Building a geo-referenced microsimulation model with discrete optimization

Ulf Friedrich[1], Ralf Münnich[2], and Kendra M. Reiter[1]

Keywords: microsimulation, synthetic geo data, big data, household address matching

1 Introduction

Microsimulation methods have become an important tool to support evidence-based policies. Li and O'Donoghue [1], however, stress the importance of generating adequate synthetic data that support conducting sophisticated microsimulations. Within the research of the MikroSim project (https://mikrosim.uni-trier.de/), the aim is to generate a synthetic, but realistic dataset based on the full population of Germany. In order to produce a fully geo-coded dataset, where information is generally available on aggregate levels of different hierarchies, micro units (households and persons) have to be placed into geo-coded dwellings.
The microsimulation model involves mathematical optimization problems with integrality constraints on some or all of the optimization variables, e.g., to model each member of a simulated population or to describe (binary) decisions within the model. It is therefore very natural to employ combinatorial optimization techniques to handle these discrete structures efficiently. More specifically, optimization algorithms have to be developed for the subproblem of *address selection*: Given a population generated in the first step of the microsimulation process and a target region, the households in the population have to be assigned to actual addresses within the region, i.e., an address has to be selected for each household in the population.

Table 1: Data sources and resolutions.

Data	Resolution:		
	Administrative	Grid cell	Geo-referenced
Households and persons (census)	+	+	-
Further statistical information (micro-census)	+	(-)	-
Houses (ATKIS, OSM)	+	+	+
Dwellings (census)	+	+	-

In the model, the region is defined in a pre-processing step based on register data and properties taken from Open Street Map. In the simplest version of the problem, only the potential sizes (i.e., capacities) of the addresses are considered for the assignment. This simple model is extended by additional distribution constraints to model the structural properties of the households.
While the computation time is often not crucial when considering only a subset of the population, e.g., for the simulation of a certain region or city, the big-data setting of a complete model typically requires specialized, fast algorithms, and techniques from

[1]Operations Research, Technical University of Munich
[2]Economic and Social Statistics, Trier University

data science. For example, in the address selection model for Germany, more than 40 million households are assigned to over 25 million addresses while using several statistical variables to measure the quality of the assignment. General purpose heuristics such as simulated annealing do generally not solve this instance within an acceptable time limit and do not provide quality certificates. In addition, large datasets from several sources (e.g., Open Street Map, city registers, grid-based census data, cf. the overview in Table 1) have to be combined and pre-processed in an efficient and secure way.

2 Methods

The model presented here uses a synthetic dataset for the city of Trier in Rhineland-Palatinate, Germany, resulting from a combination of data from the last German population and household census in 2011 and specific location data prepared by the Economic and Social Statistics Group at Trier University. Due to the way the German census is congregated, the data are divided into equal-sized grid cells with 100m sides, where each grid cell is assigned a unique Grid-ID, see Figure 1. In total, the dataset currently consists of 55 878 households and 19 679 Addresses, in 12 277 grid cells.

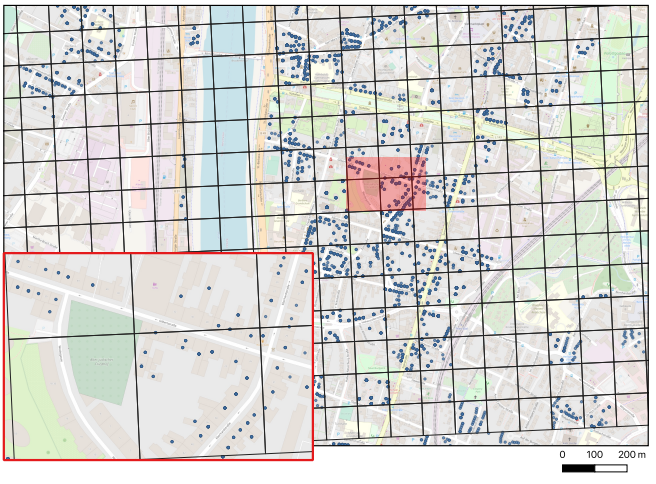

Figure 1: Example map view displaying the 100 Meter grid cells and housing markers (in blue). The map was produced using QGIS and OSM.

The dataset consists of two disjoint files: one for the households H and one for the addresses A. Each household has an attribute *household_size* which denotes the number of persons per household, with no distinction being made between adults and children. Similarly, each address is assigned a *capacity* category, which denotes the number of dwellings per building, ranging from a classic single-family home to a high-rise apartment building.
In data pre-processing, each address is split further into dwellings D, where each dwelling corresponds to a residence (house, apartment, etc.) for one household of a specific capacity. It is assumed that the more dwellings per building there are, the smaller the capacity per individual dwelling.
Based on the above-mentioned grid cells, it is possible to extract key parameters describing the (maximum) amount of persons living in the cell (denoted B_{per}) and the (maximum) amount of households living in the cell (denoted B_{hh}). These parameters are used to improve the model's depiction of the real-life living situation.

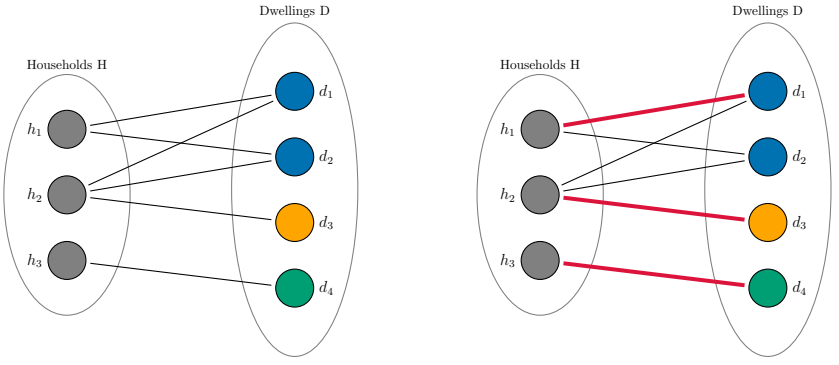

Figure 2: Example of a bipartite graph (left) with a maximal matching (right) in red.

The main goal is to assign each household into one dwelling, which is achieved through formulating a matching problem, a classical question in discrete mathematics. A *matching* M in a graph $G = (V, E)$ is defined as a subset of the edges such that no two edges in M share the same node, i.e., they are pairwise non-adjacent. A largest possible matching for a graph G is called a *maximum matching*. The interpretation of the model as a matching problem is visualized in Figure 2.
Since the sets of households and dwellings are disjoint, the graph $G = (H \cup D, E)$ is a so-called *bipartite* graph, where $|H| \leq |D|$ by design. An edge is drawn between each household $h \in H$ and each dwelling $d \in D$ if and only if the capacity of d is greater than or equal to the household size of h. The objective function f utilizes specific edge weights, which encode the fit between the household h and the dwelling d. To account for the differences in household size and capacities, several additional constraints are introduced as penalties in the linear objective function f. The task, then, is to find a (not necessarily unique) maximum weight bipartite matching in G. This way, the problem remains a pure matching formulation, as given below, for which polynomial-time algorithms exist.

$$
\begin{aligned}
\max_{x} \quad & f(x) & \\
\text{s.t.} \quad & \sum_{h \in H} x_{h,d} \leq 1 & \forall d \in D, \\
& \sum_{d \in D} x_{h,d} \leq 1 & \forall h \in H, \\
& x_{h,d} \in \{0, 1\} & \forall h \in H,\ \forall d \in D
\end{aligned}
$$

An efficient, polynomial algorithm is especially important considering the sizes of typical datasets for this model. A similar matching model has been used in [2] with data from the Swiss census 2000.

3 Results

The model is written in Python using the Gurobi Optimizer with its Python API to solve the optimization problem. The Python pandas package was used to keep the data organized in DataFrame objects. All maps were created using the QGIS application and the original data were compiled and loaded in the statistical software R. In the data processing step, the data are read from CSV and JSON files using the pandas package. All necessary parameters are extracted, indexed, and passed on to set up the constraints and objective function.
In speeding up the running time, it is important to not only consider the solver independently, but also to take the specific model structure into account. In particular,

a naïve approach of passing the constraints to the Gurobi solver is not efficient enough for the dataset size. Major improvements to the implementation could be achieved by passing constraints through the *Linear Expression* object instead, which results in an improvement in build time of the model from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$, cf. Figure 3.

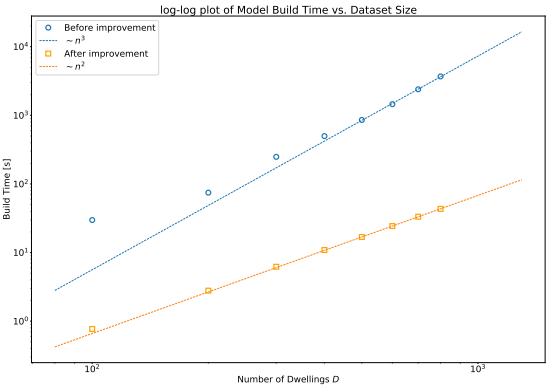

Figure 3: Performance log-log plot of model build time vs. dwelling dataset size.

The matching results are passed over to the microsimulation model as a DataFrame object where an entry of 1 in a cell means that the corresponding row (household) and column (dwelling) are matched, 0 otherwise. By combining this matrix with the original parameters of dwelling capacity and household size, it is possible to determine a measurement of the quality of the matching.

4 Conclusions

In conclusion, it is shown how using advanced techniques from discrete optimization can help to greatly improve the performance of microsimulation models. The approach is not limited to the application presented above, but can, in principle, be used to support geo-referenced analyses in further fields. The next steps of the research will extend the model to consider further constraints that stem from different data sources and on different hierarchies, such as grid cell or community information from surveys. In order to consider vagueness of estimates, soft constraints can be implemented.

Acknowledgements: The research of the second author is supported by the FOR 2559 research unit MikroSim, funded by the German Research Foundation. The first author receives support from the Volkswagen Foundation via the Experiment! initiative.

References

[1] Jinjing Li and Cathal O'Donoghue. A survey of dynamic microsimulation models: uses, model structure and methodology. *International Journal of Microsimulation*, 6(2):3–55, 2013.

[2] Paul Anderson, Bilal Farooq, Dimitrios Efthymiou, and Michel Bierlaire. Associations generation in synthetic population for transportation applications graph-theoretic solution. *Transportation Research Record*, 2429:38–50, 2014.