Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 280 (2026) 269–276

Procedia Computer Science

www.elsevier.com/locate/procedia

The 17th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 14-16, 2026, Istanbul, Türkiye

Fast and Scalable Population Synthesis Using Equivalence Classes and Hierarchical Distribution

Robin Andre[a,*], Pia Tulodetzki[a], Peter Vortisch[a]

[a] *Karlsruhe Institute of Technology (KIT), Institute for Transport Studies, 76131 Karlsruhe, Germany*

Abstract

Synthetic populations are essential for transportation research, yet full real-world population data are rarely available. Existing approaches, particularly Iterative Proportional Fitting (IPF) and Iterative Proportional Updating (IPU), struggle with heterogeneous geographic resolutions and large household samples containing redundant information. This paper introduces a new population synthesis algorithm that combines a top-level generation step with an optimized hierarchical allocation process. The method uses equivalence classes to merge content-identical households, reducing computational effort, and reverses the hierarchical IPU workflow by synthesizing households at the highest geographic level before distributing them downward. Experiments using German census data and the MiD 2017 survey show that equivalence classes reduce IPU runtime by up to a 31-fold factor, with the proposed algorithm achieving an additional 2x–4x speedup with substantially lower memory use. Solution quality, evaluated via mean absolute percentage error across 41 control variables, consistently surpasses hierarchical IPU, especially for small marginal sums.

Keywords: Population synthesis;Spatial heterogeneity; modeling; transportation;iterative proportional updating;agent based;

1. Introduction

Many transportation research models require detailed information on the structure and attributes of the population in the study area to improve their accuracy. The full information of the real populace is usually unobtainable, due to limitations in data collection, privacy concerns, and the overall infeasibility of full population surveys. A digital twin, a synthetic population, is used as proxy to provide access and modify the survey information spectrum, while matching the observed data of the real counterpart as closely as possible. The literature classifies methods used for this process into three categories: Synthetic Reconstruction (SR), combinatorial optimization (CO), and statistical learning (SL) [3]. The most prominent class has been synthetic reconstruction with the well established Iterative Proportional Fitting (IPF) procedure [1], which has been refined to handle both household and person level attributes in the Iterative Proportional Update (IPU) procedure [18]. Both the hierarchical IPF method [12] and the hierarchical IPU method

* Corresponding Author. Tel.:+49 721 608-47772
E-mail address: robin.andre@kit.edu

[6] provide an enhancement to handle input data at different resolution levels. More recent work has focused on the use of metaheuristics such as genetic algorithms [9], variational autoencoders [16] [5], or a combination of IPF with machine learning [17]. However, most tools in practice use a variant of IPF or IPU such as SILO [10], PopGen [14], ILUTE [15], CEMDAP [4], and MOBIUS[2]. The algorithm proposed in this paper will provide a new way for IPU to handle different input resolution levels, using both a SR approach to generate the initial population and a CO approach to map the population to the study area.

2. Methodology

IPU. The main benefit of IPU is that, since households are treated as an atomic unit, interdependencies in the household structure are maintained in the synthesis process [18]. It requires a set of microdata sample households H and a set of target marginal sums M as input. Each marginal sum m has a target value t_m and a contribution function $c_m : H \to \mathbb{R}$ to measure the contribution of a target household to m. For example, the marginal sum $m_A = (42, |\{p \in h : 0 \leq age(p) < 18\}|)$ would encode that the expected amount of minors in area A is 42. While fractional and negative contributions of households to a marginal sum are theoretically possible, in practical cases the contribution function either counts the amount of attribute occurrences in the household (such as the amount of minors) as a natural number or encodes a boolean attribute (such as household size) via 1 and 0. The households are converted to a frequency matrix, where each household is represented by a vector $v_h(M) := (c_m(h))_{m \in M}$ that encodes the contribution of the household to each target marginal sum. Each vector is assigned a scalar value δ^h, which represents the number of copies of the corresponding household that should be placed. During an iteration i the algorithm alters the scalar values to improve the solution. We indicate the current scalar in iteration i with δ^h_i. Each scalar is initialized to $\delta^h_0 = 1.0$. The algorithm then iterates through the target marginal sums in a fixed order, usually ordered by household attributes and person attributes. During an iteration, the algorithm selects a marginal sum m and optimizes the scalars of all households that contribute to the marginal sum called H_m.

$$H_m = \{h \in H | c_m(h) \neq 0\} \tag{1}$$

The algorithm then calculates the scaling factor f by which the scalars should be modified in the iteration.

$$f = \frac{t_m}{\sum_{h \in H_m} \delta^h_i \cdot c_m(h)} \tag{2}$$

Then, all scalars for households in H_m are scaled uniformly by:

$$\delta^h_{i+1} = \delta^h_i \cdot f \tag{3}$$

An example of this scaling process is shown in Figure 2, where the rule for household size 1 is optimized. This procedure is repeated until a stopping criterion is met, typically either a maximum number of iterations or no improvement in solution quality. Upon completion the scalar values are converted to a synthetic population by creating a number of copies of the associated household. Note that the scalar values are fractional and need to be converted to an integer to create a population sample.

In Figure 2 the set of households $\{h_1, h_2, h_3, h_4\}$ is converted to the vector representation using the marginal sums for gender in area A_1, A_2 and household size in B, the logics are the amount of households of size 1: $|1|$, the amount of households of size 2: $|2|$, the amount of males: ♂, and the amount of females: ♀ . Note that even though households h_2 and h_4 have a residents with different ages, the corresponding vectors v_2 and v_4 are encoded identically, since age is not part of the evaluation of any marginal sum. During the first iteration, the marginal sum for $B^{|1|}$ is optimized. Each vector with a nonzero entry, displayed with a dot, has its corresponding δ scaled by the calculated scaling factor f (Equation 3). At the end of this iteration, the marginal sum is matched, but other marginal sums will change as a side effect.

Hierarchical IPU. The classic IPU algorithm assumes that all marginal sums are present for a single target area. In practice many data sets for marginal sums are available at different levels of geographic resolution, either by different survey design, or to prevent deduction of personal information. The hierarchical IPU [6] provides a solution to handle

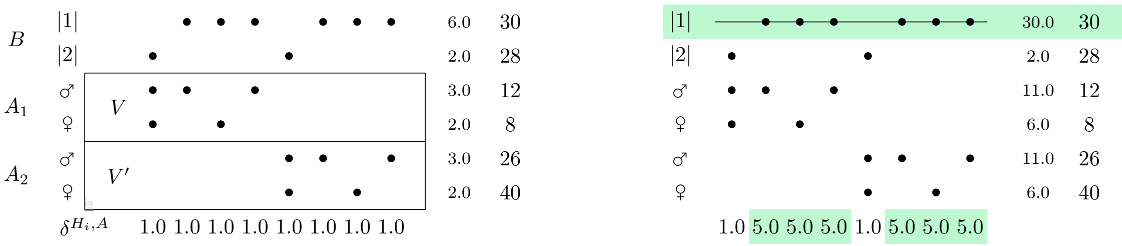

Fig. 1. An example for the conversion from households and marginal sums to the vector representation used in the IPU procedure showing the conversion process for a hierarchical area B that includes A_1 and A_2. Also showcasing the first iteration step where the rule |1| is optimized. The households are $h_1 = \{(♂, 30), (♀, 29)\}, h_2 = \{(♂, 42)\}, h_3 = \{(♀, 91)\}, h_4 = \{(♂, 17)\}$.

marginal sums of different resolutions. It copies the vector set for each child geographic unit, and appending each vector with the attribute encoding of the parent geographic unit marginal sums. The augmented vector set can then be optimized using the standard IPU procedure. This procedure can be repeated recursively to represent an arbitrary amount of resolution levels. Moreno and Moeckel synthesized the greater munich area this way with 3 resolution levels [11].

Equivalence classes. Our first contribution is to show that the input size for IPU can be reduced by using equivalence classes. The classic IPU algorithm will generate a $N \times m$ matrix, where N is the number of input households. Therefore, the algorithm runtime will scale with the input household set size, as shown in [7]. However, for both large sample inputs or a small amount of target sums, several households will be represented by a content identical vector. These households are indistinguishable from each other under the given target sums. As such, we can define an equivalence class as follows:

$$[h] = \{h_i \in H \mid v_h(M) = v_{h_i}(M)\} \tag{4}$$

where h is a representative for all households in the equivalence class. We define the set of equivalence classes as follows:

$$H/_M := \{[h]_M | h \in H\} \tag{5}$$

The IPU algorithm will produce the same output for input H or $H/_M$ when the scalar is initialized to $\delta_0^{[h]} = |[h]|$. We show this via induction over the iterations. For an arbitrary iteration i let

$$\forall h' \in [h] : \delta_i^{h'} = \delta_i^h = \frac{\delta^{[h]}}{|[h]|} \tag{6}$$

If the induction hypothesis holds 6 then the scaling factors f for the normal iteration step and f' for the equivalence class iteration step are identical since

$$\forall h' \in [h] : h' \in H_m \Leftrightarrow h \in H_m \Leftrightarrow [h] \in H_m/_M \tag{7}$$

and

$$\forall h' \in [h] : c_m(h) = c_m(h') =: c_m([h]) \tag{8}$$

by definition of the equivalence class in Equation 4. We can rearrange the sum of the factor calculation into groups of equivalence classes $[h_1], [h_2] \cdots [h_n]$ where $[h_i] \in H_m/_M$ since all elements $h' \in [h]$ are in H_m (Equation 7)

$$\sum_{h \in H_m} \delta_i^h \cdot c_m(h) = \sum_{[h_j] \in H_m/_M} \sum_{h' \in [h_j]} \delta_i^{h'} \cdot c_m(h') = \sum_{[h_j] \in H_m/_M} \delta_i^{h_j} \cdot \sum_{h' \in [h_j]} c_m(h') = \sum_{[h] \in H_m/_M} \delta_i^h \cdot |[h]| \cdot c_m(h) = \sum_{[h] \in H_m/_M} \delta_i^{[h]} \cdot c_m([h]) \tag{9}$$

Which, when inserted into equation 2 gives:

$$f = \frac{t_m}{\sum_{h \in H_m} \delta_i^h \cdot c_m(h)} = \frac{t_m}{\sum_{[h] \in H/_M} \delta^{[h]} \cdot c_m([h])} = f' \tag{10}$$

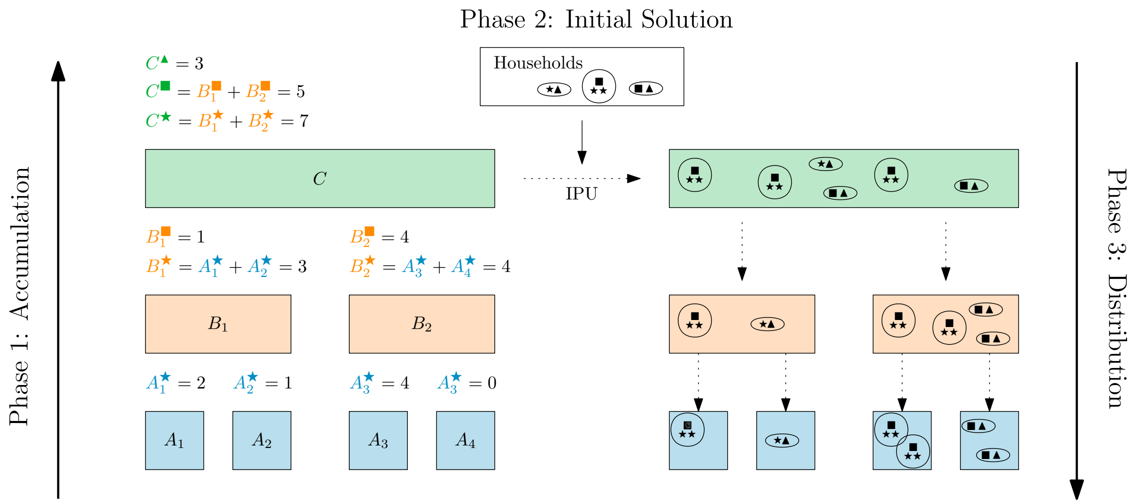

Fig. 2. A high level illustration of the hierarchical distribution (HD) algorithm, showcasing both the target sum accumulation, initial solution and distribution phases used to generate a synthetic population.

And thus by equation 3 that

$$\forall h' \in [h] : \delta_{i+1}^{h'} = \delta_i^{h'} \cdot f = \delta_i^h \cdot f = \delta_{i+1}^h \tag{11}$$

$$\delta_{i+1}^{[h]} = f' \cdot \delta_i^{[h]} = f \cdot \sum \delta_i^{h'} = |[h]| \cdot \delta_{i+1}^h \tag{12}$$

Where Equations 11 and 12 complete the induction step. The induction start for $i = 0$ holds automatically since $\delta_0^h = 1$ and $\delta_0^{[h]} = \delta_0^h \cdot |[h]|$ by definition of the variables.

Hierarchical Distribution (HD) algorithm. Our new HD algorithm inverts the conceptual procedure of the hierarchic IPU approach. Rather than optimizing a massive matrix at the lowest resolution level, a smaller matrix at the top level is optimized, and the generated households at the top level are distributed to the lower resolution levels. Figure 2 provides a graphical explanation of the algorithm using a small example study area. The algorithm itself can be structured in three phases. In the first phase the marginal sums of the child areas are recursively accumulated into the parent areas. For example, when two subareas A_1 and A_2 of the parent area B_1 have a common marginal sum, then B_1 can inherit a proxy marginal sum equal to the summation of the targets. For simplicity purposes, we will assume that the marginal sums at different resolution levels are logically distinct and that within a level, all target areas have an associated marginal sum for a given logic. As illustrated in Figure 2, the regions of level A have a marginal sum for the target ★, the regions of level B have a marginal sum for ■ and the region C has a marginal sum defined for ▲. The algorithm sifts up the target marginal sums of a level by creating a new rule with the target set to the sum of the targets. For example, the target for $B_1^{\bigstar}$ is resolved to be 3, the sum of $A_1^{\bigstar}$ and $A_2^{\bigstar}$.

Once the accumulation process has generated all target marginal sums for the top level area, a standard IPU procedure synthezises the households to produce an initial population. Unlike the standard IPU procedure, these households are not automatically assigned to their lowest area resolution, being more of a floating pool of households in the regional level C. To distribute the households in the pool to the lowest areas, we use a recursive algorithm. As long as there are households assigned to a high level area, the algorithm distributes the entire set of households from this area to the corresponding child areas. Figure 3 provides a small example of the distribution process.

We use a round-robin selection strategy: The subareas may each after another pick a single household from the set of remaining households in the parent area or withdraw from the selection process permanently. During an iteration, the subarea S calculates the expected gain of each household h using the sum of relative square difference for all associated marginal sums m of the area for the currently placed households H_s and $H_s + h$ (Equation 13). If at least

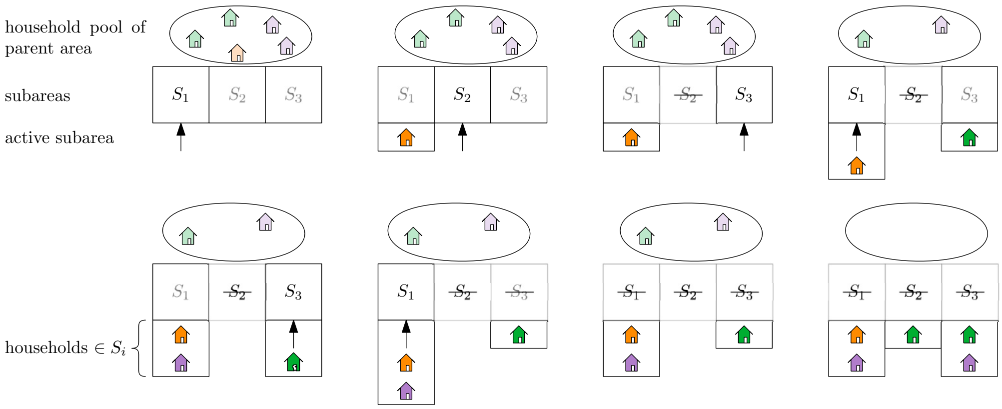

Fig. 3. A graphic visualizing the distribution process. The arrow indicates the currently active subarea that needs to perform a selection. S_1 selects the orange household. S_2 then immediately withdraws from the selection process. S_1 and S_3 then select their preferred household alternating until both areas also withdraw from the selection process. One green and one purple household remain and need to be assigned with the fallback strategy, which places the purple household in S_3 and the green household in S_2. The color is indicating the equivalence classes of the households, which can be used to speed up the calculation process.

one household yields a positive value under this metric, the household with the highest value is added to S, and the subsequent subarea proceeds with the selection process.

$$\sum\left(\frac{m(H_s + h) - m(H_s)}{m_t}\right)^2 \tag{13}$$

If none of the households exhibit a positive gain, the subarea selects no households and withdraws from the selection process. The withdrawal mechanism prevents sparsely populated subareas from being forced to select a household once their marginal sums have already been satisfied. Because the set of available households monotonically decreases over time, a subarea that finds all remaining households unsuitable at a given iteration will continue to do so in all subsequent iterations. Therefore, the subarea can be safely removed from further consideration. If all subareas ultimately withdraw, a fallback procedure is required to allocate the remaining households. In this case, all subareas are reinstated, and each household is assigned to the subarea that minimizes the relative error according to the selection metric defined in Equation 13.

3. Results

The German census [8] provides marginal sums for different levels of precision: municipality, municipal association, county, administrative district, federal state, and Germany. We can arbitrarily control the precision of the resolution quality by choosing the resolution level at which the marginal sums are extracted. For our experiments, we chose to acquire age and gender distributions at the municipality level and the remaining marginal sums at a variable level.

We use 41 different control variables for the marginal sums: For the person attributes we use age categorized by age groups, gender, current school type of students, gender distribution of students, the employment type by gender categorized in 3 classes, and the highest degree by gender of people aged 15 and above. For the household attributes household size, the seniority status of the household and the household type is used. Due to a variable mismatch in census data and survey household data we have only been able to associate the household type 1 (Single) to the corresponding census variable. As a workaround we grouped all other household types into its own type variable.

In the following, we use two separate rule sets of marginal sums to compare the algorithm runtime and solution quality. A small rule set R_{min} consisting only of: age, gender and household size. And R_{all} using all variables listed in Table 1 as input set. We chose 5 geographic areas within the census data with increasing population size. The name, number of municipalities and population size of each experiment area is listed in Table 2.

Table 1. The control attributes extracted from the census database for marginal sums.

Attribute	Name	Number of Variables	Variable Categories
Person	Age	11	0, 2, 5, 9, 15, 18, 24, 39, 59, 66, 74
	Gender	2	Male/Female
	School by level	3	Elementary / Middle school / Highschool
	School by gender	2	Male/Female students
	Employment by gender	6	Male/Female × Employed,Unemployed,Other
	Highest degree by gender	6	Male/Female × General,Secondary School, Higher School
Household	Size	6	1, 2, 3, 4, 5, 6+
	Type	2	Type 1 (Single)/Other
	Seniority Status	3	Seniors Only/Mixed/None

Table 2. A description of the geographic areas used in the experiments.

Name	Resolution Level	Number of Municipalities	Number of Residents
Marne	municipal association	13	12 848
Herzogtum Lauenburg	county	140	203 381
Oberbayern	administrative district	500	4 678 189
Nordrhein-Westfalen	state	396	17 890 489
Germany	country	10 786	82 711 282

Table 3. Runtimes of the standard IPU and the HD algorithm in seconds (s) using equivalence classes and unfiltered households for R_{min} andR_{all}.

				Runtime [s]				
Input	Rule set	Equivalence class	Amount of vectors	Marne	Lauenburg	Oberbayern	Nordrhein-Westfalen	Germany
HD	R_{min}	✓	2514	0.89	1.83	25.4	94.9	1019
IPU	R_{min}	✓	2514	2.9	44.3	166	130	3509
IPU	R_{min}	×	130274	119	1255	4850	3785	-
HD	R_{all}	✓	18088	4.04	21.3	265	758	5897
IPU	R_{all}	✓	18088	58.0	735	2964	2360	-
IPU	R_{all}	×	130724	379	4395	16370	12771	-

Table 3 shows the run times of the different algorithms. We compare three algorithm configurations: The hierarchic IPU algorithm using the full household survey set, the hierarchic IPU algorithm using equivalence classes as input and our hierarchical distribution (HD) algorithm, also using equivalence classes.

In the experiments, a curated household set with 130724 household records from the Mobilität in Deutschland (MiD) data set from 2017 was used [13]. When creating equivalence classes using the small rule set R_{min} the number of unique vectors of the data set is reduced to 2514 (1.92%). When using the complete rule set R_{all} the number of unique vectors in the MiD is 18088 (13.84%). Comparing equivalence classes to the full vector set for IPU shows that the run time can be reduced to a fraction of the time required, taking only in average 3% of the time for R_{min} and 17% of the time for R_{all}. These numbers closely correlate to the reduction in vectors of $\frac{2514}{130724} = 0.02$ for R_{min} and $\frac{18088}{130724} = 0.14$ for R_{all} indicating a linear relation between runtime and vector set size. IPU failed to produce results for 3 of the 4 instances for Germany, each time exceeding the maximum allowed amount of RAM (256 GB). In contrast, our algorithm had a peak RAM consumption of 70 GB for the experiment run of Germany using R_{all}. In addition, our algorithm outperformed the traditional IPU in all experiment instances. The largest speedups were achieved on the Lauenburg instance, being 24 times as fast for R_{min} and 34 times as fast for R_{all}. The smallest speedups were achieved on the Nordrhein-Westfalen instance, generating only a speedup of 1.36 and 3.1 in comparison to IPU.

Our algorithm tends to perform much better when the experiment area is partitioned into many subareas, and less when the area is densely populated but sparsely partitioned. The latter phenomenon is caused by the distribution process of the algorithm, which only places a single household in each iteration. In large populations, the insertion

Table 4. Comparison of the algorithm solution quality using the mean absolute percentage error as metric.

Algorithm	Rule set	Marne	Herzogtum Lauenburg	Oberbayern	Nordrhein-Westfalen	Germany
HD	R_{min}	0.91%	0.65%	0.49%	0.48%	0.36%
IPU	R_{min}	8.88%	4.89%	1.17%	0.58%	2.18%
HD	R_{all}	1.23%	2.70%	2.94%	3.37%	3.57%
IPU	R_{all}	7.11%	8.47 %	4.85 %	5.48%	-

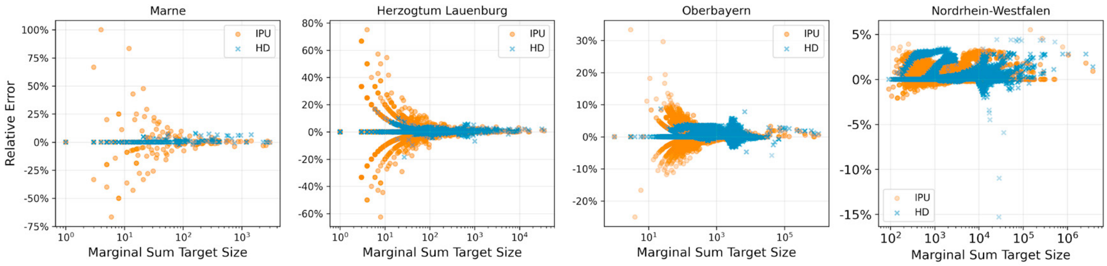

Fig. 4. All relative errors of the experiment areas for HD (blue) and IPU (orange) in comparison when using R_{min} as input rule set.

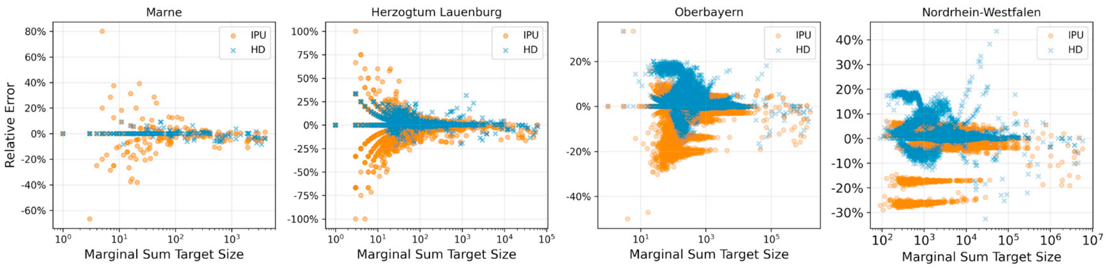

Fig. 5. All relative errors of the experiment areas for HD (blue) and IPU (orange) in comparison when using R_{all} as input rule set.

of a single household exerts a negligible influence on the selection metric. However, the metric is recalculated for each element. When comparing the run times of the algorithms in regard to the different rule sets, we can see that our algorithm increases in runtime with a factor of 8.08 from R_{min} to R_{all} whereas IPU scales with a factor of 18.08, indicating that our algorithm is less impacted by an increment of the variable set.

To measure the quality of the population synthesis we compute the mean absolute percentage error over the expected and observed values of the marginal sums as suggested by Moreno and Moeckel [11].

Table 4 shows the errors for our algorithm and the IPU algorithm using equivalence classes for each of the experiment areas. Our algorithm produces less average error for each of the experiments. In particular, IPU significantly drops in solution quality for the smaller instances down to a 8.8% deviation from the expected values. Our algorithm produces solutions of similar quality irrespective of the instance size. This phenomenon is in line with the aforementioned integerization process producing more pronounced errors for smaller marginal sum targets. Our algorithm mitigates this issue as the integerization is performed at the highest level area instead of the lowest areas and a significantly smaller number of vectors that need to be converted in the first place.

Figure 4 shows the algorithms' relative errors of the marginal sums in relation to their expected target value for 4 of the 5 experiment areas using R_{min}. IPU miscalculates smaller marginal sums more prominently for all instances. When comparing to the errors of R_{all} in Figure 5, both algorithms begin to produce larger errors in general. IPU produces outliers of -100% for Lauenburg and our algorithm produces outliers of 45% for Nordrhein-Westfalen. Another observation is the formation of error clusters in the Oberbayern and Nordrhein-Westfalen instances. The corresponding marginal sums measure the age groups $[0, 2]$, $[3, 5]$ and $[6, 9]$ which are underrepresented in the survey households. For small target values IPU forms visible error lines, due to more prominent integerization errors.

4. Conclusion

This paper has shown that the usage of equivalence classes can reduce the run time of IPU by a significant factor down to a fraction of 0.03 times the original execution duration. We introduced a new algorithm to calculate a synthetic population for hierarchic study areas. The conceptual contribution of this work is the inversion of the population generation mechanic, synthesizing at the highest geographic area and distributing downward instead of generating directly at the lowest geographic resolution. This process avoids induced errors by reducing the amount of rounding processes and high memory consumption. The algorithm outperforms the hierarchic IPU procedure in terms of runtime by a factor of 2x-4x while producing qualitatively better results.

However, there is still room for improvement, as the distribution algorithm is using a simple greedy selection strategy accounting for 90% of the entire execution time. There are ideas to improve this process. For example, the distribution process could easily be parallelized after the first distribution layer. Also a multilevel optimization scheme with reordering and swapping of households could improve the distribution quality by allowing poorly placed households to be rearranged.

References

[1] Beckman, R.J., Baggerly, K.A., McKay, M.D., 1996. Creating synthetic baseline populations. Transportation Research Part A: Policy and Practice 30, 415–429. URL: https://linkinghub.elsevier.com/retrieve/pii/0965856496000043, doi:10.1016/0965-8564(96)00004-3.

[2] Bigi, F., Rashidi, T.H., Viti, F., 2024. Synthetic population: A reliable framework for analysis for agent-based modeling in mobility. Transportation Research Record 2678, 1–15.

[3] Fabrice Yaméogo, B., Gastineau, P., Hankach, P., Vandanjon, P.O., 2021. Comparing Methods for Generating a Two-Layered Synthetic Population. Transportation Research Record: Journal of the Transportation Research Board 2675, 136–147. URL: https://journals.sagepub.com/doi/10.1177/0361198120964734, doi:10.1177/0361198120964734.

[4] Guo, J.Y., Bhat, C.R., 2007. Population synthesis for microsimulating travel behavior. Transportation Research Record 2014, 92–101.

[5] Kim, E.J., Bansal, P., 2023. A deep generative model for feasible and diverse population synthesis. Transportation Research Part C: Emerging Technologies 148, 104053. URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X23000426, doi:10.1016/j.trc.2023.104053.

[6] Konduri, K.C., You, D., Garikapati, V.M., Pendyala, R.M., 2016. Enhanced Synthetic Population Generator That Accommodates Control Variables at Multiple Geographic Resolutions. Transportation Research Record: Journal of the Transportation Research Board 2563, 40–50. URL: https://journals.sagepub.com/doi/10.3141/2563-08, doi:10.3141/2563-08.

[7] Lenormand, M., Deffuant, G., 2012. Generating a synthetic population of individuals in households: Sample-free vs sample-based methods. arXiv preprint arXiv:1208.6403 .

[8] Ämter des Bundes und der Länder, S., 2024. Zensusdatenbank. https://ergebnisse.zensus2022.de/datenbank/online/. Accessed: 2025-09-22.

[9] Mahmood, I., Bishop, N., Calinescu, A., Wooldridge, M., Zachos, I., 2024. A multi-objective combinatorial optimisation framework for large scale hierarchical population synthesis. URL: http://arxiv.org/abs/2407.03180, doi:10.48550/arXiv.2407.03180. arXiv:2407.03180 [cs].

[10] Moeckel, R., 2017. Constraints in household relocation: Modeling land-use/transport interactions that respect time and monetary budgets. Journal of Transport and Land Use 10, 211–228.

[11] Moreno, A.T., Moeckel, R., 2018. Population Synthesis Handling Three Geographical Resolutions. ISPRS International Journal of Geo-Information 7, 174. URL: https://www.mdpi.com/2220-9964/7/5/174, doi:10.3390/ijgi7050174.

[12] Müller, K., Axhausen, K.W., 2012. Multi-level fitting algorithms for population synthesis , 20 p.URL: http://hdl.handle.net/20.500.11850/53912, doi:10.3929/ETHZ-A-010192943. artwork Size: 20 p. Medium: application/pdf Publisher: ETH Zurich.

[13] Nobis, C., Kuhnimhof, T., 2018. Mobilität in Deutschland- MiD: Ergebnisbericht. Technical Report. Bundesministerium für Verkehr und digitale Infrastruktur (BMVI).

[14] Pendyala, R., Christian, K., Konduri, K., 2011. Popgen 1.1 user's guide.

[15] Salvini, P., Miller, E.J., 2005. Ilute: An operational prototype of a comprehensive microsimulation model of urban systems. Networks and spatial economics 5, 217–234.

[16] Sané, A.R., Belaroussi, R., Hankach, P., Vandanjon, P.O., 2025. Population synthesis with deep generative model: a joint household-individual approach. Computational Urban Science 5, 34. URL: https://link.springer.com/10.1007/s43762-025-00195-9, doi:10.1007/s43762-025-00195-9.

[17] Tozluoğlu, , Dhamal, S., Yeh, S., Sprei, F., Liao, Y., Marathe, M., Barrett, C.L., Dubhashi, D., 2023. A synthetic population of Sweden: datasets of agents, households, and activity-travel patterns. Data in Brief 48, 109209. URL: https://linkinghub.elsevier.com/retrieve/pii/S2352340923003281, doi:10.1016/j.dib.2023.109209.

[18] Ye, X., Konduri, K., Pendyala, R.M., Sana, B., Waddell, P., 2009. A methodology to match distributions of both household and person attributes in the generation of synthetic populations, in: 88th Annual Meeting of the transportation research Board, Washington, DC.