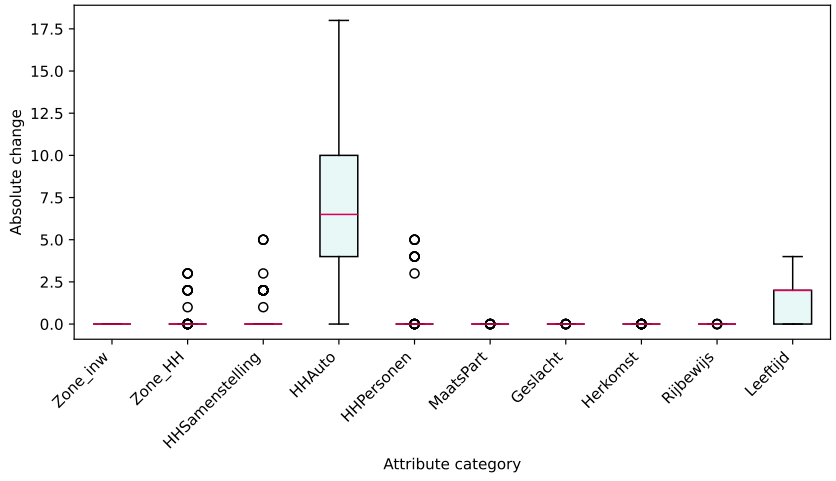
FIGURE C.11: *Total absolute change in the input attribute totals of zone 20 when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*



(A) *MILP SAE attributes.*



(B) *MILP SAE segments.*



(C) *MILP SAE attributes + relative entropy.*

FIGURE C.12: *Total absolute change in the output attribute totals of zone 20 with different MILPs when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*

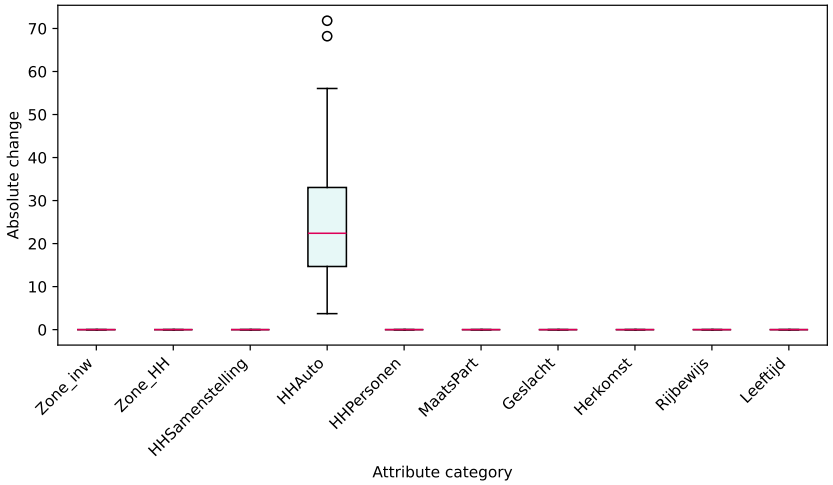
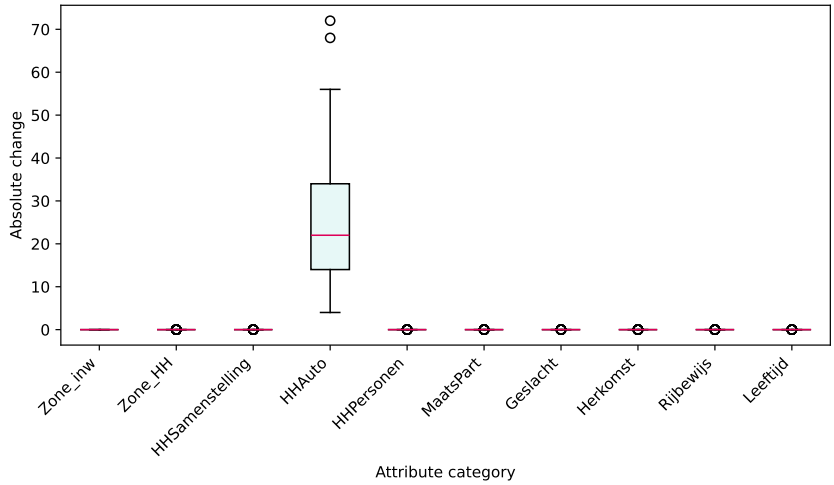
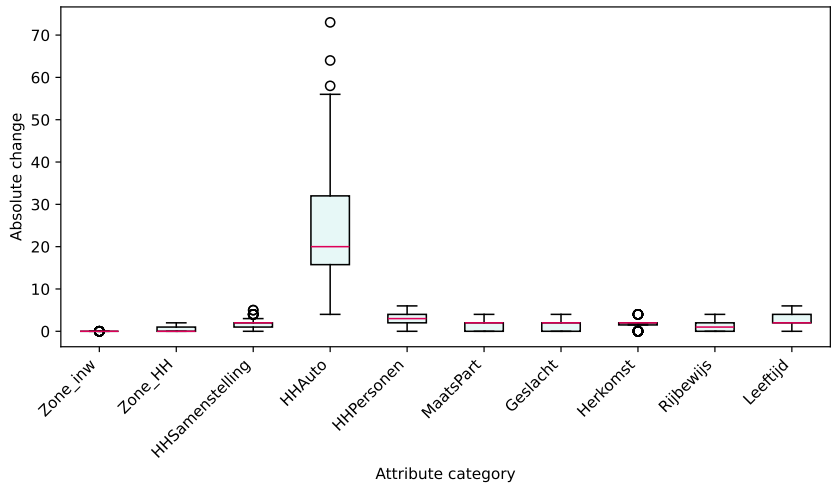


FIGURE C.13: *Total absolute change in the input attribute totals of zone 198 when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*



(A) *MILP SAE attributes.*



(B) *MILP SAE segments.*

FIGURE C.14: *Total absolute change in the output attribute totals of zone 198 with different MILPs when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*

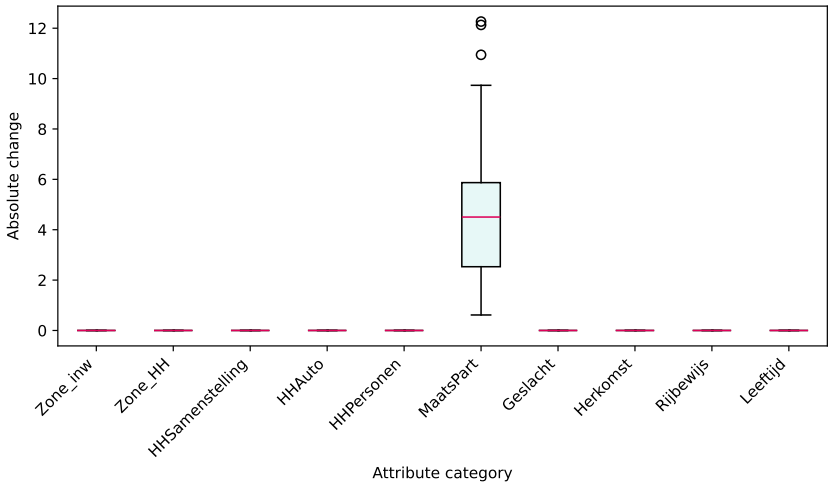
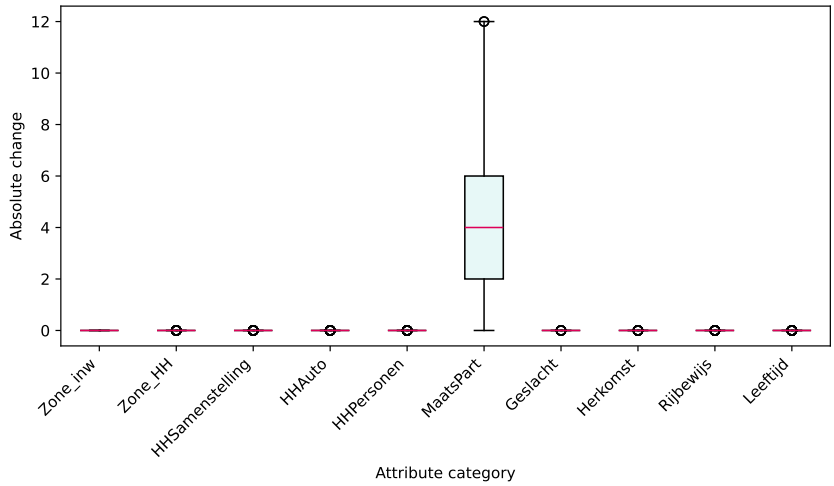
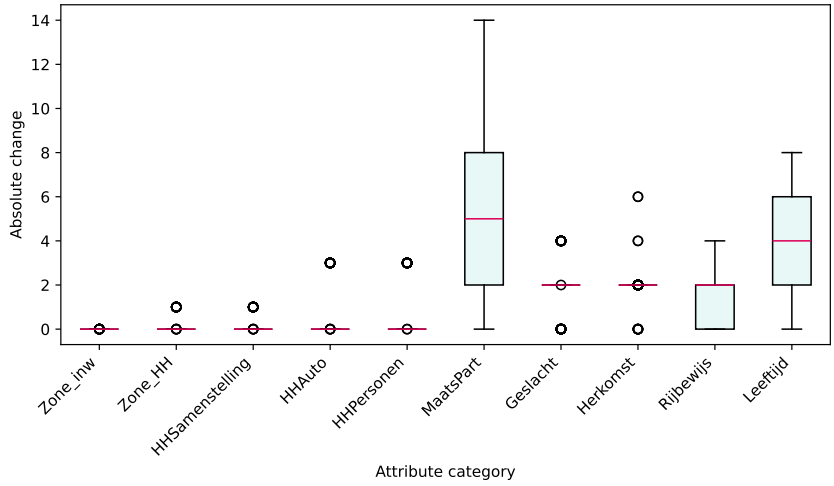


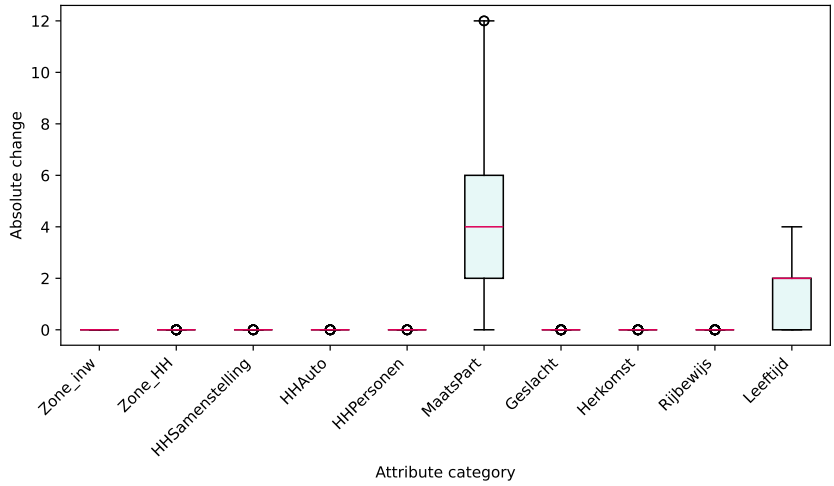
FIGURE C.15: *Total absolute change in the input attribute totals of zone 16 when noise is applied to the attribute totals of social participation in 100 runs.*



(A) *MILP SAE attributes.*



(B) *MILP SAE segments.*



(C) *MILP SAE attributes + relative entropy.*

FIGURE C.16: *Total absolute change in the output attribute totals of zone 16 with different MILPs when noise is applied to the attribute totals of social participation in 100 runs.*

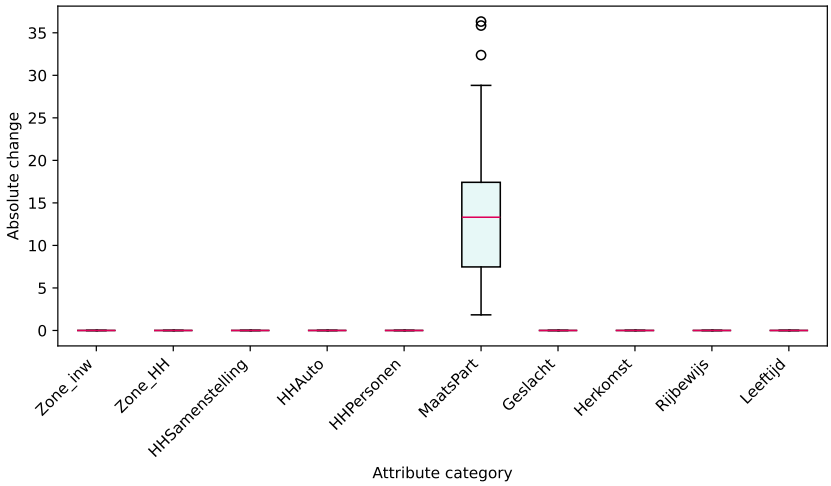
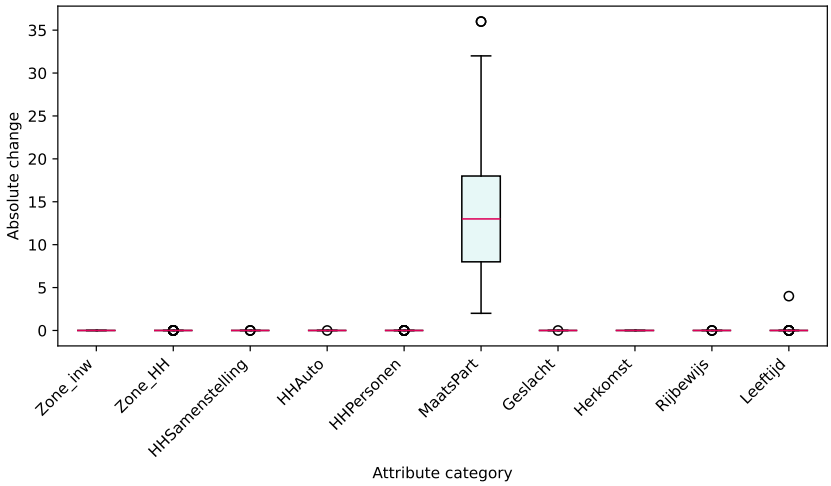
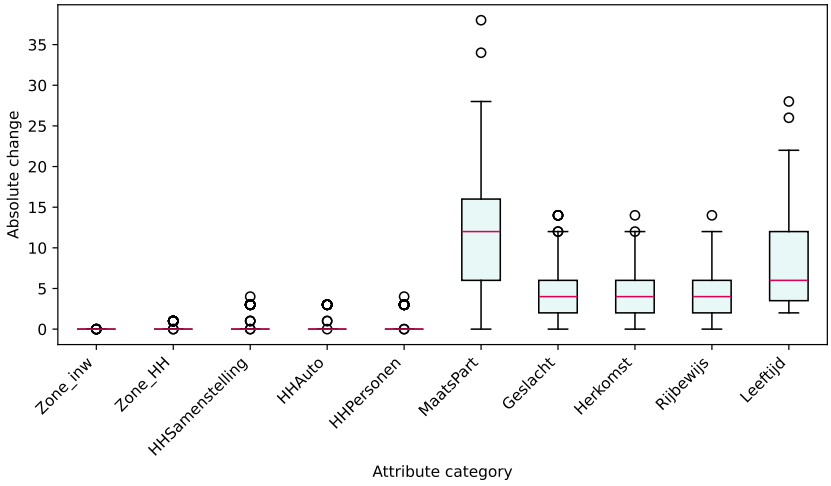


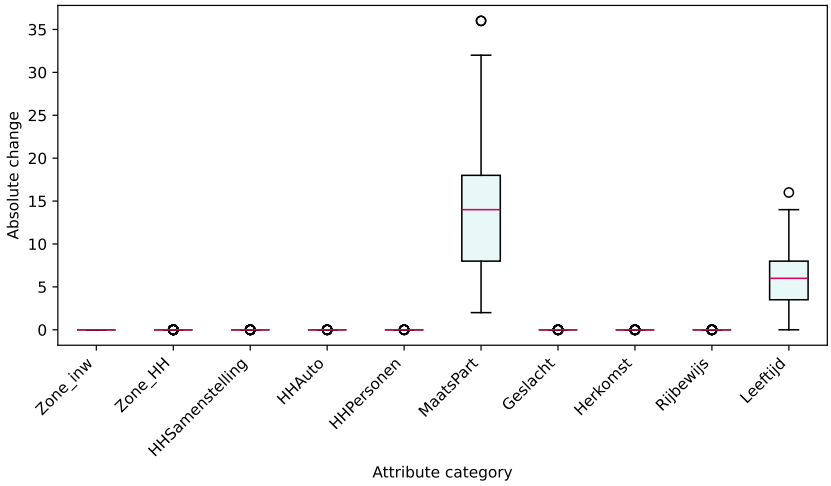
FIGURE C.17: *Total absolute change in the input attribute totals of zone 20 when noise is applied to the attribute totals of social participation in 100 runs.*



(A) *MILP SAE attributes.*



(B) *MILP SAE segments.*



(C) *MILP SAE attributes + relative entropy.*

FIGURE C.18: *Total absolute change in the output attribute totals of zone 20 with different MILPs when noise is applied to the attribute totals of social participation in 100 runs.*

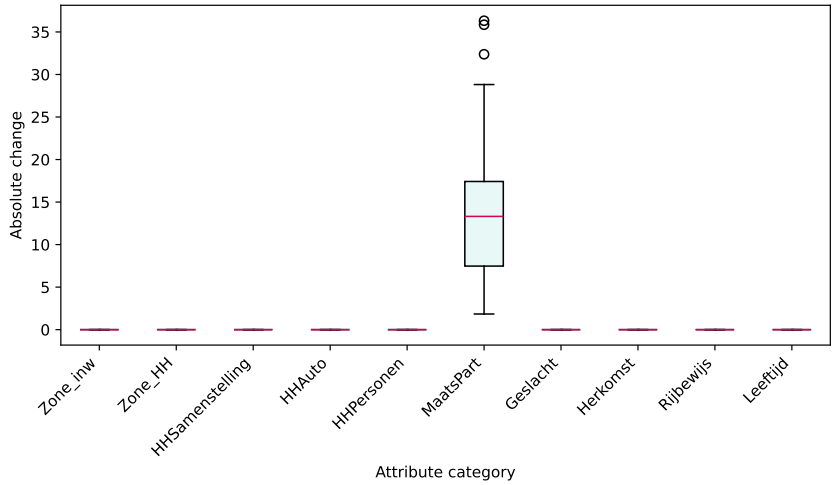
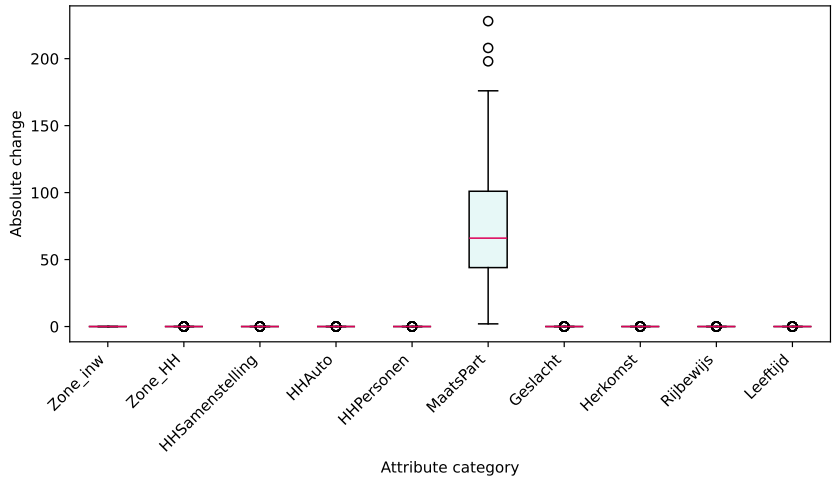
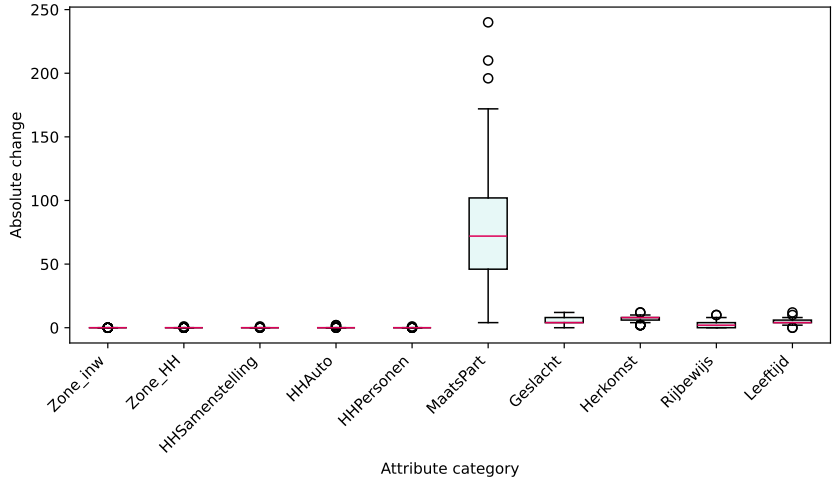


FIGURE C.19: *Total absolute change in the input attribute totals of zone 20 when noise is applied to the attribute totals of social participation in 100 runs.*



(A) *MILP SAE attributes.*

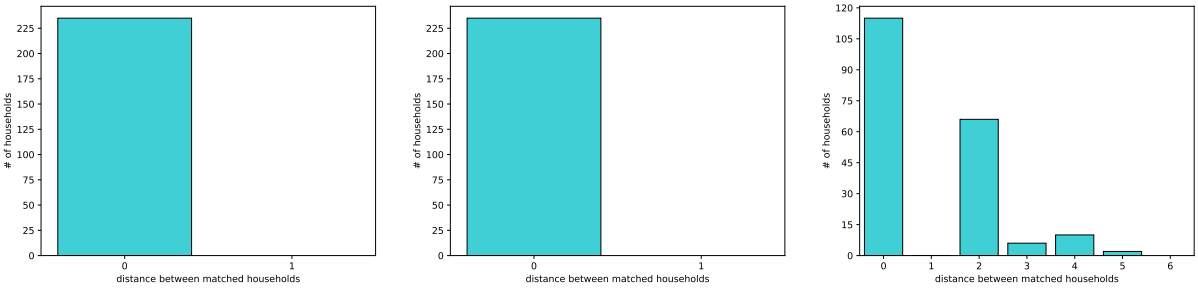


(B) *MILP SAE segments.*

FIGURE C.20: *Total absolute change in the output attribute totals of zone 198 with different MILPs when noise is applied to the attribute totals of social participation in 100 runs.*

D Additional results of stability analysis

D.1 Distance between households in reference and scenario

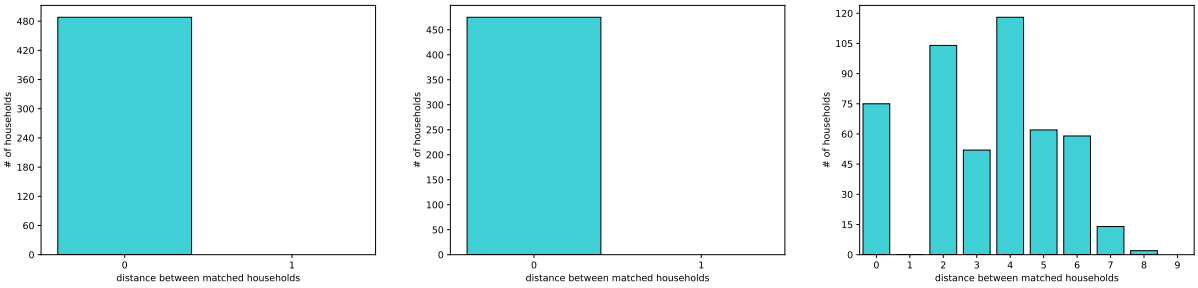


(A) *MILP SAE attributes.*

(B) *MILP SAE segments.*

(C) *MILP SAE attributes + relative entropy.*

FIGURE D.1: *Histograms of the distance between matched households for the scenario in zone 20.*



(A) *MILP SAE attributes.*

(B) *MILP SAE segments.*

(C) *MILP SAE attributes + relative entropy.*

FIGURE D.2: *Histograms of the distance between matched households for the scenario in zone 198.*

D.2 Input change compared to output change

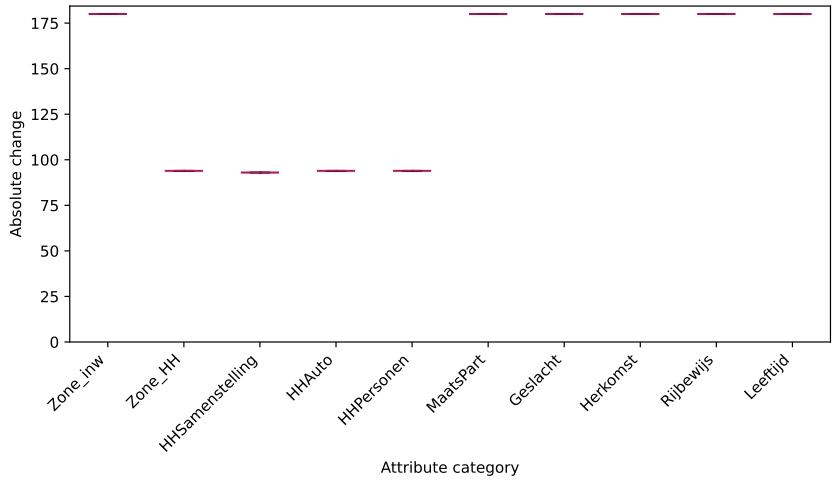


FIGURE D.3: *Total absolute change in the input attribute totals of zone 16 when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*

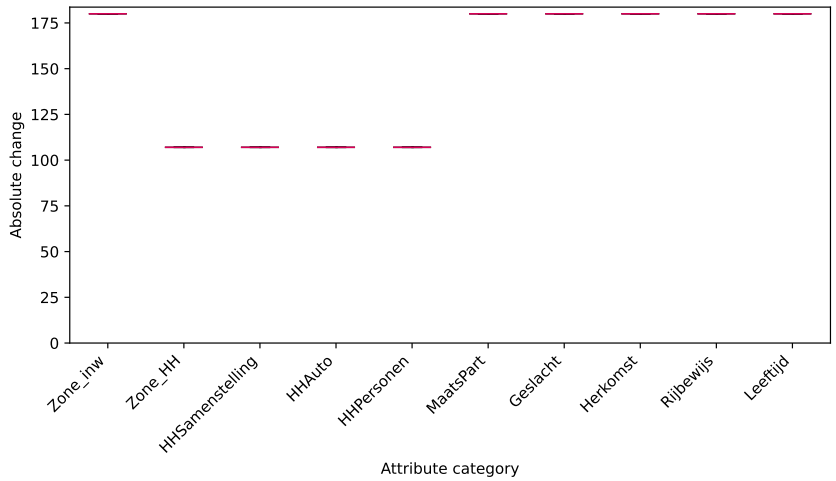


FIGURE D.4: *Total absolute change in the input attribute totals of zone 20 when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*

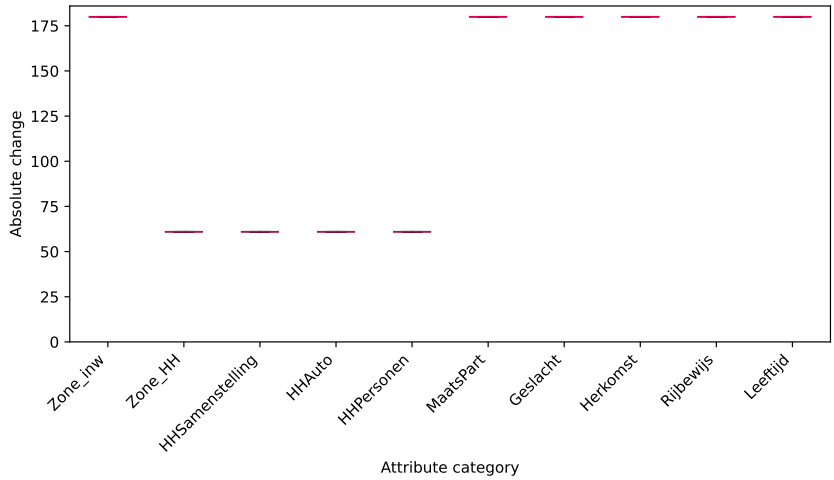
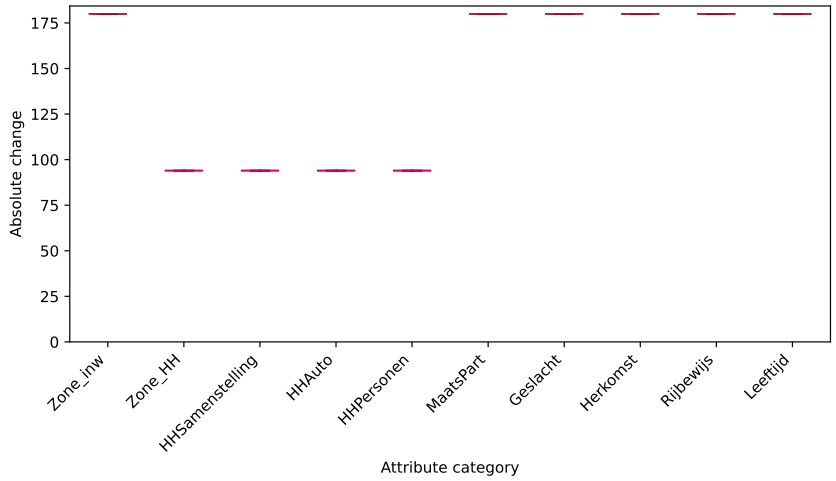
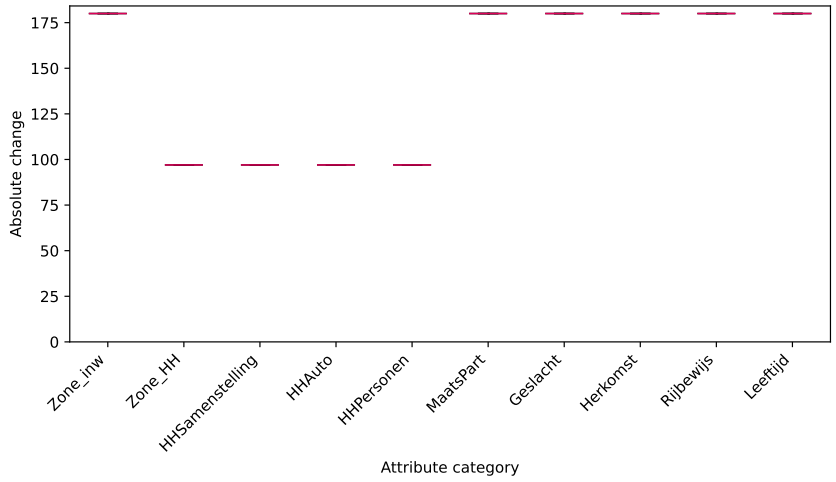


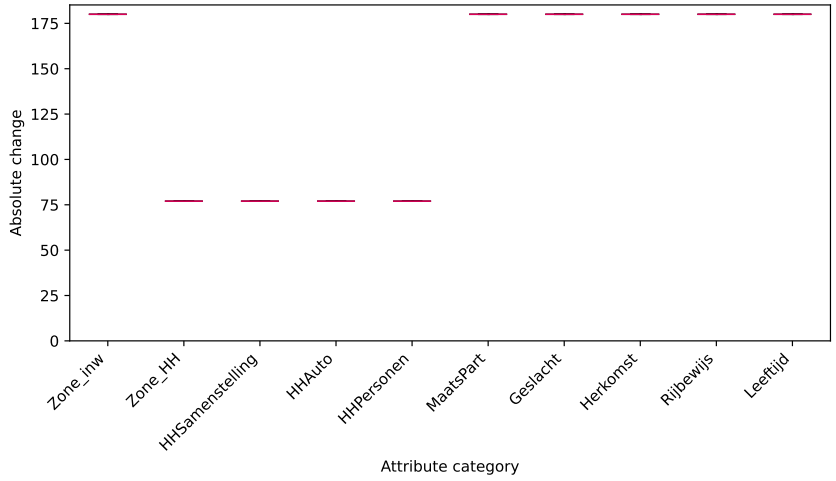
FIGURE D.5: *Total absolute change in the input attribute totals of zone 198 when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*



(A) *MILP SAE attributes.*

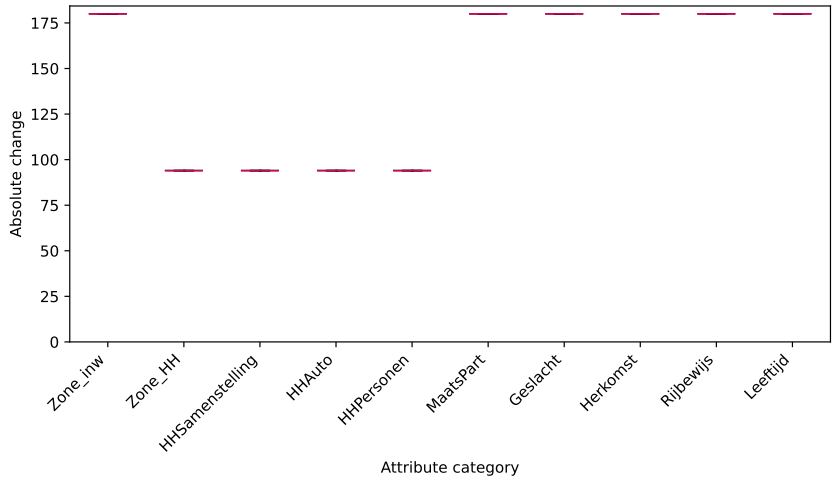


(B) *MILP SAE segments.*

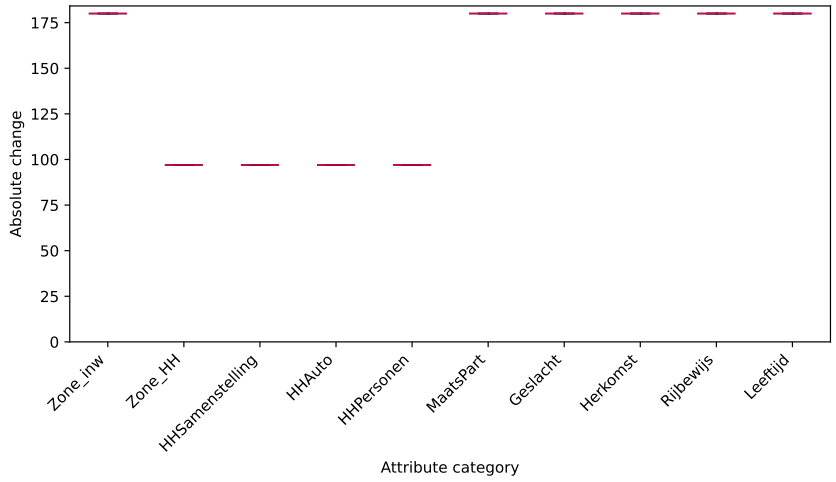


(C) *MILP SAE attributes + relative entropy.*

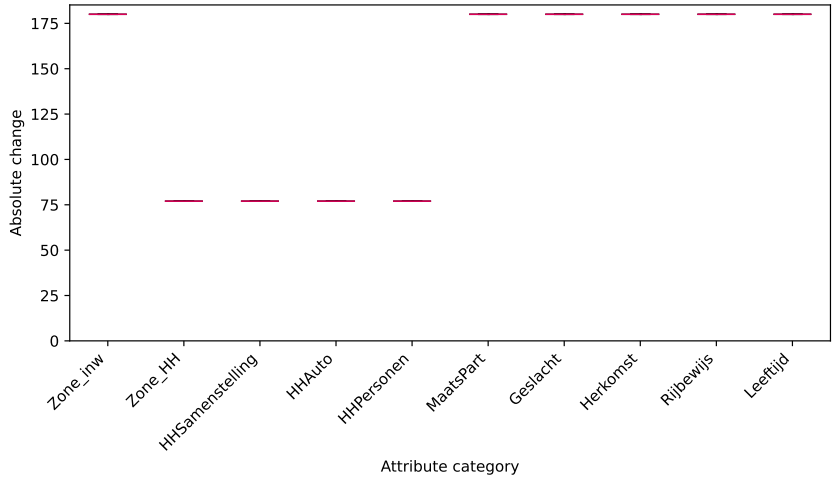
FIGURE D.6: *Total absolute change in the output attribute totals of zone 16 with different MILPs when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*



(A) *MILP SAE attributes.*

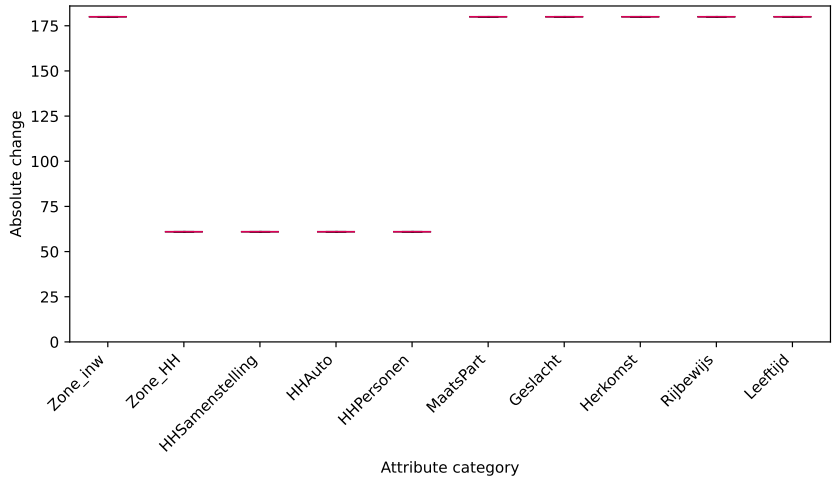


(B) *MILP SAE segments.*

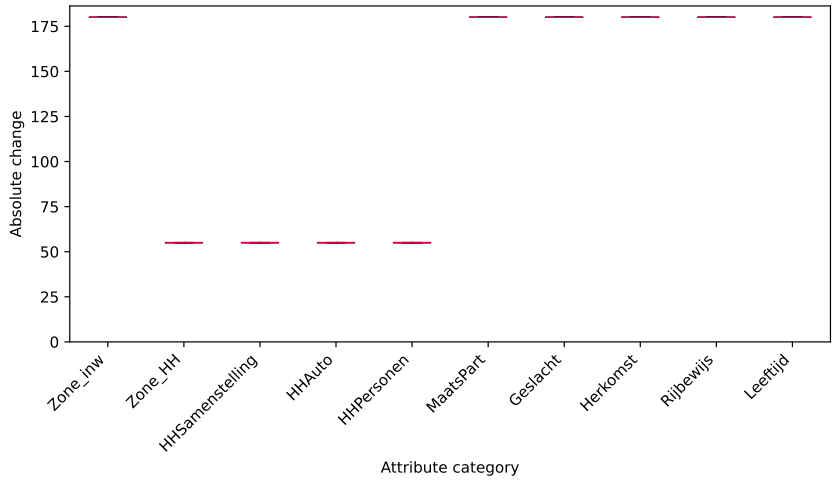


(C) *MILP SAE attributes + relative entropy.*

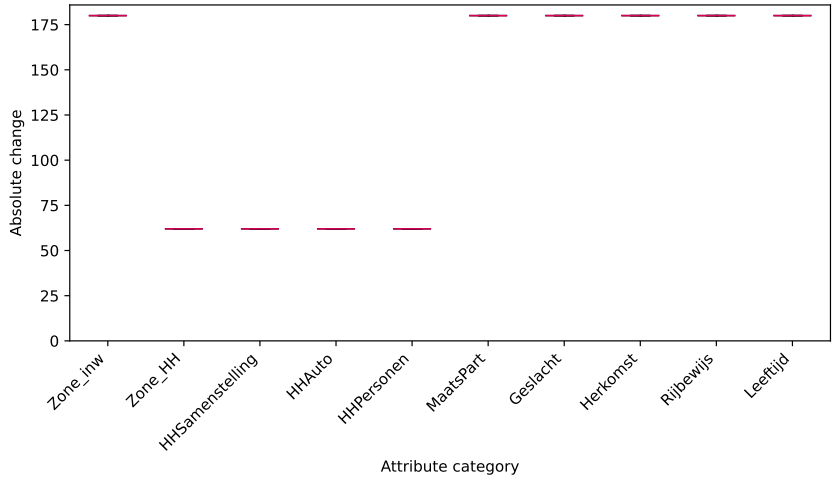
FIGURE D.7: *Total absolute change in the output attribute totals of zone 16 with different MILPs when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*



(A) *MILP SAE attributes.*



(B) *MILP SAE segments.*



(C) *MILP SAE attributes + relative entropy.*

FIGURE D.8: *Total absolute change in the output attribute totals of zone 198 with different MILPs when noise is applied to the attribute totals of the number of cars in the households in 100 runs.*