Computational Methods for the Household Assignment Problem

Ulf Friedrich, Lucas Moschen, Ralf Münnich, Martin Schmidt

Abstract. We consider the household assignment problem as it occurs in the geo-referencing step of spatial microsimulation models. The resulting model is a maximum weight matching problem with additional side constraints. For real-world instances such as the one for the city of Trier in Germany, the number of binary variables exceeds 10^9, and the resulting instances are far from being solvable with standard solvers for mixed-integer linear optimization. Hence, we derive two methods to compute feasible points of good quality—one based on the Lagrangian relaxation of the side constraints and the other one based on problem-tailored decomposition strategies. For both, we theoretically analyze the obtained feasible points. Moreover, we extensively test the two proposed methods on real-world and synthetic data sets and compare it with Gurobi as a benchmark. Our results show that the methods are significantly faster while computing points that are very close to being optimal. The methods are also much more efficient in terms of memory usage, which renders the application of classic branch-and-bound solvers impossible for real-world instances. Finally, our results for the city of Trier also show a realistic demographic distribution, which illustrates the applicability of our approach in practice.

1. Introduction

Decisions made concerning public policies and large-scale investments of today's societies usually have significant consequences for the population. For this reason, tools capable of predicting situations and providing significant information in this regard are becoming increasingly important (O'Donoghue et al. 2014). In particular, microsimulation models are gaining prominence as a powerful tool for such purposes and have many applications in various sectors of society like in economics (Flory and Stöwhase 2012; Basu et al. 1998; Pellegrino et al. 2011), in social policy (Atkinson et al. 2002; Ballas and Clarke 1999), or in the health system (Morrissey et al. 2008; Spielauer 2007).

Microsimulation models were first presented by Orcutt (1957) and the interest in considering geographical aspects in microsimulation models quickly started (Hägerstrand 1957). Since then, the study of a large number of social phenomena can be studied using the application of spatial microsimulation models. For instance, in Morrissey and O'Donoghue (2011), the model SMILE is used to analyze the spatial distribution of labor force participation and market earnings in Ireland. In Rephann and Holm (2004), the effects of Sweden's liberal immigration policy are studied using the dynamic spatial microsimulation model SVERIGE, which can evaluate scenarios arising from the projection of the population over time. Currently, a dynamic spatial microsimulation model of Germany's population is under development in the project MikroSim (Münnich et al. 2020). This model considers a detailed and realistic synthetic construction of the population at the micro level, taking into

Date: July 17, 2024.

2020 *Mathematics Subject Classification.* 90Bxx, 90B80, 90C11, 90C59, 90C90.

Key words and phrases. Household assignment, Microsimulation, Maximum weight matching, Decomposition methods, Approximation algorithms.

account a range of different statistical information. This allows the study of "what if" scenarios and projections over time in various areas like demography, health, transportation, housing, and others.

An important step in a spatial microsimulation model is the geo-referencing of its individuals, which usually translates into the need to get a matching for two or more data sets. This step is usually performed using statistical tools such as iterative proportional fitting (Deming and Stephan 1940; Birkin and Clarke 1988) or heuristic approaches such as simulated annealing or genetic algorithms (Williamson et al. 1998; Birkin et al. 2006; Ballas et al. 1999). Regarding the level of information, there are studies in which precise spatial coordinates are considered in this step; see, e.g., Cullinan (2010), where the authors define the location of households by a random assignment. However, there are also many cases in which the geo-referencing of units is done on a small area level (Rephann and Holm 2004; Morrissey et al. 2008; Ballas and Clarke 1999; Lovelace and Dumont 2017).

Similar to the model considered by Reiter (2021), we present a strategy in which population units are aggregated into households and assigned to dwellings in specific spatial coordinates on the municipality level by solving the so-called household assignment problem (HAP). To this end, we formulate this problem as a maximum weight matching (MWM) problem in a bipartite graph for which statistical information associated to the households and dwellings is used to define the weights of the edges. In addition, application-specific side constraints are included in the formulation to ensure that the resulting allocation satisfies statistical properties or distributions that can be observed in the considered regions on different hierarchical levels.

For real-world instances, the HAP's size can easily reach the range of billions of variables. Consequently, even its linear programming (LP) relaxation is difficult to treat, which prevents the application of LP-based techniques (Wolsey 1998). At the same time, its structure does not allow the problem to be decomposed without a resulting loss in optimality. In these situations, it is possible to observe the development of case-specific heuristics that aim to find high-quality feasible points by decomposing the problem into reasonably sized sub-problems; see, e.g., Giortzis et al. (2000) or Chapter 5 of Noor-E-Alam (2013). Hence, one of our core contributions is that we derive a similar algorithm in which specific attributes of the households and dwellings are used to decompose the model. Additionally, the concept of a side-constraint-maximal matching is introduced and it is shown that our decomposition approach obtains a matching that satisfies this property.

The structure of the considered model also suggests the application of Lagrangian-relaxation-based techniques (Schrijver 1986; Korte and Vygen 2019). These techniques are characterized by relaxing a subset of constraints so that the resulting problem has a simpler structure. Therefore, an approximate solution to the original formulation is obtained by solving the relaxed models along with the search for proper Lagrange multipliers. In our case, the relaxed model corresponds to an MWM problem, for which there are several graph-based algorithms (Edmonds and Karp 1972; Korte and Vygen 2019; Munkres 1957) and also the possibility of solving it as an LP due to the total unimodularity property (Schrijver 1986, Chapter 19). However, in our case, the size of the models prevents the application of such strategies within reasonable time and memory resources. Therefore, another contribution of this paper is a novel approach using approximation algorithms to tackle the relaxed models in an iterative way. For this approach, we show that the approximation guarantee for the relaxed model leads to a quality guarantee for the objective value of the obtained matching.

Our final contribution is an extensive computational study on synthetic data sets as well as on a real-world data set for the city of Trier in Germany. The experiments clearly show that the proposed methods are significantly faster and less memory-consuming than the standard approach to apply a commercial mixed-integer optimization solver and also compute feasible points that are very close to optimal ones. Our results for the city of Trier also show a realistic demographic distribution, which illustrates the applicability of our approach in practice.

The remainder of the paper is organized as follows. In Section 2, we derive the model. We discuss the decomposition approach in Section 3 and the Lagrangian-relaxation-based approach in Section 4. In Section 5, we present our computational study before we conclude in Section 6.

2. Modeling

The population units are grouped into a household data set with the aim that, in the geo-referencing process, each household shares the same dwelling within an address. Furthermore, a dwelling data set is used, which carries specific geospatial coordinates. In this sense, the population is geo-referenced by matching household and dwelling data sets. For this problem, a bipartite graph $G = (V, E)$ is constructed whose vertex set V is the disjoint union of the set H of households and the set D of dwellings, i.e., $V = H \cup D$. Based on available statistical information in the data sets, an edge $\{h, d\} \in E$ is added to the graph G if it reflects a realistic housing possibility.

Using the statistical information from the data sets, it is possible to define a weight $w_{h,d} \in (0, 1]$ so that, given any edge $\{h, d\} \in E$, the weight $w_{h,d}$ represents how adequate the dwelling d is for the household h. It would also be possible to consider real-world data having $w_{h,d} = 0$ but since the respective edge will never be chosen in an optimal solution, we omit this here. We represent a matching M in the graph G by a set of variables $x_{h,d}$ defined by

$$x_{h,d} = \begin{cases} 1, & \text{if } \{h, d\} \in M, \\ 0, & \text{otherwise.} \end{cases}$$

Geo-referencing of population units is then achieved by solving the maximum-weight matching (MWM) problem

$$\max_{x} \quad \sum_{\{h,d\} \in E} w_{h,d}\, x_{h,d} \tag{1a}$$

$$\text{s.t.} \quad \sum_{d:\{h,d\} \in E} x_{h,d} \leq 1, \qquad h \in H, \tag{1b}$$

$$\sum_{h:\{h,d\} \in E} x_{h,d} \leq 1, \qquad d \in D, \tag{1c}$$

$$x_{h,d} \in \{0, 1\}, \qquad \{h, d\} \in E. \tag{1d}$$

Obviously, depending on infrastructural or socio-economic properties, different regions may have significantly different housing properties. In this regard, side constraints are incorporated into Problem (1) to ensure that any feasible matching M reflects an allocation that aligns with local housing characteristics. These constraints can be added to the model on a given grid structure with grid cells considered over the entire region. The relation between the set of grid cells K and the set of dwellings D is established by the binary encoding

$$s_{d,k} = \begin{cases} 1, & \text{if dwelling } d \text{ is located in grid cell } k, \\ 0, & \text{otherwise.} \end{cases}$$

Let $p_h \in \mathbb{N}$ denote the number of persons in the household h, and let B_k^{hhd} and B_k^{per} be an upper bound on the total number of households and persons in grid cell k, respectively. The constraints

$$\sum_{\{h,d\}\in E} s_{d,k}\, x_{h,d} \le B_k^{\text{hhd}}, \quad k \in K^{\text{hhd}},$$

and

$$\sum_{\{h,d\}\in E} p_h\, s_{d,k}\, x_{h,d} \le B_k^{\text{per}}, \quad k \in K^{\text{per}},$$

are added to Problem (1), where $K^{\text{hhd}}, K^{\text{per}} \subseteq K$ are sets of grid cells. In general, these additional constraints are added to the model only for a fraction of all grid cells, depending on the regional properties. The MWM problem with side constraints considered in this work is inspired by the formulation given in Reiter (2021, Chapter 6) and reads as follows:

$$\max_x \quad \sum_{\{h,d\}\in E} w_{h,d}\, x_{h,d} \tag{2a}$$

$$\text{s.t.} \quad \sum_{d:\{h,d\}\in E} x_{h,d} \le 1, \qquad h \in H, \tag{2b}$$

$$\sum_{h:\{h,d\}\in E} x_{h,d} \le 1, \qquad d \in D, \tag{2c}$$

$$\sum_{\{h,d\}\in E} s_{d,k}\, x_{h,d} \le B_k^{\text{hhd}}, \qquad k \in K^{\text{hhd}}, \tag{2d}$$

$$\sum_{\{h,d\}\in E} p_h\, s_{d,k}\, x_{h,d} \le B_k^{\text{per}}, \qquad k \in K^{\text{per}}, \tag{2e}$$

$$x_{h,d} \in \{0,1\}, \qquad \{h,d\} \in E. \tag{2f}$$

In instances of practical size, Problem (2) has a very large number of variables, which easily reaches the range of billions. Theorem 1 shows that a specific part of the binary variables can be relaxed to continuous variables in the problem formulation to simplify the problem, while keeping the problem size, however, unchanged.

Theorem 1. *For $D_K = \{d \in D : s_{d,k} = 1, k \in K^{hhd} \cup K^{per}\}$ consider the mixed-integer linear problem (MILP)*

$$\max_x \quad \sum_{\{h,d\}\in E} w_{h,d}\, x_{h,d} \tag{3a}$$

$$s.t. \quad \sum_{d:\{h,d\}\in E} x_{h,d} \le 1, \qquad h \in H, \tag{3b}$$

$$\sum_{h:\{h,d\}\in E} x_{h,d} \le 1, \qquad d \in D, \tag{3c}$$

$$\sum_{\{h,d\}\in E} s_{d,k}\, x_{h,d} \le B_k^{hhd}, \qquad k \in K^{hhd}, \tag{3d}$$

$$\sum_{\{h,d\}\in E} p_h\, s_{d,k}\, x_{h,d} \le B_k^{per}, \qquad k \in K^{per}, \tag{3e}$$

$$x_{h,d} \in \{0,1\}, \qquad \{h,d\} \in E, \quad d \in D_K, \tag{3f}$$

$$0 \le x_{h,d} \le 1, \qquad \{h,d\} \in E, \quad d \in D \setminus D_K. \tag{3g}$$

If all binary variables $x_{h,d}$ with $\{h,d\} \in E$ and $d \in D_K$ in Problem (3) are arbitrarily fixed such that (3d) and (3e) hold, then all extreme points of the feasible set of the

remaining linear problem are integers. In particular, there is an optimal solution for Problem (3) that is binary.

Proof. Let y be the vector of variables $x_{h,d}$ with $\{h,d\} \in E$ and $d \in D_K$ and let z be the vector of variables $x_{h,d}$ with $\{h,d\} \in E$ and $d \in D \setminus D_K$. Moreover, let Y be the set of variable vectors y that satisfy

$$\sum_{h:\{h,d\}\in E} x_{h,d} \leq 1, \quad d \in D_K,$$

$$\sum_{\{h,d\}\in E} s_{d,k}\ x_{h,d} \leq B_k^{\text{hhd}}, \quad k \in K^{\text{hhd}},$$

$$\sum_{\{h,d\}\in E} p_h\ s_{d,k}\ x_{h,d} \leq B_k^{\text{per}}, \quad k \in K^{\text{per}},$$

$$x_{h,d} \in \{0,1\}, \quad \{h,d\} \in E, \quad d \in D_K,$$

and let $Z(y)$ be the y-depending set of variable vectors z that satisfy

$$\sum_{d\in D\setminus D_K:\{h,d\}\in E} x_{h,d} \leq 1 - \sum_{d\in D_K:\{h,d\}\in E} x_{h,d}, \quad h \in H, \tag{4a}$$

$$\sum_{h:\{h,d\}\in E} x_{h,d} \leq 1, \quad d \in D \setminus D_K, \tag{4b}$$

$$0 \leq x_{h,d} \leq 1, \quad \{h,d\} \in E,\ d \in D \setminus D_K. \tag{4c}$$

By setting

$$w^y y = \sum_{d\in D_K:\{h,d\}\in E} w_{h,d}\ x_{h,d}$$

and

$$w^z z = \sum_{d\in D\setminus D_K:\{h,d\}\in E} w_{h,d}\ x_{h,d},$$

Problem (3) is equivalent to the problem

$$\max_y \quad P(y) \quad \text{s.t.} \quad y \in Y,$$

where the problem $P(y)$ is given by

$$\max_z \quad w^z z + w^y y \quad \text{s.t.} \quad z \in Z(y).$$

Given any fixed values for the binary variables $x_{h,d}$ with $d \in D_K$ that satisfy Constraints (3d) and (3e), these values correspond to some vector $y \in Y$. Since Constraints (4a) and (4b) correspond to a totally unimodular matrix, this implies that all the extreme points of the polytope defined by (4) are integer-valued (Schrijver 1986, Chapter 19), from which the result follows. □

3. Decomposition Methods

In many microsimulation models, the assignment of households to dwellings shall be performed at a municipal or district level. For a given municipality, the amount of variables for Model (3) depends on the corresponding amount of households and dwellings as well as on the available statistical information, which directly impacts the number of possible assignments of each household. Consequently, it is possible to observe reasonably-sized municipalities of Germany for which the number of variables of the Problem (3) is in the range of billions. In such cases, it is generally not possible to solve the respective instances even with today's most involved commercial solvers—let it be due to memory restrictions even on large-scale high performance computing systems or due to time limits for the solution process.

For optimization problems that require vast amounts of time or memory, applying heuristics and exploiting problem-specific information facilitate the computation of high-quality feasible points. These points are often found via solving smaller sub-problems (Giortzis et al. 2000; Noor-E-Alam 2013, Chapter 5), usually arising from a decomposition of the original problem. In this section, we present two such decomposition strategies.

First, the information about the households' sizes is used to decompose Problem (3) into sub-problems of the same mathematical structure, but considering only possible assignments of households of a specific size to dwellings with a sufficient capacity. Since for smaller household sizes the amount of possibilities to find dwellings tends to increase naturally, it can often be seen that these sub-problems still have too many variables to be solved using reasonable time and memory resources. For this reason, we derive a second strategy that takes regional information into account to additionally decompose these sub-problems if necessary. This leads to an algorithm that is capable of finding a feasible solution for the full problem. For the latter decomposition heuristic, we incorporate verification steps to obtain both a good objective value for the assignments made and the maximality of the corresponding matching with respect to the side constraints.

3.1. **Decomposition by Household Size.** Given a possible assignment $\{h,d\} \in E$, one of the most important aspects for computing the weight $w_{h,d}$ is the relationship between the household's size $p_h \in \mathbb{N}$ and the dwelling's capacity $c_d \in \mathbb{N}$. Considering general housing aspects of German municipalities, it is reasonable to define that an edge $\{h,d\}$ only exists if $p_h \leq c_d$. Moreover, the closer $c_d - p_h$ is to zero, the bigger is the weight $w_{h,d}$. Therefore, although the computation of the weights considers all the available statistical information to define how realistic a possible assignment $\{h,d\}$ is, the values p_h and c_d play an important role. This fact implies that an algorithm that decomposes Problem (3) into sub-problems related to the allocation of households of a specific size has a good chance to lead to a good objective function value.

Motivated by this discussion, we present a decomposition by household size for Problem (3). Each iteration of the algorithm starts by finding the largest household size $p_h^{\max}$. With this value, we define $H_{\max} = \{h \in H : p_h = p_h^{\max}\}$, $D_{\max} = \{d \in D : c_d \geq p_h^{\max}\}$, and $E_{\max} = \{\{h,d\} \in E : h \in H_{\max}, d \in D_{\max}\}$. Then, we first assign the households in $H_{\max}$ by solving the problem

$$\max_x \quad \sum_{\{h,d\} \in E_{\max}} w_{h,d}\, x_{h,d} \tag{5a}$$

$$\text{s.t.} \quad \sum_{d:\{h,d\} \in E_{\max}} x_{h,d} \leq 1, \qquad h \in H_{\max}, \tag{5b}$$

$$\sum_{h:\{h,d\} \in E_{\max}} x_{h,d} \leq 1, \qquad d \in D_{\max}, \tag{5c}$$

$$\sum_{\{h,d\} \in E_{\max}} s_{d,k}\, x_{h,d} \leq B_k^{\text{hhd}}, \qquad k \in K^{\text{hhd}}, \tag{5d}$$

$$\sum_{\{h,d\} \in E_{\max}} p_h\, s_{d,k}\, x_{h,d} \leq B_k^{\text{per}}, \qquad k \in K^{\text{per}}, \tag{5e}$$

$$x_{h,d} \in \{0,1\}, \qquad \{h,d\} \in E_{\max}, \quad d \in D_K, \tag{5f}$$

$$0 \leq x_{h,d} \leq 1, \qquad \{h,d\} \in E_{\max}, \quad d \in D \setminus D_K. \tag{5g}$$

After each solution of Problem (5), we iterate the process by considering all households except for those in $H_{\max}$ and taking only those dwellings into account that no household was assigned to. This heuristic is formally detailed in Algorithm 1.

Algorithm 1: Decomposition by Household Size

Input: Problem (3)
Output: A feasible point $\hat{x}$ for Problem (3).

1 Initialize $\hat{x} \leftarrow 0$.
2 **while** $H \neq \emptyset$ **do**
3 Set $p_h^{\max} \leftarrow \max\{p_h : h \in H\}$.
4 Set $H_{\max} \leftarrow \{h \in H : p_h = p_h^{\max}\}$ and $D_{\max} \leftarrow \{d \in D : c_d \geq p_h^{\max}\}$.
5 Set $E_{\max} \leftarrow \{\{h, d\} \in E : h \in H_{\max}, d \in D_{\max}\}$.
6 Solve Problem (5) and let x be the solution.
7 Set $D_{\text{assign}} \leftarrow \emptyset$.
8 **for** d **in** $D_{\max}$ **do**
9 **if** $\exists h \in H_{\max} : x_{h,d} = 1$ **then**
10 Set $D_{\text{assign}} \leftarrow D_{\text{assign}} \cup \{d\}$.
11 Let k_d be the grid cell in which d is located.
12 **if** $k_d \in K^{hhd}$ **then**
13 Set $B_{k_d}^{\text{hhd}} \leftarrow B_{k_d}^{\text{hhd}} - 1$.
14 **end**
15 **if** $k_d \in K^{per}$ **then**
16 Set $B_{k_d}^{\text{per}} \leftarrow B_{k_d}^{\text{per}} - p_h$.
17 **end**
18 **end**
19 **end**
20 Set $H \leftarrow H \setminus H_{\max}$ and $D \leftarrow D \setminus D_{\text{assign}}$.
21 **for** $\{h, d\}$ **in** $E_{\max}$ **do**
22 Set $\hat{x}_{h,d} \leftarrow x_{h,d}$.
23 **end**
24 **end**

In Algorithm 1, a sequence of instances of Problem (5) is solved, which are smaller in size than Problem (3), but still contain the most accurate dwelling options for the households in $H_{\max}$. By Line 20 of Algorithm 1, households that are not assigned in the iteration in which they are in $H_{\max}$ will never be assigned to a dwelling. If this situation occurs or not, obviously depends on the given instance.

In addition, in Line 6 of Algorithm 1 it can still be the case that the respective MILP (5) is too large to be solved in a reasonable amount of time or with a reasonable amount of available memory. In particular, in present-day municipalities, the number of small-sized households and dwellings is significantly larger than the number of large ones. For this reason, in later iterations of the algorithm, the sets $H_{\max}$ and $D_{\max}$ tend to have a large number of elements, which may lead to instances of Problem (5) that still cannot be solved in practice. Hence, in the next section, we present another decomposition method to compute feasible points for Problem (5) in such cases.

3.2. **Regional Decomposition.** Large-scale MILPs with a grid structure can often be tackled using decomposition strategies based on the decomposition of the considered geographical region; see, e.g., Noor-E-Alam (2013, Section 5.4). In our case, we exploit sub-regions consisting of disjoint sets of grid cells and the sub-problems are then generated according to the information corresponding to each sub-region. Following this idea, we present a regional decomposition strategy to compute a feasible solution for Problem (5) if a direct solution is impossible.

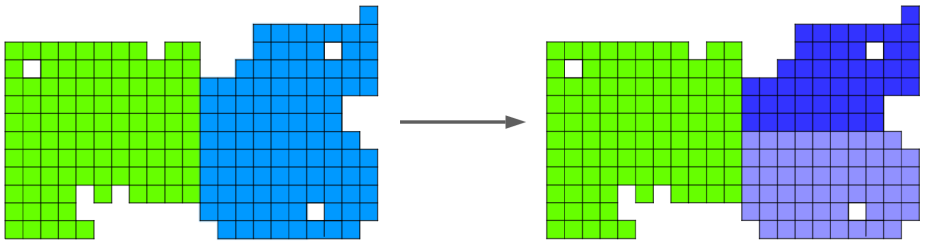

FIGURE 1. Iterative Regional Decomposition

The key idea is to halve the geographical region iteratively until each obtained sub-region contains a sufficiently small amount of dwellings; see Figure 1. Afterward, the households in $H_{\max}$ are assigned to these sub-regions so that the average weight between each household and the dwellings contained in the assigned sub-region is maximized. This is realized by solving an auxiliary integer linear problem that we describe in the following.

Let us denote the sub-regions created as non-empty, disjoint subsets $D^1_{\max}, \dots, D^n_{\max}$ of $D_{\max}$ such that $\bigcup_{i=1,\dots,n} D^i_{\max} = D_{\max}$. Now, each possible assignment of a household h to a sub-region $D^i_{\max}$ is represented by the variable $y_{h,D^i_{\max}} \in \{0,1\}$ for $h \in H_{\max}$ and $i \in \{1,\dots,n\}$, having the interpretation

$$y_{h,D^i_{\max}} = \begin{cases} 1, & \text{if } h \text{ is assigned to } D^i_{\max}, \\ 0, & \text{otherwise.} \end{cases}$$

Given a household h and a sub-region $D^i_{\max}$, the weight $\chi_{h,D^i_{\max}} \in [0,1]$ of the possible assignment is expressed by the arithmetic mean of the values $w_{h,d}$ for the dwellings located in $D^i_{\max}$, i.e.,

$$\chi_{h,D^i_{\max}} = \frac{1}{|D^i_{\max}|} \sum_{d \in D^i_{\max}} w_{h,d}.$$

To ensure that each household is assigned to exactly one sub-region, we add the constraints

$$\sum_{i=1}^{n} y_{h,D^i_{\max}} = 1, \quad h \in H_{\max}.$$

Finally, to guarantee that the household distribution occurs proportionally to the number of dwellings in each sub-region, we use the constraints

$$\sum_{h \in H_{\max}} y_{h,D^i_{\max}} \leq B_{D^i_{\max}}, \quad i = 1,\dots,n,$$

with

$$B_{D^i_{\max}} = \left\lceil \frac{|D^i_{\max}| \cdot |H_{\max}|}{|D_{\max}|} \right\rceil.$$

The integer linear problem is then given by

$$\max_{y} \quad \sum_{h\in H_{\max}} \sum_{i=1}^{n} \chi_{h,D^i_{\max}}\, y_{h,D^i_{\max}} \tag{6a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_{h,D^i_{\max}} = 1, \qquad h \in H_{\max}, \tag{6b}$$

$$\sum_{h\in H_{\max}} y_{h,D^i_{\max}} \le B_{D^i_{\max}}, \qquad i = 1,\dots n, \tag{6c}$$

$$y_{h,D^i_{\max}} \in \{0,1\}, \qquad h \in H_{\max},\ i = 1,\dots n. \tag{6d}$$

After solving Problem (6), $H_{\max}$ can be decomposed into $H^1_{\max},\dots,H^n_{\max}$ with $H^i_{\max} = \{h \in H_{\max} : \chi_{h,D^i_{\max}} = 1\}$ for $i \in \{1,\dots,n\}$. Let us define the edge set $E^i_{\max} = \{\{h,d\} \in E : h \in H^i_{\max},\, d \in D^i_{\max}\}$, which consists of the possible assignments of households in $H^i_{\max}$ to dwellings in $D^i_{\max}$. The allocation of households to dwellings in every sub-region is then finally computed by solving

$$\max_{x} \quad \sum_{\{h,d\}\in E^i_{\max}} w_{h,d}\, x_{h,d} \tag{7a}$$

$$\text{s.t.} \quad \sum_{d:\{h,d\}\in E^i_{\max}} x_{h,d} \le 1, \qquad h \in H^i_{\max}, \tag{7b}$$

$$\sum_{h:\{h,d\}\in E^i_{\max}} x_{h,d} \le 1, \qquad d \in D^i_{\max}, \tag{7c}$$

$$\sum_{\{h,d\}\in E^i_{\max}} s_{d,k}\, x_{h,d} \le B^{\text{hhd}}_k, \qquad k \in K^{\text{hhd}}, \tag{7d}$$

$$\sum_{\{h,d\}\in E^i_{\max}} p_h\, s_{d,k}\, x_{h,d} \le B^{\text{per}}_k, \qquad k \in K^{\text{per}}, \tag{7e}$$

$$x_{h,d} \in \{0,1\}, \qquad \{h,d\} \in E^i_{\max},\ d \in D_K, \tag{7f}$$

$$0 \le x_{h,d} \le 1, \qquad \{h,d\} \in E^i_{\max},\ d \in D \setminus D_K. \tag{7g}$$

We now embed the sketched procedure in the iterative method given by Algorithm 2. An iteration of the algorithm begins by decomposing the region until all the sub-regions $D^1_{\max},\dots,D^n_{\max}$ are sufficiently small; see Line 3 of Algorithm 2. Then, Problem (6) is solved so that the households are distributed among these sub-regions. For each $i \in \{1,\dots,n\}$ Problem (7) is solved, thereby assigning the households to the dwellings in each sub-region $D^i_{\max}$. To guarantee a good quality of the feasible point, a parameter $\alpha \in [0,1]$ is set so that an assignment $\{h,d\}$ is only included in the final output if $w_{h,d} \ge \alpha$ holds; see Lines 8 and 9. Afterward, the sets of households and dwellings are updated to be considered in subsequent iterations (Line 10) and the side constraint data is updated for the next iterations as well; see Lines 12–18.

The procedure described above may terminate although feasible assignments are still available at the very end of the while-loop. After the while-loop, the number of remaining dwellings is sufficiently small and Problem (5) is solved with an exact method on the remaining households and dwellings.

In a given iteration of the while-loop, it is possible that many available assignments $\{h,d\}$ do not satisfy $w_{h,d} \ge \alpha$, which leads to only a few assignments being accepted to the final output. This can lead to many iterations and, in some cases, can even avoid the termination of the algorithm. Therefore, a parameter $A_{\min} \in \mathbb{N}$ is used to check if the amount of accepted assignments is sufficiently large; see Lines 23–32. If it is not, α is decreased by a fixed parameter $\beta \in (0,1]$ for the next

iteration. Moreover, a parameter $\gamma \in [0,1]$ is used so that, if $\alpha \leq \gamma$, then both α and $A_{\min}$ are set to zero. By doing so, a larger number of accepted assignments is guaranteed, which increases the objective value. Moreover, in an iteration for which α and $A_{\min}$ are equal to zero, the algorithm terminates if the number of dwellings is still larger than $\bar{D}$ and no new assignment is made. The theoretical properties guaranteed by the algorithm are discussed in detail in the next section.

In the remainder of this paper, the version of Algorithm 1 that additionally uses Algorithm 2 for solving Problem (5) in Line 6 will be referred to as the decomposition approach.

3.3. **Theoretical Properties of the Decomposition Approach.** We now formally define the concept of maximality w.r.t. the side constraints (2d) and (2e).

Definition 1. *Given an MWM problem with side constraints in a bipartite graph $G = (V, E)$, a feasible matching M is said to be side-constraint-maximal (SCM) if for any assignment $\{h, d\} \in E \setminus M$, the set $M' = M \cup \{\{h, d\}\}$ is infeasible.*

Lines 23–32 of Algorithm 2 are key to ensure that the algorithm returns a SCM matching for Problem (5). We first show this property for Algorithm 2, see Theorem 2, before we then prove it for Problem (2), see Theorem 3.

Theorem 2. *The output of Algorithm 2 is an SCM matching for Problem (5).*

Proof. We start with proving that Algorithm 2 terminates after a finite number of while-loop iterations. Let us assume the opposite. Then, since the sets $H_{\max}$ and $D_{\max}$ are finite, there is an iteration r such that for every later iteration no new assignments are made, i.e., $A = 0$ holds. Thus, for some iteration t with $r \leq t \leq r + \lceil (\alpha_0 - \gamma)/\beta \rceil$ it holds

$$\alpha_t = \alpha_0 - \left\lceil \frac{\alpha_0 - \gamma}{\beta} \right\rceil \beta, \tag{8}$$

where α_t is the value computed for α in Line 27 (in iteration t) and α_0 is its initial value. However, observe that

$$\alpha_0 - \left\lceil \frac{\alpha_0 - \gamma}{\beta} \right\rceil \beta \leq \alpha_0 - \frac{\alpha_0 - \gamma}{\beta} \beta = \gamma,$$

holds. Then, from (8) we get $\alpha_t \leq \gamma$ and, thus, the parameters α and $A_{\min}$ are set to 0 after Line 28. Therefore, in iteration $t+1$ the conditions in Lines 23 and 24 are satisfied and we stop, which contradicts our assumption.

Next, we prove that the output of Algorithm 2 is an SCM matching. Again, let us suppose it is not. Definition 1 then leads to the existence of $\{h_0, d_0\} \in E_{\max} \setminus M$ such that $M' = M \cup \{\{h_0, d_0\}\}$ is a feasible matching. In particular, this implies $\{h_0, d_0\} \notin M$, which leads to $h_0 \in H_{\max}$ and $d_0 \in D_{\max}$ at the end of Algorithm 2. Hence, the optimal solution x computed for Problem (5) in Line 35 satisfies

$$x_{h_0,d} = 0, \ d \in D_{\max}, \quad \text{and} \quad x_{h,d_0} = 0, \ h \in H_{\max}.$$

Let us define a solution $\hat{x}$ as

$$\hat{x}_{h,d} = \begin{cases} 1, & \text{if } h = h_0 \text{ and } d = d_0, \\ x_{h,d}, & \text{otherwise.} \end{cases}$$

Since M' is a feasible matching, $\hat{x}$ is feasible for Problem (5) and

$$\sum_{\{h,d\} \in E_{\max}} w_{h,d} \ \hat{x}_{h,d} = \sum_{\{h,d\} \in E_{\max}} w_{h,d} \ x_{h,d} + w_{h_0,d_0},$$

which contradicts that x is an optimal solution. This concludes the proof. □

Algorithm 2: Regional Decomposition

Input: Problem (5), $\bar{D}, A_{\min} \in \mathbb{N}$, $\alpha, \gamma \in [0,1]$, $\beta \in (0,1]$.
Output: A feasible point $\hat{x}$ for Problem (5).

Set $\hat{x}_{h,d} \leftarrow 0$.
while $|D_{\max}| > \bar{D}$ **and** $|H_{\max}| > 0$ **do**
  Decompose $D_{\max}$ into $D^1_{\max}, \ldots, D^n_{\max}$ so that $|D^i_{\max}| \leq \bar{D}$ holds for all $i = 1, \ldots, n$.
  Decompose $H_{\max}$ into $H^1_{\max}, \ldots, H^n_{\max}$ by solving Problem (6).
  Set $A \leftarrow 0$.
  for $i = 1, \ldots, n$ **do**
    Solve Problem (7) and let x^i denote the optimal solution.
    Set $\hat{H}^i_{\max} \leftarrow \{h \in H^i_{\max} : x^i_{h,d} = 1, w_{h,d} \geq \alpha \text{ for some } d \in D^i_{\max}\}$.
    Set $\hat{D}^i_{\max} \leftarrow \{d \in D^i_{\max} : x^i_{h,d} = 1, w_{h,d} \geq \alpha \text{ for some } h \in H^i_{\max}\}$.
    Set $H_{\max} \leftarrow H_{\max} \setminus \hat{H}^i_{\max}$ and $D_{\max} \leftarrow D_{\max} \setminus \hat{D}^i_{\max}$.
    Set $A \leftarrow A + |\hat{H}^i_{\max}|$.
    for d **in** $\hat{D}^i_{\max}$ **do**
      Set $h_d \leftarrow h \in \hat{H}^i_{\max}$ so that $x^i_{h,d} = 1$ and $k_d \leftarrow k \in K$ so that $s_{d,k} = 1$.
      if $k_d \in K^{hhd}$ **then**
        Set $B^{\text{hhd}}_{k_d} \leftarrow B^{\text{hhd}}_{k_d} - 1$.
      end
      if $k_d \in K^{per}$ **then**
        Set $B^{\text{per}}_{k_d} \leftarrow B^{\text{per}}_{k_d} - p_{h_d}$.
      end
      Set $\hat{x}_{h_d,d} \leftarrow 1$.
    end
  end
  if $A \leq A_{\min}$ **then**
    if $\alpha = 0$ **and** $A_{\min} = 0$ **then**
      Stop.
    else
      Set $\alpha \leftarrow \alpha - \beta$.
      if $\alpha \leq \gamma$ **then**
        Set $\alpha \leftarrow 0$ and $A_{\min} \leftarrow 0$.
      end
    end
  end
end
if $|H_{\max}|, |D_{\max}| > 0$ **then**
  Solve Problem (5) and let x denote the optimal solution.
  for h **in** $H_{\max}$ **do**
    if $\exists d \in D_{\max}$ *with* $x_{h,d} = 1$ **then**
      Set $\hat{x}_{h,d} \leftarrow 1$.
    end
  end
end

Theorem 3. *If Algorithm 2 is used to compute a feasible point for the instances of Problem (5), the output of Algorithm 1 is an SCM matching for Problem (2).*

Proof. Since Algorithm 2 terminates after finitely many iterations, the same applies to Algorithm 1. Let M be the matching obtained by Algorithm 1. Its SCM property is again shown by contradiction. Let us assume that M is not an SCM matching. Then, there exists $\{h_0, d_0\} \in E \setminus M$ such that $M' = M \cup \{\{h_0, d_0\}\}$ is a feasible matching for Problem (2). As explained at the beginning of Section 3.1, given any $h \in H$ and $d \in D$, if $\{h, d\} \in E$ then $p_h \leq c_d$ holds, which implies $p_{h_0} \leq c_{d_0}$. Consequently, this possible assignment is considered in Problem (5) for some iteration of Algorithm 1. In this iteration, the matching obtained in Line 6 does not satisfy the SCM property and, thus, it is not optimal for Problem (5), which can be shown in the same way as in the last proof. Thus, this matching is obtained by Algorithm 2, which contradicts Theorem 2 and the proof is complete. □

Note that the latter theorem holds true if Algorithm 2 is replaced by any other method that computes an SCM matching for Problem (5).

4. A Lagrangian-Relaxation-Based Approximation Method

Problem (2) is an MWM problem with side constraints (2d) and (2e). Without the latter, mainly due to the total unimodularity property, there would be many attractive solution strategies such as LP-based (Schrijver 1986, Chapter 19) or graph-based techniques, like the Hungarian algorithm and the Edmonds–Karp algorithm (Munkres 1957; Edmonds and Karp 1972).

In this section, we first present a Lagrangian-relaxation-based (LR-based) reformulation of Problem (2) and then prove that this reformulation can be written in the form (1) for any vector of Lagrange multipliers. However, we do not solve the min-max problem associated to the Lagrangian relaxation since this is too costly for the size of the considered problems. Instead, we derive an iterative method for only adjusting the multipliers without explicitly considering the dual problem. In every iteration of this method, a problem of the form (1) is considered. In particular, this means that these problems are still as large as the original one, which is why we resort to approximation algorithms to tackle these problems. Finally, we prove that the approximation guarantee for these sub-problems yields an approximation guarantee for the overall problem.

4.1. A Lagrangian-Relaxation-Based Reformulation.

Let $\lambda \in \mathbb{R}^{|K^{\mathrm{hhd}}|+|K^{\mathrm{per}}|}$ be the vector of multipliers λ_k^{hhd} for all $k \in K^{\mathrm{hhd}}$ and λ_k^{per} for all $k \in K^{\mathrm{per}}$. With this notational convention, we consider the formulation

$$\max_x \quad f_\lambda(x) \tag{9a}$$

$$\text{s.t.} \quad \sum_{d:\{h,d\}\in E} x_{h,d} \leq 1, \qquad h \in H, \tag{9b}$$

$$\sum_{h:\{h,d\}\in E} x_{h,d} \leq 1, \qquad d \in D, \tag{9c}$$

$$x_{h,d} \in \{0,1\}, \qquad \{h,d\} \in E, \tag{9d}$$

where $f_\lambda : \{0,1\}^{|E|} \to \mathbb{R}$ is a function defined by

$$f_\lambda(x) = \sum_{\{h,d\}\in E} w_{h,d}\ x_{h,d} - \sum_{k\in K^{\text{hhd}}} \lambda_k^{\text{hhd}} \left(\sum_{\{h,d\}\in E} s_{d,k}\ x_{h,d} - B_k^{\text{hhd}} \right) - \sum_{k\in K^{\text{per}}} \lambda_k^{\text{per}} \left(\sum_{\{h,d\}\in E} p_h\ s_{d,k}\ x_{h,d} - B_k^{\text{per}} \right)$$

with $\lambda_k^{\text{hhd}}, \lambda_k^{\text{per}} \geq 0$. In this problem, λ is of crucial importance. If the values of λ are close to zero, the optimal solution x^λ to Problem (9) can easily violate the side constraints of Problem (2). If these values are too large, x^λ tends to satisfy these constraints strictly, which generates a considerable difference between both objective functions. Therefore, our iterative approach is designed to find solutions with a good objective value for Problem (2) for λ-values not being too large but still guaranteeing feasibility of the computed points.

Before we discuss the mentioned iterative procedure, we first show that Problem (9) can be written in the form of an MWM (1).

Lemma 1. *Let $\lambda_k^{hhd} \geq 0$ for all $k \in K_{hhd}$ and $\lambda_k^{per} \geq 0$ for all $k \in K^{per}$ be given. Then, Problem (9) is equivalent to*

$$\max_x \quad \sum_{\{h,d\}\in E} \chi_{h,d}\ x_{h,d} \tag{10a}$$

$$s.t. \quad \sum_{d:\{h,d\}\in E} x_{h,d} \leq 1, \qquad h \in H, \tag{10b}$$

$$\sum_{h:\{h,d\}\in E} x_{h,d} \leq 1, \qquad d \in D, \tag{10c}$$

$$x_{h,d} \in \{0,1\}, \qquad \{h,d\} \in E, \tag{10d}$$

with

$$\chi_{h,d} = \begin{cases} w_{h,d}, & \text{if } s_{d,k} = 1 \text{ for } k \notin K^{hhd} \cup K^{per}, \\ w_{h,d} - \lambda_k^{hhd}, & \text{if } s_{d,k} = 1 \text{ for } k \in K^{hhd} \setminus K^{per}, \\ w_{h,d} - \lambda_k^{per} p_h, & \text{if } s_{d,k} = 1 \text{ for } k \in K^{per} \setminus K^{hhd}, \\ w_{h,d} - \lambda_k^{hhd} - \lambda_k^{per} p_h, & \text{if } s_{d,k} = 1 \text{ for } k \in K^{hhd} \cap K^{per}. \end{cases}$$

Proof. The function f_λ in Problem (9) can be written as

$$f_\lambda(x) = \sum_{\{h,d\}\in E} w_{h,d}\ x_{h,d} - \sum_{k\in K^{\text{hhd}}} \sum_{\{h,d\}\in E} \lambda_k^{\text{hhd}}\ s_{d,k}\ x_{h,d} + \sum_{k\in K^{\text{hhd}}} \lambda_k^{\text{hhd}}\ B_k^{\text{hhd}} - \sum_{k\in K^{\text{per}}} \sum_{\{h,d\}\in E} \lambda_k^{\text{per}}\ p_h\ s_{d,k}\ x_{h,d} + \sum_{k\in K^{\text{per}}} \lambda_k^{\text{per}} B_k^{\text{per}}. \tag{11}$$

Since each dwelling is contained in exactly one grid cell, by the definition of $s_{d,k}$ we can write

$$\sum_{\{h,d\}\in E} \sum_{k\in K^{\text{hhd}}} \lambda_k^{\text{hhd}}\ s_{d,k}\ x_{h,d} =: \sum_{\{h,d\}\in E} \omega_d^{\text{hhd}}\ x_{h,d} \tag{12}$$

and

$$\sum_{\{h,d\}\in E} \sum_{k\in K^{\text{per}}} \lambda_k^{\text{per}}\ p_h\ s_{d,k}\ x_{h,d} =: \sum_{\{h,d\}\in E} \omega_d^{\text{per}}\ x_{h,d}, \tag{13}$$

with

$$\omega_d^{\text{hhd}} = \begin{cases} \lambda_k^{\text{hhd}}, & \text{if } s_{d,k} = 1 \text{ for } k \in K^{\text{hhd}}, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\omega_d^{\text{per}} = \begin{cases} \lambda_k^{\text{per}}\ p_h, & \text{if } s_{d,k} = 1 \text{ for } k \in K^{\text{per}}, \\ 0, & \text{otherwise.} \end{cases}$$

Substituting (12) and (13) in (11) leads to

$$\begin{aligned} f_\lambda(x) = &\sum_{\{h,d\}\in E} w_{h,d}x_{h,d} - \sum_{\{h,d\}\in E} \omega_d^{\text{hhd}} x_{h,d} - \sum_{\{h,d\}\in E} \omega_d^{\text{per}} x_{h,d} \\ &+ \sum_{k\in K^{\text{hhd}}} \lambda_k^{\text{hhd}} B_k^{\text{hhd}} + \sum_{k\in K^{\text{per}}} \lambda_k^{\text{per}} B_k^{\text{per}}. \end{aligned} \tag{14}$$

Since the last two terms in (14) are constant, they can be omitted and we obtain

$$\begin{aligned} \max_x \quad & \sum_{\{h,d\}\in E} (w_{h,d} - \omega_d^{\text{hhd}} - \omega_d^{\text{per}}) x_{h,d} \\ \text{s.t.} \quad & \sum_{d:\{h,d\}\in E} x_{h,d} \le 1, && h \in H, \\ & \sum_{h:\{h,d\}\in E} x_{h,d} \le 1, && d \in D, \\ & x_{h,d} \in \{0,1\}, && \{h,d\} \in E, \end{aligned}$$

from which the result follows. □

4.2. **Approximation Guarantees.** Although the structure of Model (10) is simpler than the one of Problem (2), the number of variables is still the same and this still renders the application of exact methods generally impossible. As a remedy, approximation algorithms can be used to obtain a feasible matching (Drake and Hougardy 2003; Preis 1999). These methods usually have a faster running time and additionally guarantee a certain quality of the obtained points. In our setup, the latter guarantee is of the form

$$\sum_{e\in M^a} w_e \ge \eta \sum_{e\in M^*} w_e, \tag{15}$$

where M^* is the maximum weight matching and M^a is the matching computed by the approximation algorithm with $\eta \in (0,1]$ being the guaranteed approximation factor.

Before we move on, let us briefly comment on the fact that Problem (10) might have objective function coefficients $\chi_{h,d}$ that are non-positive. Obviously, these edges will never appear in any optimal solution. Hence, from now on, for a given vector of multipliers λ we always consider Problem (10) on the graph $G(\lambda) = (H \cup D, E(\lambda))$ with $E(\lambda) = E \setminus \{\{h,d\} \in E : \chi_{h,d} \le 0\}$. We further assume that $E(\lambda) \neq \emptyset$ always holds. The coefficients $\chi_{h,d}$ are functions of λ so that the notation makes sense. Moreover, the restriction to this sub-graph also implies that all optimal solutions to Problem (10) have positive objective function values.

We now prove that the application of an approximation algorithm to the instances of Problem (10) implies an approximation guarantee similar to (15) for the original Problem (2).

Theorem 4. *Let $\lambda > 0$ be given and fixed and consider Problem (10) on the graph $G(\lambda)$. Moreover, let $\tilde{x}^{\lambda,a} \in \{0,1\}^{|E(\lambda)|}$ be the point computed by an approximation algorithm with approximation factor $\eta \in (0,1]$ for Problem (10) and define*

$x^{\lambda,a} \in \{0,1\}^{|E|}$ by extending $\tilde{x}^{\lambda,a}$ with zeros for all edges $e \in E \setminus E(\lambda)$. Then,

$$\begin{aligned}
&\sum_{\{h,d\}\in E} w_{h,d}\ x^*_{h,d} - \sum_{\{h,d\}\in E} w_{h,d}\ x^{\lambda,a}_{h,d} \\
&\leq\ (1-\eta) \sum_{\{h,d\}\in E} w_{h,d}\ x^*_{h,d} - \sum_{k\in K^{hhd}} \lambda^{hhd}_k \left(\sum_{\{h,d\}\in E} s_{d,k}\ x^{\lambda,a}_{h,d} - B^{hhd}_k \right) \\
&\quad - \sum_{k\in K^{per}} \lambda^{per}_k \left(\sum_{\{h,d\}\in E} p_h\ s_{d,k}\ x^{\lambda,a}_{h,d} - B^{per}_k \right)
\end{aligned}$$

holds, where x^* is an optimal solution of Problem (2).

Proof. Let $\tilde{x}^{\lambda,*}$ be an optimal solution of Problem (10) on the graph $G(\lambda)$ and define $x^{\lambda,*}$ by extending $\tilde{x}^{\lambda,*}$ with zeros for all edges $e \in E \setminus E(\lambda)$. The approximation guarantee of the approximation method yields

$$\frac{\sum_{\{h,d\}\in E} \chi_{h,d}\ x^{\lambda,a}_{h,d}}{\sum_{\{h,d\}\in E} \chi_{h,d}\ x^{\lambda,*}_{h,d}} = \frac{\sum_{\{h,d\}\in E(\lambda)} \chi_{h,d}\ \tilde{x}^{\lambda,a}_{h,d}}{\sum_{\{h,d\}\in E(\lambda)} \chi_{h,d}\ \tilde{x}^{\lambda,*}_{h,d}} \geq \eta. \tag{16}$$

Note that optimal solutions always have positive objective function values and, hence, the expression in (16) is well-defined. We now define

$$B := \sum_{k\in K^{\mathrm{hhd}}} \lambda^{\mathrm{hhd}}_k\ B^{\mathrm{hhd}}_k + \sum_{k\in K^{\mathrm{per}}} \lambda^{\mathrm{per}}_k\ B^{\mathrm{per}}_k$$

and we obtain

$$f_\lambda(x) = \sum_{\{h,d\}\in E} \chi_{h,d}\ x_{h,d} + B.$$

Moreover, it holds

$$\frac{f_\lambda(x^{\lambda,a})}{f_\lambda(x^{\lambda,*})} = \frac{\sum_{\{h,d\}\in E} \chi_{h,d}\ x^{\lambda,a}_{h,d} + B}{\sum_{\{h,d\}\in E} \chi_{h,d}\ x^{\lambda,*}_{h,d} + B} \geq \frac{\sum_{\{h,d\}\in E} \chi_{h,d}\ x^{\lambda,a}_{h,d}}{\sum_{\{h,d\}\in E} \chi_{h,d}\ x^{\lambda,*}_{h,d}} \geq \eta. \tag{17}$$

By Lemma 1, $x^{\lambda,*}$ is also an optimal solution to (9) and since x^* is feasible for this problem, we obtain

$$\sum_{\{h,d\}\in E} w_{h,d}\ x^*_{h,d} \leq f_\lambda(x^*) \leq f_\lambda(x^{\lambda,*}),$$

which, in turn, implies

$$\frac{f_\lambda(x^{\lambda,a})}{\sum_{\{h,d\}\in E} w_{h,d}\ x^*_{h,d}} \geq \frac{f_\lambda(x^{\lambda,a})}{f_\lambda(x^{\lambda,*})}. \tag{18}$$

Thus, by (17) and (18) we have

$$\frac{f_\lambda(x^{\lambda,a})}{\sum_{\{h,d\}\in E} w_{h,d} x^*_{h,d}} \geq \eta.$$

Using the definition of f_λ again, we finally obtain

$$\begin{aligned}
&\sum_{\{h,d\}\in E} w_{h,d}\; x^{\lambda,a}_{h,d} \\
&\geq \eta \sum_{\{h,d\}\in E} w_{h,d}\; x^{*}_{h,d} + \sum_{k\in K^{\text{hhd}}} \lambda^{\text{hhd}}_k \left(\sum_{\{h,d\}\in E} s_{d,k}\; x^{\lambda,a}_{h,d} - B^{\text{hhd}}_k \right) \\
&\quad + \sum_{k\in K^{\text{per}}} \lambda^{\text{per}}_k \left(\sum_{\{h,d\}\in E} p_h\; s_{d,k}\; x^{\lambda,a}_{h,d} - B^{\text{per}}_k \right),
\end{aligned}$$

from which the result follows. □

Algorithm 3: An LR-Based Iterative Approximation Algorithm

Input: Problem (2), an initial vector multipliers $\lambda \in (0,1]^{|K^{\text{hhd}}|+|K^{\text{per}}|}$, and an update factor $\zeta > 1$.

Output: An approximate solution $\hat{x}$ to Problem (2).

```
Initialize Violation ← True.
while Violation = True do
    Set Violation ← False.
    Approximately solve Problem (10) on the graph G(λ) for the current λ
     and let x̃^{λ,a} ∈ {0,1}^{|E(λ)|} denote the approximate solution that we
     extend to x^{λ,a} ∈ {0,1}^{|E|} by inserting zeros for all edges e ∈ E \ E(λ).
    for k in K^hhd do
        if Σ_{h,d}∈E s_{d,k} x^λ_{h,d} > B^hhd_k then
            Update λ^hhd_k ← ζ λ^hhd_k and set Violation ← True.
        end
    end
    for k in K^per do
        if Σ_{h,d}∈E p_h s_{d,k} x^λ_{h,d} > B^per_k then
            Update λ^per_k ← ζ λ^per_k and set Violation ← True.
        end
    end
end
Set x̂ ← x^{λ,a}.
```

4.3. **The Iterative Method.** The iterative method based on Lagrangian relaxations of the problem and their approximate solutions is given in Algorithm 3. In the light of this iterative method, the inequality provided by Theorem 4 is particularly interesting for our application. If $x^{\lambda,a}$ is an approximate solution to Problem (10) in the last iteration of the method, the number of households and persons allocated in the grid cells in K^{hhd} and K^{per} tend to be close to their upper bounds B^{hhd}_k and B^{per}_k. This implies that the values

$$\sum_{k\in K^{\text{hhd}}} \lambda^{\text{hhd}}_k \left(\sum_{\{h,d\}\in E} s_{d,k}\; x^{\lambda,a}_{h,d} - B^{\text{hhd}}_k \right), \quad \sum_{k\in K^{\text{per}}} \lambda^{\text{per}}_k \left(\sum_{\{h,d\}\in E} p_h\; s_{d,k}\; x^{\lambda,a}_{h,d} - B^{\text{per}}_k \right)$$

are small in comparison to the optimal objective value of Problem (2). By ignoring these small terms in the inequality of the theorem, we see that we obtain the same approximation guarantee η as in the original approximation method (15).

Let us close this section with a final remark. Obviously, there are many different approximation algorithms that one could choose to approximately solve the MWM problem in every iteration of Algorithm 3. For our implementation, we use the path-growing algorithm (Drake and Hougardy 2003) due to three reasons. First, the time complexity of this algorithm is $O(|E|)$, which is, to the best of our knowledge, the best available for MWM problems. Second, it has a good approximation guarantee of $\eta = 1/2$, which is, e.g., the same as for Greedy-type methods. Third, it is possible to warm-start the path-growing algorithm in every iteration based on the approximate solution obtained in the iteration before. Since the details depend on specific aspects of the path-growing algorithm and since using the latter is not at the core of the contribution of this paper, we omit the details.

A matching obtained from the path-growing algorithm does not need to be maximal in the corresponding graph. Therefore, in each iteration of Algorithm 3, we incorporated a post-processing step that guarantees that we obtain a maximal matching w.r.t. Problem (10). In this post-processing step, each household that is not assigned by the path-growing algorithm is processed and the corresponding available edge with the largest weight is used for the assignment. This enhances the objective value in each iteration of the method. However, it does not guarantee that the output of the overall method is an SCM matching for Problem (2).

5. Computational Study

Section 5.1 and 5.2 contain the details of the real-world and the synthetic data sets used in the computational study, as well as the details about the construction of the side constraints. In Section 5.3, a description of the hardware and software setup is given. Finally, we discuss the results in Section 5.4 and 5.5.

The focus of the computational study is on the comparison between the decomposition approach presented in Section 3, the Lagrangian-relaxation-based approximation method (LRBAM) as discussed in Section 4, and the direct application of a MILP solver. The performance of each method is evaluated by considering its run time, its memory usage, and the quality of the obtained points. Further discussions concern how the characteristics of the instances affect the performance of each method.

5.1. **Real-World Data Sets.**

The real-world data used in the study represent the city of Trier, in the federal state of Rhineland-Palatinate, Germany, with 103 100 inhabitants and 20 701 residential buildings according to the Census 2011. The dwelling data set is developed by Reiter (2021, Chapter 4) as an extension to a building data set (Weymeirsch et al. 2024). In this data set, each of the 52 709 dwellings contains information on its capacity, i.e., the maximum number of people that can properly live in it, along with the precise geo-coordinates (X, Y) of its location, which implies its assignment to a specific grid cell of the region.

The household data set used in this study has been synthetically generated from the Census 2011 data as part of the MikroSim project.[1] For this paper, we focus on a subset of the data associated with the city of Trier, comprising 49 109 households. The central information of this data set concerns the size of each household, i.e., the number of persons in the household. Therefore, as explained at the beginning of Section 3.1, given a household $h \in H$ and a dwelling $d \in D$, the edge $\{h, d\}$ only exists if $p_h \leq c_d$. Similarly to Reiter (2021, Chapter 6), the weight of each edge is computed as

$$w_{h,d} = \frac{1}{1 + c_d - p_h}. \tag{19}$$

[1] https://mikrosim.uni-trier.de/de/

To improve memory resource usage in numerical experiments, we do not consider edges $\{h, d\}$ such that $w_{h,d} < 0.2$. This data set is called Trier in what follows.

To evaluate the effect of the data set sizes in the treatment of the HAP by the proposed methods, we carry out further numerical experiments based on two additional subsets of the Trier data set: one with 14 752 households and 15 839 dwellings, and another one containing 24 600 households and 26 428 dwellings. These data sets are called Trier_30 and Trier_50, respectively, since they correspond to approximately 30 % and 50 % of the Trier data set. For each of the three real-world data sets, two different instances of the HAP are analyzed: one with the side constraints (2d) and (2e), and the other one without these constraints. In those instances that include the constraints, the sets K^{hhd} and K^{per} comprise the five grid cells with the highest number of residential dwellings and the five grid cells with the largest total dwelling capacities, respectively. The corresponding upper bounds for such constraints are set to

$$B_k^{\text{hhd}} = \left\lceil 0.8 \cdot \sum_{d \in D : c_d \neq 0} s_{d,k} \right\rceil \tag{20}$$

and

$$B_k^{\text{per}} = \left\lceil 0.8 \cdot \sum_{d \in D} c_d s_{d,k} \right\rceil . \tag{21}$$

5.2. **Instances of Synthetic Data Sets.** Since statistical information on households and dwellings is used to define the corresponding edge weights, a larger variability of the given statistical information in the data sets implies a larger variability of the possible weight values. To analyze how this affects the final solution and the performance of the proposed methods, we generate further synthetic data sets using Gaussian Mixture Models. The approach used here is inspired by the one by Reiter (2021, Chapter 5) but differs in two relevant aspects. The first one is the insertion of new types of statistical information in the data sets. The second one is the creation of a workplace data set and an address data set as an intermediate step, which allows the geographical distribution of the household and dwelling data sets to resemble the one observed in urban regions. We do not go into the details here but refer to Appendix A. With the synthetically generated data sets, we then consider all instances obtained from each possible combination of a synthetically generated household and the synthetically generated dwelling data set. For each one, an instance of the HAP is defined considering K^{hhd} as the n grid cells with the largest number of residential dwellings and K^{per} as the n grid cells with the largest total dwelling capacities for $n \in \{0, 3, 5, 10, 20\}$. For each instance, the values B_k^{hhd} and B_k^{per} are defined by the expressions (20) and (21) for all $k \in K^{\text{hhd}}$ and $k \in K^{\text{per}}$, respectively.

5.3. **Software and Hardware Setup.** We compare three different methods:

(1) The decomposition approach with the parameters of Algorithm 2 set to $\bar{D} = 4000$, $A_{\min} = 100$, $\alpha = 0.7$, $\beta = 0.2$, and $\gamma = 0.3$.
(2) The LRBAM with $\lambda_k^{\text{hhd}} = 0.1$ for all $k \in K^{\text{hhd}}$, $\lambda_k^{\text{per}} = 0.06$ for all $k \in K^{\text{per}}$, and $\zeta = 1.1$.
(3) The direct application of Gurobi (version 10.0.3) to Problem (3).

All of the above methods are implemented in Python 3.6.8. The solver Gurobi is also used in Lines 7 and 35 of Algorithm 2, and in Line 6 of Algorithm 1 (if this is applicable, otherwise Algorithm 2 is used in Line 6 of Algorithm 1). The maps shown in this work are built using QGIS 3.28.11 (QGIS Development Team

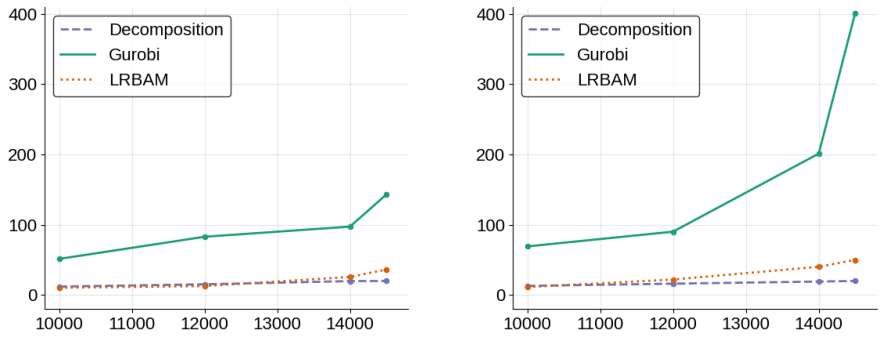

FIGURE 2. Run time (in minutes; y-axis) vs. number of households (x-axis). Left: K^{hhd} and K^{per} with 5 grid cells each. Right: K^{hhd} and K^{per} with 10 grid cells each.

2024). In particular, for the real-world data sets, a base map by OpenStreetMap (OpenStreetMap contributors 2017) is used.

Given the significant difference in computational resource requirements between the experiments for the synthetic data sets and for the real-world data sets, distinct computational settings are selected for each scenario. For the synthetic data sets, all the computations are executed on an Intel XEON SP 6126 at 2.6 GHz using a maximum 24 cores and 360 GB RAM. For the real-world data sets, we use an AMD EPYC 9754 at 2.25 GHz using a maximum of 32 cores. Here, for the decomposition approach and the LRBAM we use 500 GB RAM, while we use 700 GB RAM when we directly apply Gurobi. We always set a time limit of 48 h.

5.4. **Numerical Results for the Synthetic Data Sets.**

5.4.1. *Run Time.* If one fixes the number of dwellings in the synthetic data sets, the resulting instances usually get harder to solve when the number of households is increased, which can be seen in Figure 2. It can be observed that Gurobi is significantly slower than the two methods and its run time is much more impacted by the number of households. This could have been expected for two reasons. First, Gurobi is designed to find an optimal solution, whereas both proposed approaches focus on "only" finding good feasible points. Second, unlike the other methods, Gurobi must consider all the variables of the problem simultaneously, making it more sensitive to the increase in the number of households.

Figure 2 also shows that there is an increase in run time of the methods if more side constraints are considered. In particular, there is a much larger increase for the run time of Gurobi than for the two other methods when the number of grid cells in K^{hhd} and K^{per} raises from 5 (left plot) to 10 (right plot). Moreover, Figure 3 also shows that the run time increases in dependence on the number of side constraints—although Gurobi has a less monotonic behavior w.r.t. this parameter. Hence, although the inclusion of side constraints raises the complexity of the problem, in some cases more of these constraints seem to be beneficial for Gurobi. The latter can be seen when the number of grid cells in K^{hhd} and K^{per} increases from 0 to 3, with a reduction of 12.58 min, and from 10 to 20, with a reduction of 57.6 min in run time. Figure 3 additionally indicates that the increase in the number of side constraints implies an increase in the LRBAM run time because a larger number of iterations is needed. Finally, the decomposition approach does not seem to be severely impacted by these parameters and is the fastest method.

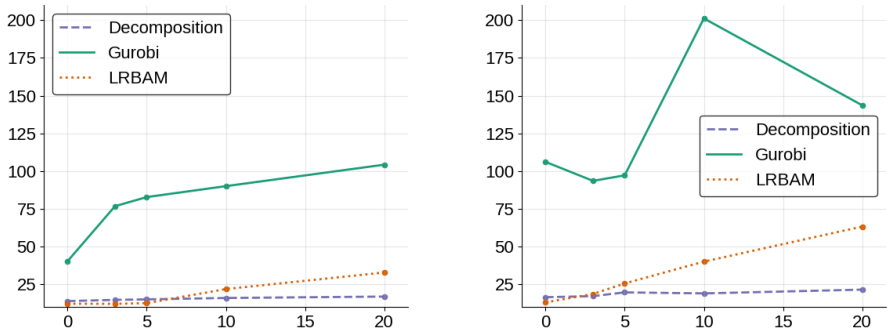

FIGURE 3. Run time (in minutes) vs. number of grid cells in K^{hhd} and K^{per}. Left: 12 000 households. Right: 14 000 households.

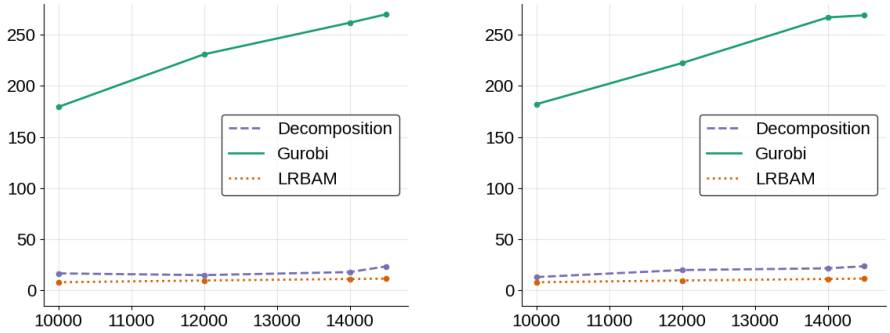

FIGURE 4. Memory usage (in GB RAM) vs. number of households. Left: K^{hhd} and K^{per} with 5 grid cells each. Right: K^{hhd} and K^{per} with 10 grid cells each.

5.4.2. *Memory Usage.* As already mentioned, the size of the considered MILP models is huge, leading to enormous memory requirements for Gurobi. Note that while Gurobi is not used in LRBAM, it is used in the decomposition approach, where the size of the Gurobi models is controlled in Line 3 of Algorithm 2. The most memory-consuming step of the two newly proposed methods is storing of the $w_{h,d}$-values for all $\{h,d\} \in E$, while for the direct application of Gurobi, the required memory is much larger. Moreover, the memory requirement of Gurobi also increases for larger number of households. This is confirmed by Figure 4.

A similar trend can also be observed w.r.t. the number of side constraints (2d) and (2e) in an instance, as shown in Figure 5. The increase in the number of these constraints affects the effort needed by the solving process of Gurobi, and thus tends to increase its memory usage.

5.4.3. *Quality of the Final Point.* In all our numerical experiments, the two newly proposed methods obtain objective values that are very close to the optimal one. This is shown in Figure 6, which shows the particular cases in which K^{hhd} and K^{per} contain 5 (left plot) and 10 (right plot) grid cells each. Specifically, it can also be seen that even in the hardest instances, where the difference between the number of

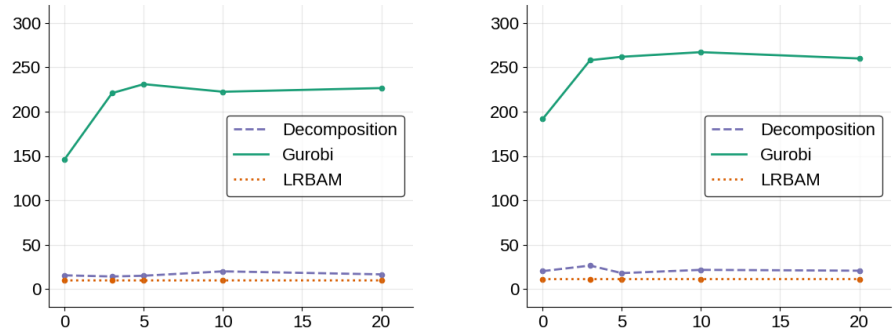

FIGURE 5. Memory usage (in GB RAM) vs. number of grid cells in K^{hhd} and K^{per}. Left: 12 000 households. Right: 14 000 households.

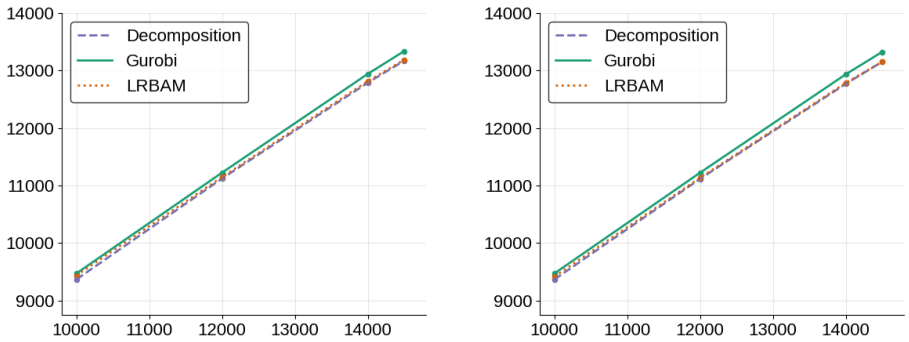

FIGURE 6. Objective value vs. number of households. Left: K^{hhd} and K^{per} with 5 grid cells each. Right: K^{hhd} and K^{per} with 10 grid cells each.

dwellings and the number of households is relatively small, only a slight increase occurs in the difference between each objective value.

In all the computed allocations, most of the addresses have an average weight of assignments between 0.8 and 1, which highlights the geographical quality of the allocations obtained. This is shown in Figure 7 and 8 for specific instances. In particular, Figure 7 shows that the number of addresses with an average weight between 0 and 0.2 is strictly related to the difference between the number of dwellings and the number of households in the data set.

5.5. **Numerical Results for Real-World Data Sets.** The instances corresponding to the real-world data sets have significantly more variables than those associated with the synthetic data sets. As a result, the increase in the size of the bipartite graph implies an important change in the run time of the LRBAM, as shown in Table 1 and 2. Without side constraints, where the LRBAM terminates in one iteration, it is slower than the decomposition approach but considerably faster than Gurobi for all instances. However, the inclusion of side constraints implies that more iterations are needed in the LRBAM to find a feasible point. Thus, the need to deal with a very large graph in every iteration increases its run time, which makes it

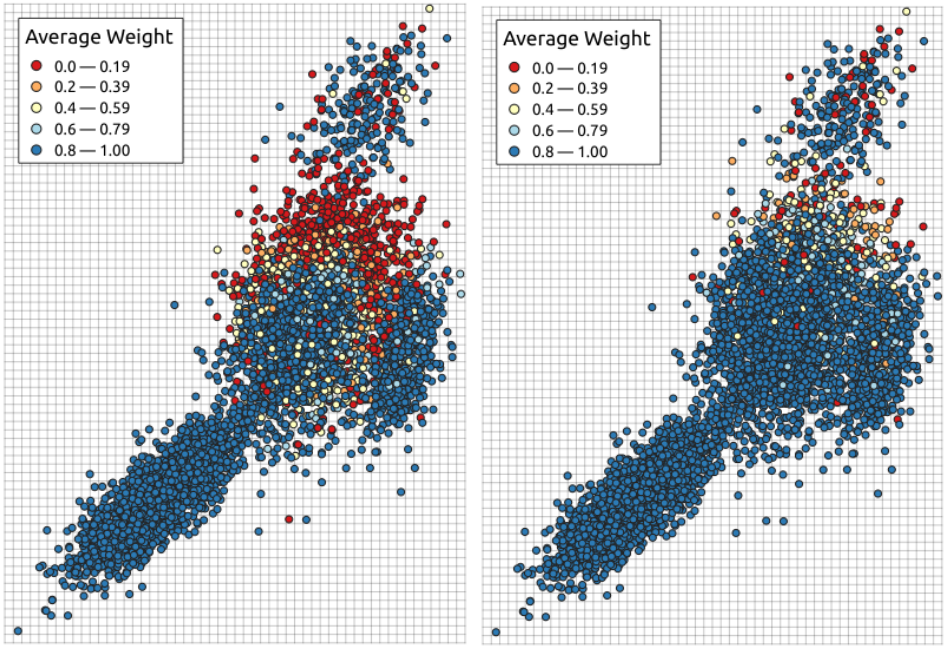

FIGURE 7. Average weight per address in the allocation computed by Gurobi for K^{hhd} and K^{per} with 10 grid cells each. The grid cells shown in the plot actually correspond to the grid cells of the model. Left: 12 000 households. Right: 14 000 households.

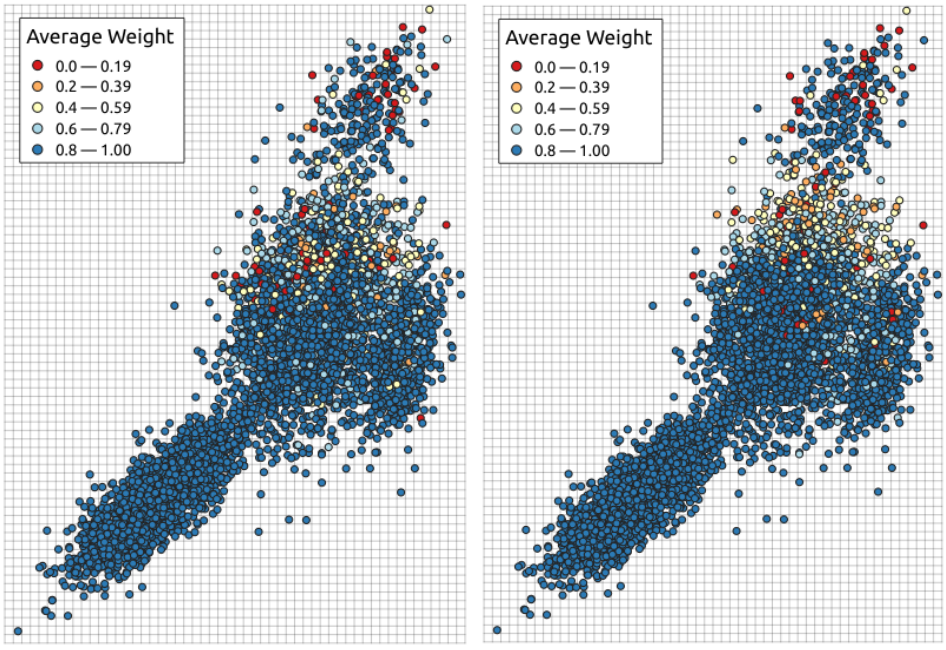

FIGURE 8. Average weight per address considering 14 000 households for K^{hhd} and K^{per} containing 10 grid cells each. The grid cells shown in the plot actually correspond to the grid cells of the model. Left: Allocation of the decomposition approach. Right: Allocation of LRBAM.

	Gurobi	LRBAM	Decomposition
Trier_30	56.35	13.85	7.89
Trier_50	167.20	38.80	16.17
Trier	–	189.51	43.64

TABLE 1. Run time (in minutes) of the methods for instances without side constraints for each of the real-world data sets.

	Gurobi	LRBAM	Decomposition
Trier_30	37.91	94.18	8.79
Trier_50	109.32	292.06	17.33
Trier	–	1410.28	45.01

TABLE 2. Run time (in minutes) of the methods for instances with side constraints for each of the real-world data sets.

	Gurobi	LRBAM	Decomposition
Trier_30	120.00	11.41	18.70
Trier_50	363.54	38.68	49.95
Trier	–	189.33	180.51

TABLE 3. Memory usage (in GB RAM) for instances of the HAP with side constraints for each of the real-world data sets.

the slowest method. In the meanwhile, the decomposition approach is not strongly affected by this factor, being the fastest approach in both situations.

An important advantage of the proposed methods in relation to Gurobi is the memory usage. As shown in Table 3, the memory needed by both proposed approaches is similar for all instances and significantly smaller than the memory usage of Gurobi. In particular, Gurobi is unable to solve the Trier data set instance with the available 700 GB RAM.

Although the proposed methods do not guarantee to find an optimal solution, Table 4 shows that the final objective value is very close to the optimal one. Let us define the relative gap between the objective value of the optimal solution x^* and the objective value of the point x obtained by one of the proposed methods as

$$\frac{\sum_{\{h,d\}\in E} w_{h,d}\ x^*_{h,d} - \sum_{\{h,d\}\in E} w_{h,d}\ x_{h,d}}{\sum_{\{h,d\}\in E} w_{h,d}\ x^*_{h,d}}.$$

For the Trier_50 data set, which corresponds to the biggest instances that Gurobi solves with 700 GB RAM, the relative difference between the optimal objective value found by Gurobi and the objective value obtained by the LRBAM and the decomposition approach are 0.15 % and 0.01 %, respectively. This suggests that the proposed methods obtain near-optimal (if not optimal) solutions. In particular, for all instances the objective value found by the decomposition approach is slightly higher than the one of the LRBAM.

Taking a closer look at the computed allocations, Figure 9 shows a pronounced symmetry in the frequency of the points outside the diagonal of the plot. The figure depicts a balanced number of households assigned with a higher weight by either of the methods compared. This explains the similarity in the total objective

	Gurobi	LRBAM	Decomposition
Trier_30	14 381.24	14 358.02	14 380.52
Trier_50	24 059.12	24 021.52	24 056.81
Trier	–	47 982.38	48 053.51

TABLE 4. Objective value obtained by each method for instances of the HAP with side constraints for each of the real-world data sets.

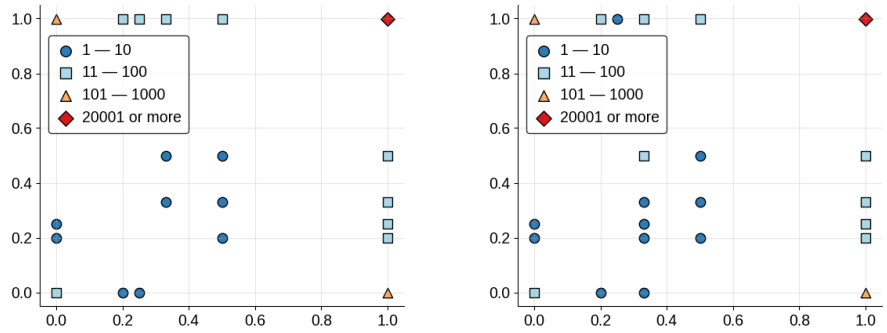

FIGURE 9. Distribution of households of Trier_50 colored by frequency considering its assignment weight by Gurobi (y-axis) and by each of the proposed methods (x-axis) for instances with side constraints. Left: decomposition approach. Right: LRBAM.

values observed in Table 4 while part of the allocation differs between the methods. Moreover, Figure 9 also shows that more than 75 % of the households in Trier_50 are assigned with weight equal to 1 by Gurobi and by each proposed method. This is expected, since the computation of the weights $w_{h,d}$ by expression (19) depends only on two measures (p_h and c_d). In particular, since the assignments with a weight equal to 1 represent the case where $p_h = c_d$, this implies a trend where most of the addresses have an occupancy gap near 0. Figure 10 shows this trend also for those instances of the Trier data set which Gurobi cannot solve. These results emphasize the quality of the allocation made by the proposed methods from a demographic perspective because all constraints are respected and the still available living space is small.

6. CONCLUSION

We consider the household assignment problem as it occurs in the geo-referencing step of microsimulation models. For realistically sized instances, this problem cannot be solved to global optimality by today's most advanced commercial solvers due to its enormous memory and run time requirements. Therefore, we introduce two algorithms designed to derive approximate solutions. One is a Lagrangian-relaxation-based method in which an approximation algorithm is used in each iteration. The other one is a decomposition approach in which a feasible point of good quality is obtained by solving smaller sub-problems. We also derive theoretical results regarding the quality of the computed points for both methods.

To evaluate the performance of our methods, we generate synthetic data sets and, additionally, consider real-world datasets for the city of Trier in Germany. For all

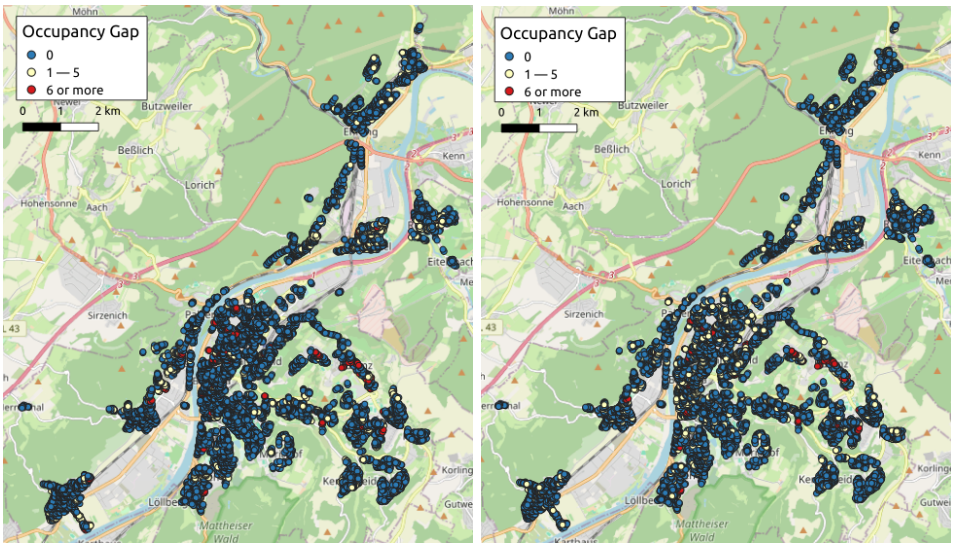

FIGURE 10. Difference between overall capacity and number of persons assigned in each address of the Trier data set for the HAP instance with side constraints. Left: decomposition approach allocation. Right: LRBAM allocation. Both images use maps by OpenStreetMap contributors (2017).

instances, both proposed approaches are less memory-consuming than the direct application of an MILP solver such as Gurobi. This aspect is even more important when considering larger areas such as, e.g., Berlin instead of Trier. Moreover, the approximate solutions obtained have objective values that are almost optimal. Concerning run times, for the synthetic data sets, both presented methods are faster than Gurobi, while for the real-world data sets, the decomposition approach clearly outperforms the other approaches. Furthermore, for the complete real-world data sets, the allocations obtained by the proposed approaches show a realistic demographic distribution.

Finally, the algorithmic ideas developed in this paper can also be applied to other matching-type problems with additional side constraints.

ACKNOWLEDGEMENTS

The authors thank the DFG for their support within RTG 2126 “Algorithmic Optimization” (Trier) and RTG 2297 “Mathematical Complexity Reduction” (Magdeburg) as well as within FOR 2559 “Multi-sectoral Regional Microsimulation Model – MikroSim” (third author). The computations were executed on the high performance cluster “Elwetritsch” at the TU Kaiserslautern, which is part of the “Alliance of High Performance Computing Rheinland-Pfalz” (AHRP). We kindly acknowledge the support of RHRK.

REFERENCES

Atkinson, T., F. Bourguignon, C. O’Donoghue, H. Sutherland, and F. Utili (2002). “Microsimulation of Social Policy in the European Union: Case Study of a European Minimum Pension.” In: *Economica* 69, pp. 229–243. DOI: 10.1111/1468-0335.00281.

Ballas, D. and G. Clarke (1999). "Modelling the Local Impacts of National Social Policies: A Spatial Microsimulation Approach." In: *11th European Colloquium on Theoretical and Quantitative Geography, Durham, England*. DOI: 10.1068/c0003.

Ballas, D., G. Clarke, and I. Turton (1999). "Exploring Microsimulation Methodologies for the Estimation of Household Attributes." In: *4th International Conference on GeoComputation, Mary Washington College, Virginia, USA*.

Basu, S. N., T. Quint, and R. Pryor (1998). "ASPEN: A Microsimulation Model of the Economy." In: *Computational Economics* 12, pp. 223–241. DOI: 10.1023/A:1008691115079.

Birkin, M. and M. Clarke (1988). "SYNTHESIS—a synthetic spatial information system for urban and regional analysis: methods and examples." In: *Environment and Planning A* 20.12, pp. 1645–1671. DOI: 10.1068/a201645.

Birkin, M., A. Turner, and B. Wu (2006). "A Synthetic Demographic Model of the UK Population : Methods , Progress and Problems." In: *Proceedings of the Second international conference on e-Social Science, NCESS, Manchester, England*.

Cullinan, J. (2010). "Developing a Continuous Space Representation of a Simulated Population." In: *Spatial Economic Analysis* 5.3, pp. 317–338. DOI: 10.1080/17421772.2010.493954.

Deming, W. E. and F. F. Stephan (1940). "On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known." In: *The Annals of Mathematical Statistics* 11.4, pp. 427–444. DOI: 10.1214/aoms/1177731829.

Drake, D. E. and S. Hougardy (2003). "A simple approximation algorithm for the weighted matching problem." In: *Information Processing Letters* 85.4, pp. 211–213. DOI: 10.1016/S0020-0190(02)00393-9.

Edmonds, J. and R. M. Karp (1972). "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems." In: *Journal of the ACM (JACM)* 19.2, pp. 248–264. DOI: 10.1145/321694.321699.

Flory, J. and S. Stöwhase (2012). "MIKMOD-ESt: A static microsimulation model of personal income taxation in Germany." In: *International Journal of Microsimulation* 5.2, pp. 66–73. DOI: 10.34196/IJM.00073.

Giortzis, A. I., L. F. Turner, and J. Barria (2000). "Decomposition technique for fixed channel assignment problems in mobile radio networks." In: *IEE Proceedings Communications* 147.3, pp. 187–194. DOI: 10.1049/ip-com:20000336.

Hägerstrand, T. (1957). "Migration and Area." In: *Migration in Sweden: a Symposium*. Lund Studies in Geography. Series B, Human Geography N. 13. Lund, Sweden: C.W.K. Gleerup.

Korte, B. and J. Vygen (2019). *Combinatorial Optimization*. Springer. DOI: 10.1007/978-3-662-56039-6.

Lovelace, R. and M. Dumont (2017). *Spatial Microsimulation with R*. Chapman and Hall/CRC. URL: https://spatial-microsim-book.robinlovelace.net.

Morrissey, K., G. Clarke, D. Ballas, S. Hynes, and C. O'Donoghue (2008). "Examining access to GP services in rural Ireland using microsimulation analysis." In: *Area* 40.3, pp. 354–364. DOI: 10.1111/j.1475-4762.2008.00844.x.

Morrissey, K. and C. O'Donoghue (2011). "The Spatial Distribution of Labour Force Participation and Market Earnings at the Sub-National Level in Ireland." In: *Review of Economic Analysis* 3.1, pp. 80–101. DOI: 10.15353/rea.v3i1.1378.

Munkres, J. (1957). "Algorithms for the Assignment and Transportation Problems." In: *Journal of the Society for Industrial and Applied Mathematics* 5.1, pp. 32–38. URL: http://www.jstor.org/stable/2098689.

Münnich, R., R. Schnell, H. Brenzel, H. Dieckmann, S. Dräger, J. Emmenegger, P. Höcker, J. Kopp, H. Merkle, K. Neufang, M. Obersneider, J. Reinhold, J.

Schaller, S. Schmaus, and P. Stein (2020). "A Population Based Regional Dynamic Microsimulation of Germany: The MikroSim Model." In: *Methods, data, analyses* 15.2. DOI: 10.12758/mda.2021.03.

Noor-E-Alam, M. (2013). "Advanced Integer Linear Programming Techniques for Large Scale Grid-Based Location Problems." PhD thesis. University of Alberta. DOI: 10.7939/R3WW5S.

O'Donoghue, C., K. Morrissey, and J. Lennon (2014). "Spatial microsimulation modelling: A review of applications and methodological choices." In: *International Journal of Microsimulation* 7.1, pp. 26–75. DOI: 10.34196/IJM.00093.

OpenStreetMap contributors (2017). *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org.

Orcutt, G. H. (1957). "A New Type of Socio-Economic System." In: *The Review of Economics and Statistics* 39.2, pp. 116–123. DOI: 10.2307/1928528.

Pellegrino, S., M. Piacenza, and G. Turati (2011). "Developing a static microsimulation model for the analysis of housing taxation in Italy." In: *International Journal of Microsimulation* 4.2, pp. 73–85. DOI: 10.34196/IJM.00054.

Preis, R. (1999). "Linear Time 1/2-Approximation Algorithm for Maximum Weighted Matching in General Graphs." In: *Annual Symposium on Theoretical Aspects of Computer Science*. Springer, pp. 259–269. DOI: 10.1007/3-540-49116-3_24.

QGIS Development Team (2024). *QGIS Geographic Information System*. QGIS Association. URL: https://www.qgis.org.

Reiter, K. M. (2021). "A Weighted Matching Model for Georeferenced Microsimulations." MA thesis. Technische Universität München.

Rephann, T. and E. Holm (2004). "Economic-Demographic Effects of Immigration: Results from a Dynamic Spatial Microsimulation Model." In: *International Regional Science Review* 27.4, pp. 379–410. DOI: 10.1177/0160017604267628.

Schrijver, A. (1986). *Theory of Linear and Integer Programming*. John Wiley & Sons.

Spielauer, M. (2007). "Dynamic microsimulation of health care demand, health care finance and the economic impact of health behaviours: Survey and review." In: *International Journal of Microsimulation* 1.1, pp. 35–53. DOI: 10.34196/IJM.00005.

Statistisches Bundesamt (2024). *Anteil der Wohnkosten am verfügbaren Haushaltseinkommen*. https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Wohnen/Tabellen/eurostat-anteil-wohnkosten-haushaltseinkommen-mz-silc.html. In: Destatis. Accessed: July 11, 2024.

Weymeirsch, J., H. Dieckmann, and R. Münnich (2024). *Construction of a Georeferenced House Data Set for the City of Trier within the MikroSim Project*. Research Papers in Economics 9/24. Trier University.

Williamson, P., M. Birkin, and P. H. Rees (1998). "The Estimation of Population Microdata by Using Data From Small Area Statistics and Samples of Anonymised Records." In: *Environment and Planning Analysis* 30, pp. 785–816. DOI: 10.1068/a300785.

Wolsey, L. A. (1998). *Integer Programming*. John Wiley & Sons, Inc.

Appendix A. Generation of Synthetic Data Sets

Originally, Gaussian mixture models (GMMs) are used to define a probability distribution as weighted sums of N Gaussian distributions. The probability density function of a GMM is defined as

$$g(x) = \sum_{r=1}^{N} \theta_r f(x \mid \mu^r, \Sigma^r)$$

where $f : \mathbb{R}^m \to \mathbb{R}$ is the probability density function of the Gaussian distribution and $\theta_r \in [0,1]$ is the probability of that a given data point x belongs to the distribution given by the mean $\mu^r = (\mu_1^r, \ldots, \mu_m^r)$ and the covariance matrix

$$\Sigma^r = \begin{bmatrix} \sigma_{11}^r & \cdots & \sigma_{1m}^r \\ \vdots & \ddots & \vdots \\ \sigma_{m1}^r & \cdots & \sigma_{mm}^r \end{bmatrix} \quad \text{with} \quad \sigma_{ij}^r = \operatorname{Cov}(X_i^r, X_j^r).$$

The following steps generically describe the construction of a data set using GMM.

Step 1: Define the size of the data set and the number N of Gaussian distributions. For each $r \in \{1, \ldots, N\}$, set the probability θ_r, the mean vector μ^r, the vector of standard deviations σ^r, and the correlation matrix

$$\begin{bmatrix} \rho_{11}^r & \cdots & \rho_{1m}^r \\ \vdots & \ddots & \vdots \\ \rho_{m1}^r & \cdots & \rho_{mm}^r \end{bmatrix}.$$

Step 2: Compute the covariance matrix

$$\Sigma^r = \begin{bmatrix} (\sigma_1^r)^2 & \sigma_1^r \sigma_2^r \rho_{12}^r & \cdots & \sigma_1^r \sigma_m^r \rho_{1m}^r \\ \sigma_2^r \sigma_1^r \rho_{21}^r & (\sigma_2^r)^2 & \cdots & \sigma_2^r \sigma_m^r \rho_{2m}^r \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_m^r \sigma_1^r \rho_{m1}^r & \sigma_m^r \sigma_2^r \rho_{m2}^r & \cdots & (\sigma_m^r)^2 \end{bmatrix}$$

for all $r \in \{1, \ldots, N\}$.

Step 3: Generate a value u from the uniform distribution $U(0,1)$ and select an index ξ as

$$\xi = \begin{cases} 1, & \text{if } 0 \le u \le \theta_1, \\ 2, & \text{if } \theta_1 < u \le \theta_1 + \theta_2, \\ \quad \vdots & \\ r, & \text{if } \sum_{i=1}^{r-1} \theta_i < u \le \sum_{i=1}^{r} \theta_i, \\ \quad \vdots & \\ N, & \text{if } \sum_{i=1}^{N-1} \theta_i < u \le 1. \end{cases}$$

Step 4: Sample a data point x from the normal distribution with mean vector μ^ξ and covariance matrix Σ^ξ.

Step 5: Repeat the steps 3–5 until the amount of data set units is reached.

In our strategy, each Gaussian distribution r of the GMM is seen as an urban sub-region so that the parameters μ^r and Σ^r are used to define characteristics of the workplaces and addresses generated for this sub-region.

A GMM is used to create a workplace data set where each unit contains information about its coordinates (X, Y) and the number of main providers of the household that it contains. Afterward, this data set is extended to a household data set so that from each workplace it is generated the number of households corresponding to its number of household main providers. By doing so, each household h obtains the location coordinates of the main provider's workplace $(X, Y)_h \in \mathbb{R}^2$. In this process, the assignment of size $p_h \in \mathbb{N}$ and monthly income $\iota_h \in \mathbb{R}^+$ to the households follows a Gaussian distribution with parameters defined by the urban sub-region of the GMM that originated its main provider's workplace. For this work, a GMM with 4 sub-regions is used to create household data sets containing a number of units equal to 10 000, 12 000, 14 000, and 14 500.

The dwelling data set is created with a similar process. Initially, an address data set is created using a GMM such that each unit is equipped with its coordinates (X, Y)

and its number of dwellings. In the following, this data set is extended to a dwelling data set so that from each address it is generated the number of dwellings that it contains. This step defines the spatial coordinates $(X, Y)_d \in \mathbb{R}^2$ of each dwelling d. Finally, the values related to cost $\gamma_d \in \mathbb{R}^+$ and capacity in terms of number of persons $c_d \in \mathbb{N}$ for each dwelling are defined by a Gaussian distribution with parameters depending on the urban sub-region of the GMM that originated its address. A dwelling data set with 15 000 units is created considering a GMM with 4 sub-regions. For this data set, a grid structure is built such that each cell has side lengths equal to 100.

Remember that the set of edges E only represents possible assignments. Hence, for a household $h \in H$ and a dwelling $d \in D$, an edge $\{h, d\}$ only exists if $p_h \leq c_d$ and $\gamma_d \leq \iota_h$ holds. The weight $w_{h,d} \in (0, 1]$ then reflects the compatibility between h and d for each aspect of the available statistical information. The first aspect to consider is the relation between the size of the household p_h and the capacity of the dwelling c_d. This measure is defined as

$$w_{h,d}^{\text{per}} := \left(\frac{\delta_{\max}^{\text{per}} - (c_d - p_h)}{\delta_{\max}^{\text{per}} - \delta_{\min}^{\text{per}}} \right)^2$$

with

$$\delta_{\max}^{\text{per}} := \max\{c_d - p_h : \{h, d\} \in E\} \quad \text{and} \quad \delta_{\min}^{\text{per}} := \min\{c_d - p_h : \{h, d\} \in E\}.$$

We square the quotient to increase the impact of one unit of variation. The second aspect considered is the comparison between the monthly cost γ_d and the household's income ι_h. By considering the data provided by Statistisches Bundesamt (2024), we assume that households tend to use approximately 30 % of their income for monthly dwelling expenses. Therefore, compatibility in this case will be measured using the expression

$$w_{h,d}^{\text{inc}} := \frac{\delta_{\max}^{\text{inc}} - |0.3\iota_h - \gamma_d|}{\delta_{\max}^{\text{inc}} - \delta_{\min}^{\text{inc}}}$$

with

$$\delta_{\max}^{\text{inc}} := \max\{|0.3\iota_h - \gamma_d| : \{h, d\} \in E\},$$
$$\delta_{\min}^{\text{inc}} := \min\{|0.3\iota_h - \gamma_d| : \{h, d\} \in E\}.$$

Finally, the distance between the dwelling location $(X, Y)_d$ and the location of the household main provider's workplace $(X, Y)_h$ is considered via

$$w_{h,d}^{\text{dist}} := \frac{\delta_{\max}^{\text{dist}} - \|(X, Y)_d - (X, Y)_h\|}{\delta_{\max}^{\text{dist}} - \delta_{\min}^{\text{dist}}}$$

with

$$\delta_{\max}^{\text{dist}} := \max\{\|(X, Y)_d - (X, Y)_h\| : \{h, d\} \in E\},$$
$$\delta_{\min}^{\text{dist}} := \min\{\|(X, Y)_d - (X, Y)_h\| : \{h, d\} \in E\}.$$

The overall measure $w_{h,d}$ is then given by

$$w_{h,d} := \tau_{\text{per}} w_{h,d}^{\text{per}} + \tau_{\text{inc}} w_{h,d}^{\text{inc}} + \tau_{\text{dist}} w_{h,d}^{\text{dist}}$$

where $\tau_{\text{per}}, \tau_{\text{inc}}, \tau_{\text{dist}} \in [0, 1]$ satisfy $\tau_{\text{per}} + \tau_{\text{inc}} + \tau_{\text{dist}} = 1$. In our computations, we consider $\tau_{\text{per}} = 0.4$, $\tau_{\text{inc}} = 0.4$, and $\tau_{\text{dist}} = 0.2$. In analogy to the real-world instances, we do not consider edges $\{h, d\}$ such that $w_{h,d} < 0.15$.

(U. Friedrich) Otto von Guericke University Magdeburg, Faculty of Mathematics, Universitätsplatz 2, 39106 Magdeburg, Germany

Email address: ulf.friedrich@ovgu.de

(L. Moschen, M. Schmidt) Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany
Email address: moschen@uni-trier.de
Email address: martin.schmidt@uni-trier.de

(R. Münnich) Trier University, Economic and Social Statistics, Universitätsring 15, 54296 Trier, Germany
Email address: muennich@uni-trier.de