City Sampling for Citizens' Assemblies

Paul Gölz[1], Jan Maly[2], Ulrike Schmidt-Kraepelin[3], Markus Utke[3], Philipp C. Verpoort[4]

[1]Cornell University
[2]WU Vienna University of Economics and Business
[3]TU Eindhoven
[4]Sortition Foundation

Abstract

In *citizens' assemblies*, a group of constituents is randomly selected to weigh in on policy issues. We study a two-stage sampling problem faced by practitioners in countries such as Germany, in which constituents' contact information is stored at a municipal level. As a result, practitioners can only select constituents from a bounded number of cities ex post, while ensuring equal selection probability for constituents ex ante.

We develop several algorithms for this problem. Although minimizing the number of contacted cities is NP-hard, we provide a pseudo-polynomial time algorithm and an additive 1-approximation, both based on separation oracles for a linear programming formulation. Recognizing that practical objectives go beyond minimizing city count, we further introduce a simple and more interpretable greedy algorithm, which additionally satisfies an ex-post monotonicity property and achieves an additive 2-approximation. Finally, we explore a notion of ex-post proportionality, for which we propose two practical algorithms: an optimal algorithm based on column generation and integer linear programming and a simple heuristic creating particularly transparent distributions. We evaluate these algorithms on data from Germany, and plan to deploy them in cooperation with a leading nonprofit organization in this space.

Code — https://github.com/markus-utke/city-sampling
Extended version — https://arxiv.org/abs/2509.07557

1 Introduction

Citizens' assemblies are an emerging form of democratic participation, in which a random sample of constituents formulate policy recommendations. The random selection of assembly members, called *sortition*, gives each person an equal chance to participate and ensures that the assembly forms a cross section of the population. Citizens' assemblies have been increasing in frequency (OECD 2020). National-level examples include assemblies on same-sex marriage, abortion, and gender equality in Ireland (Courant 2021) and German assemblies on the country's global role (Mehr Demokratie 2021), nutrition (Deutscher Bundestag 2024), and disinformation (Bertelsmann 2024).

In practice, the sortition proceeds in two stages: first, a large number of random constituents are invited by mail; second, the members of the assembly are selected among those invited who volunteered to participate. Most algorithmic work on citizens' assemblies focuses on the second stage (Flanigan et al. 2020, 2021, 2024; Flanigan, Kehne, and Procaccia 2021; Baharav and Flanigan 2024).

This work, instead, studies a practical problem arising in the first sampling stage in certain countries. Sampling constituents with equal probability is straight-forward in countries with a central population register such as the Nordic countries (Scherpenzeel et al. 2017). In countries like the UK and US no register exists and assembly organizers use postal lists to invite random households, though these lists are incomplete and under-represent "rural areas, . . . , Hispanic households, non-English-speaking households" among others (Kalton, Kali, and Sigman 2014).

The first sampling stage is more complex in countries like Germany, where each of the 10,755 municipalities maintains its own population register. Since these municipalities must be individually petitioned for sampling access in a burdensome process (Stadtmüller et al. 2023), statistical surveys first sample a set of municipalities and then sample participants only from these municipalities' registers (Wasmer et al. 2017; INAPP 2022).

Our project was sparked by discussions with German sortition practitioners, who have been following a similar two-level sampling approach (Stabsstelle Bürgerräte 2023). Using numbers from the assembly on nutrition for illustration, they were looking for a sampling process that would (1) send out 20,000 invitation letters, (2) not send letters to more than 80 distinct municipalities at once, and (3) give each German resident an equal chance of being invited.[1]

The output of any sampling process, i.e., any probability distribution determining how many invitations to send to each municipality, can be represented in a graphical form, which we illustrate in Figure 1. To sample from this distribution, one draws a number $\rho \in [0, 1)$ uniformly at random, and considers the vertical line at this position (dashed in the figure). This line intersects the shapes in the diagram, each of which is labeled with a municipality, and the number of letters sent to a municipality is equal to the total height of

[1]In fact, assembly organizers break down the sampling into 42 sampling processes of this form, one for each federal state and category of municipality size. For exposition, we focus on an individual such problem, and consider the national level in Section 6.

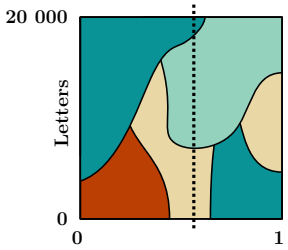


Figure 1: Graphical representation of sampling process.

the municipality's shapes at the vertical line.[2] In this representation, the practitioners' requirements are easy to express: (1) the total height of the figure at each vertical stripe should be 20,000 letters, (2) no vertical stripe should intersect with more than 80 shapes, and (3) the total area of a municipality's shapes (i.e., its expected number of received letters) must be proportional to its population.

A final requirement is that (4) the number of letters received by each municipality (or, the height of the municipality's shapes in any vertical strip) has an upper bound. Indeed, the municipality's population — which can be as low as 9 in the case of Germany — is clearly an upper bound, and many municipalities are moreover reluctant to allow sampling of more than about 10% of their population due to privacy concerns. In survey sampling, such upper constraints are not present because it is possible to upweight a resident in the analysis, effectively sampling them more than once. As a result, the solution used in survey sampling — sampling municipalities with *probability proportional to size* (Brewer and Hanif 1983), so that each vertical stripe consists of 80 equal-height layers — does not apply to assembly selection.

Whereas practitioners have so far relaxed conditions (1) and (2) (Stabsstelle Bürgerräte 2023) due to limitations in available methods, we show that all desiderata can, in fact, be satisfied and present several methods that achieve this.

Our Results and Techniques We begin by formulating our task as an optimization problem, MinFeasibleCities, which seeks a probability distribution satisfying the four conditions while minimizing the number of contacted municipalities. Although MinFeasibleCities is NP-hard, we provide a pseudo-polynomial time algorithm and an additive 1-approximation, both based on separation oracles for a linear programming formulation.

Since minimizing municipalities is only one of several practical goals, we introduce additional criteria. We first propose *ex-post monotonicity*, which states that, among the contacted municipalities, larger ones should receive at least as many letters as smaller ones. We present GreedyEqual, a natural algorithm that achieves ex-post monotonicity and an additive 2-approximation under mild assumptions.

Whereas GreedyEqual promotes balanced letter allocations, it is natural to strengthen monotonicity to *ex-post proportionality*, which states that a municipality's number of letters received scales with its size. We capture different proportionality goals through *target letter functions* and develop two algorithms to pursue them: an optimal method based on integer linear programming and a simpler heuristic.

Finally, we evaluate all algorithms on data from the source (Statistisches Bundesamt 2025) as was used for the German Citizens' Assembly on Nutrition (Stabsstelle Bürgerräte 2023). Since the selection is applied independently within 42 subgroups, we show how to lift the target letter notion from the local to the global level. Our algorithms offer practical solutions for diverse real-world requirements.

Related Work By contributing to the first stage of the assembly selection pipeline in practice, our work is complementary to, but technically independent from, algorithms for selecting the final assembly from those accepting the invitation. Flanigan et al. (2021) developed an optimization-based algorithm for this task; subsequent work studied transparent ways of drawing from the algorithm's computed probability distribution (Flanigan, Kehne, and Procaccia 2021), incentives for misrepresentation (Flanigan et al. 2024; Baharav and Flanigan 2024), accounting for self-selection bias (Flanigan et al. 2020), and the replacement of assembly members who drop out later (Assos et al. 2025).

Other works have studied sortition algorithms that directly draw the assembly from the population and resulting theoretical properties. These works study the variance of representation of features in the assembly (Benadè, Gölz, and Procaccia 2019), the social welfare if assembly members participate in a sequence of binary majority votes (Meir, Sandomirskiy, and Tennenholtz 2021), axioms and approximation bounds on the proximity of assembly members to the population in a metric space (Ebadian et al. 2022; Ebadian and Micha 2025; Caragiannis, Micha, and Peters 2024), and a proposed hierarchy of interconnected assemblies (Halpern et al. 2025). Do et al. (2021) study an online selection problem motivated by citizens' assemblies.

2 The Theoretical Model

We are given n cities[3] and a fixed number of letters $\ell \in \mathbb{N}$ to allocate. Each city has a population $\pi_i \in \mathbb{R}$ and we assume normalization wlog, i.e., $\sum_{i\in[n]} \pi_i = 1$. We also write $\vec{\pi} = (\pi_1, \dots, \pi_n)$. Every city has a maximum number of letters it can receive, denoted by $\vec{u} = (u_1, \dots, u_n) \in \mathbb{N}^n$. We assume $\pi_1 \le \dots \le \pi_n$, $u_1 \le \dots \le u_n \le \ell$. A letter allocation is a vector $a \in \mathbb{R}^n_{\ge 0}$ with the property that $\sum_{i\in[n]} a_i = \ell$ and $0 \le a_i \le u_i$ for all $i \in [n]$.[4] An allocation is *t-bounded* if at most t cities receive a non-zero number of letters; let A_t denote the set of all such allocations. Given an instance of our problem $(\vec{\pi}, \vec{u}, t)$, FeasibleCities describes the problem of deciding whether there exists a probability distribution $\mathcal{D}$ over A_t such that

$$\mathbb{E}_{a\sim\mathcal{D}}[a_i] = \pi_i \cdot \ell \text{ for all } i \in [n]. \tag{1}$$

We also refer to a probability distribution respecting Eq. (1) as *ex-ante fair*. MinFeasibleCities describes the corre-

[2]Clearly, the x-axis ordering of the diagram is arbitrary. All we need is that each color's union of shapes is measurable. Wlog, the selection within each municipality is uniform without replacement.

[3]For brevity, we use 'cities' as a synonym for 'municipalities'.

[4]For $k \in \mathbb{N}$ let $[k] = \{1, \dots, k\}$ and $[k]_0 = \{0, \dots, k\}$.

sponding optimization problem of finding the minimum t such that the answer to FEASIBLECITIES is yes.

Though letter allocations are integral in practice, i.e., $a \in \mathbb{N}^n$, this restriction is wlog for FEASIBLECITIES since any distribution over fractional allocations for t can be turned into a distribution over t-bounded integral allocations with the same ex-ante properties, through dependent randomized rounding (Gandhi et al. 2006). For convenience, we assume A_t to be integral in Section 3 and fractional in Section 4.

We assume $\pi_i \ell \leq u_i$ for all $i \in [n]$, which is a necessary condition for the existence of an ex-ante fair distribution (for any t) due to the upper bounds (see Lemma 1). Through the paper, we refer to the following running example:

Example 1. *Distribute $\ell = 60$ letters over $n = 8$ cities with sizes $\vec{\pi} = \frac{1}{360} \cdot (10, 10, 40, 40, 40, 50, 70, 100)$ and upper bounds $\vec{u} = 180 \cdot \vec{\pi} = (5, 5, 20, 20, 20, 25, 35, 50)$.*

While Section 3 studies city sampling through the lens of the optimization problem defined above, Sections 4 and 5 motivate and define additional desirable concepts: *ex-post monotonicity*, *ex-post proportionality*, and *binary outcomes*.

3 The MINFEASIBLECITIES Problem

In this section, we show that, though FEASIBLECITIES is NP-hard, it is only barely a hard problem, in the sense that pseudopolynomial time computation, or a slack of a single city suffice to overcome this complexity barrier. All missing proofs are provided in the appendix, available in the full version of our paper (Gölz et al. 2025).

We start by showing a simple lower bound that will be helpful throughout the paper. To this end, we define $w_i = \frac{\pi_i \ell}{u_i}$ for all $i \in [n]$, which yields a lower bound on the *selection probability* of a city (also interpreted as the minimum *width* within our illustrations).

Lemma 1. *For any instance $(\vec{\pi}, \vec{u}, t)$, and an ex-ante fair probability distribution $\mathcal{D}$ over A_t, it holds that*

(i) $\Pr[a_i > 0] \geq w_i$ *for all* $i \in [n]$, *and*

(ii) $t \geq \sum_{i \in [n]} w_i$.

For Example 1, Lemma 1 shows that t must be at least 3 since the minimum total width of all cities is $\sum_{i \in [n]} w_i = \frac{8}{3}$.

In the appendix, we show that FEASIBLECITIES is NP-hard via a reduction from PARTITION. In a nutshell, this reduction constructs an instance of our problem, in which all allocations in the support of a t-bounded, ex-ante fair distribution must give half of the cities 0 letters and half their upper bound u_i. The question whether any such allocation assigns exactly ℓ letters is exactly PARTITION.

Theorem 2. FEASIBLECITIES *is NP-hard.*

Since we reduce from PARTITION, which admits a pseudo-polynomial time algorithm, it is natural to ask whether our problem does too. To show that this is the case, we formulate the problem as a linear program with one variable x_a for each integral allocation $a \in A_t$. The LP searches for a fair distribution over these, with x_a representing the probability assigned to allocation a. The first constraint ensures that the probabilities sum to at most 1; the second enforces fairness. Both hold with equality in any feasible solution, but are written as inequalities for clarity in the dual.

$$
\begin{aligned}
\textbf{Primal:}\quad & \text{minimize } 0 \\
& \text{subject to } \sum_{a \in A_t} x_a \leq 1, \\
& \qquad \sum_{a \in A_t} x_a a_i \geq \pi_i \ell \qquad \text{for } i \in [n], \\
& \qquad x_a \geq 0 \qquad\qquad \text{for } a \in A_t. \\
\textbf{Dual:}\quad & \text{maximize} \sum_{i \in [n]} \pi_i \ell y_i - y \\
& \text{subject to } \sum_{i \in [n]} a_i y_i \leq y \qquad \text{for } a \in A_t, \qquad (2) \\
& \qquad y, y_i \geq 0 \qquad\qquad \text{for } i \in [n].
\end{aligned}
$$

We aim to decide whether the primal LP is feasible, which is the case iff the dual LP admits no solution with positive objective value (which could be scaled to show that the dual value is unbounded). We add a constraint to the dual requiring a strictly positive objective value.

Though the resulting system has exponentially many constraints, its feasibility can be decided with the ellipsoid method (Grötschel, Lovász, and Schrijver 1993) provided we can implement a *separation oracle* for the dual: given a vector $\big((y_i)_{i \in [n]}, y\big)$, we must decide whether it is feasible for the modified dual or return a violated constraint. We show that this separation problem can be solved in pseudo-polynomial time using a knapsack-style dynamic program.

Theorem 3. *There exists a pseudo-polynomial time algorithm for* FEASIBLECITIES.

More surprisingly, we can construct a polynomial-time *approximate separation oracle*, in the following, strong sense: given a vector $\big((y_i)_{i \in [n]}, y\big)$, our oracle either determines that the vector satisfies all dual constraints or identifies a violated constraint of type (2), but for some allocation $a \in A_{t+1} \supseteq A_t$ rather than in A_t. As Schulz and Uhan (2013) show, the ellipsoid method with such an approximate oracle can determine either that the dual above is unbounded (so the primal is infeasible) or that the dual for $t + 1$ cities is bounded (hence, the primal for $t + 1$ cities is feasible). By applying this algorithm to increasing values of t until a feasible primal is found, we can find the lowest possible number of contacted cities, up to perhaps one additional city.

Theorem 4. *There exists a polynomial-time algorithm that is a additive* 1*-approximation to* MINFEASIBLECITIES.

While the above algorithms are theoretically tractable, the ellipsoid method is a famously impractical algorithm.[5] Moreover, these algorithm may yield highly unintuitive allocations that would be difficult to justify in practice. For example, a large city might receive significantly fewer letters than a smaller one ex post, or only small cities might be selected while all larger ones are excluded. We now shift our focus from mere feasibility to fair distributions that uphold additional desirable properties, all while keeping t low.

[5] Although lacking theoretical guarantees, combining our (or similar) separation oracles with the simplex method can still lead to practical algorithms (see COLUMNGENERATION in Section 5).

4 A Simple and Monotone Approximation

In this section, we aim for *ex-post monotonicity*. A fractional allocation a is called *monotone* if $a_i \geq a_j$ whenever $i > j$. A probability distribution is ex-post monotone if its support consists of only monotone allocations.[6] We present a simple additive 2-approximation for MINFEASIBLECITIES that yields ex-post monotone distributions under mild assumptions. The algorithm is inspired by πps sampling (Brewer and Hanif 1983): given an instance $(\vec{\pi}, \vec{u}, t)$, one samples t cities with probabilities proportional to $\vec{\pi}$ without replacement and assigns ℓ/t letters to each. While this would violate cities' upper bounds, our algorithm can be viewed as a minimal adjustment to πps sampling to ensure feasibility.

Our algorithm, GREEDYEQUAL, is best understood through its geometric interpretation. GREEDYEQUAL processes cities in increasing order of size and starts by attempting to place a $\pi_i \ell$-area rectangle of height ℓ/t. If this violates the city's upper bound, it instead places a rectangle of height u_i. It then proceeds to place the next rectangle to the right. Once the first layer is filled, GREEDYEQUAL moves to the next layer, now aiming to keep the height of rectangles at the remaining vertical space divided by $t-1$. This ensures that later (and thus larger) cities can receive at least as many letters as those already placed. We remark that, starting from the second layer, cities may receive a set of rectangles summing to $\pi_i \ell$ instead of a single rectangle, which is due to shifts in lower layers. See Figure 2a for an illustration.

To formalize GREEDYEQUAL, we introduce a second type of illustration, which is a flattened version of the illustration in Figure 2a. This illustration is formalized by functions λ_i for each $i \in [n]$ that are defined on the interval $[0, t)$. The value $\lambda_i(x)$ corresponds to the height of the rectangle that the algorithm draws for city i in layer $\lfloor x \rfloor$ (0-indexed) and at position $x - \lfloor x \rfloor$ of the stacked picture. (Note that for any position $x \in [0, t)$ this value will be non-zero for exactly one city as the algorithm draws for one city at a time.) We illustrate these functions in Figure 2b.

When the algorithm draws at position x in the flat picture, it needs to know the height of all rectangles that were placed at some value $y \leq x - 1$ with $y \equiv x \pmod 1$. We define

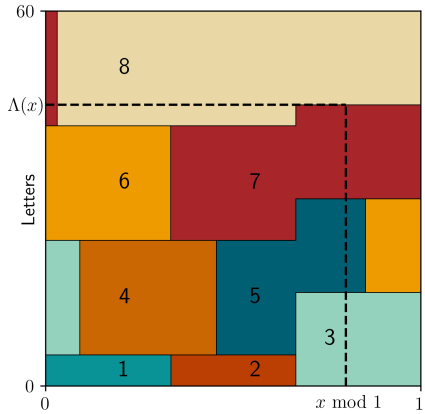
$$\Lambda(x) = \sum_{y \leq x,\ y \equiv x \pmod 1} \sum_{j \leq i} \lambda_j(y).$$

Last, we define $\mu_i(x)$, describing the height of the rectangle to be drawn, given that we place city i at position x,

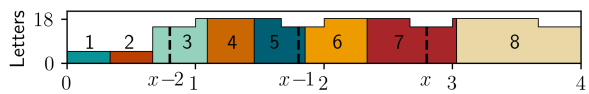
$$\mu_i(x) = \begin{cases} \min\left(u_i, \frac{\ell - \Lambda(x-1)}{t - \lfloor x \rfloor}\right) & \text{for } x \in [0, t) \\ u_i & \text{for } x \geq t, \end{cases}$$

and are now ready to formalize GREEDYEQUAL:

procedure GREEDYEQUAL($\vec{\pi}, \vec{u}, t$)
  $x \leftarrow 0, i \leftarrow 1$
  while $i \leq n$ **do**
    let $y \geq x$ such that $\int_x^y \mu_i(z)dz = \pi_i \ell$
    $\lambda_i(z) \leftarrow \mu_i(z)$ for $z \in [x, y)$, $x \leftarrow y$, $i \leftarrow i + 1$
  if $x = t$ **then return** $(\lambda_i)_{i \in [n]}$ **else** "fail"



(a) Illustration GREEDYEQUAL for Example 1 for $t = 4$.



(b) Illustration of the functions λ_i for all $i \in [n]$, where each λ_i is indicated by a different color.

Figure 2: GREEDYEQUAL applied to Example 1.

GREEDYEQUAL can fail in two ways. First, it may terminate with $x > t$, meaning it requires more than t layers and thus does not yield a t-bounded distribution. In Theorem 8, we bound the optimum of MINFEASIBLECITIES in this case. Second, and more subtly, the area assigned to a city may be so wide that it overlaps across layers, leading to an allocation that exceeds the upper bound of the city. This can only happen when a city is *oversized*, i.e., when $\pi_i > 1/t$. While such cities appear in parts of our data, they always have upper bounds well above ℓ, making this a non-issue in practice. We formalize the following assumption:

Assumption 1. *For any oversized city i, $u_i \geq \ell$.*

In our dataset, Assumption 1 is satisfied as long as $t \leq 420$, far above the past choice of $t = 80$.

Theorem 5. *Under Assumption 1,* GREEDYEQUAL *always returns an ex-ante fair and t-bounded probability distribution (if it succeeds).*

In instances without oversized cities we furthermore guarantee monotonicity. In our data we do not observe any monotonicity violation, even for oversized cities.

Theorem 6. *For instances without oversized cities,* GREEDYEQUAL *is ex-post monotone.*

Before proving the additive 2-approximation, we provide insight into the structure of GREEDYEQUAL's solutions. We say that GREEDYEQUAL *selects the average* at position $x \in$

[6] For ex-post monotonicity, the relaxation to fractional allocations is not quite wlog, but any fractional monotone allocation can be decomposed into a distribution over integral allocations that are monotone up to one letter.

$[0,t)$ if $\lambda_i(x) = \frac{\ell - \Lambda(x-1)}{t - \lfloor x \rfloor}$ (rather than $\lambda_i(x) = u_i$), where i is the unique city with $\lambda_i(x) > 0$.

Lemma 7. *Independent of whether* GREEDYEQUAL *succeeds, the following holds:*

(i) If GREEDYEQUAL *selects the average at $x \in [0,t)$, then it selects the average at all $y \in [x, \lceil x \rceil)$ as well as all $y \in [x,t)$ with $y \equiv x \pmod 1$.*

(ii) The function $\Lambda(x)$ is non-decreasing on $[0,t)$.

Theorem 8. *Under Assumption 1,* GREEDYEQUAL *is an additive 2-approximation for* MINFEASIBLECITIES.

Proof (first part). Let $(\vec{\pi}, \vec{u}, t)$ be an instance of our problem such that GREEDYEQUAL fails. We show that $\sum_{i \in [n]} w_i > t-2$, which by Lemma 1 (ii), implies that the optimal budget for MINFEASIBLECITIES is at least $t-1$.

To gain intuition for the proof, consider the following thought experiment: imagine scaling each city's shapes so that it maintains its total area but reaches its maximum height u_i, attaining its minimum width w_i. How much width do we lose in total? The original sum of widths exceeds t; we show that even after scaling, the total width remains strictly greater than $t-2$. While it may seem natural to scale each city individually, our analysis instead partitions the stacked picture into "columns" and scales each column separately.

Let $\mathcal{I}$ be a partition of the interval $[0,1)$ with the property that all functions λ_i are constant along each interval $I \in \mathcal{I}$. Now, consider the interval in $\mathcal{I}$ that starts at 0. For all $k \in [t-1]_0$ let $j(k)$ be the unique city with $\lambda_{j(k)}(k) > 0$. From now, we drop the position and write $\lambda_{j(k)}$ instead of $\lambda_{j(k)}(k)$. Since GREEDYEQUAL failed, we know that $\Lambda(x) < \ell$ for some $x \in [t-1,t)$. Moreover, by the monotonicity of Λ (Lemma 7 (ii)) we know that $\Lambda(t-1) < \ell$. By Lemma 7 (i) it holds that $\lambda_{j(k)} = u_{j(k)}$ for all $k \in [t-1]_0$.

Now consider an arbitrary interval $[\alpha, \beta) \in \mathcal{I}$. For each $k \in [t-1]_0$, let $i(k)$ be the unique city with $\lambda_{i(k)}(k+\alpha) > 0$. We write $\lambda_{i(k)}$ for $\lambda_{i(k)}(k+\alpha)$. See Figure 3 for an illustration. The original total width in column $[\alpha, \beta)$ is $(\beta - \alpha)t$. We show that scaling each subarea to its maximum height yields a total width greater than $(\beta - \alpha)(t-2)$. Since the factor $(\beta - \alpha)$ is irrelevant to our argument, we drop it.

Claim. *It holds that $\sum_{k=0}^{t-2} \frac{\lambda_{i(k)}}{u_{i(k)}} > t-2$.*

Proof of the claim. Since GREEDYEQUAL processes cities with increasing indices, it holds that $i(k) \le j(k+1)$ for all $k \in [t-2]_0$. Thus:

$$\lambda_{i(k)} \le u_{i(k)} \le u_{j(k+1)} = \lambda_{j(k+1)}. \tag{3}$$

Let t' be the first index for which $\lambda_{i(t')} = \frac{\ell - \Lambda(t'-1+\alpha)}{t-t'}$. If no such index exists, then we know that $\lambda_{i(k)} = u_{i(k)}$ for all $k \in [t-2]_0$ and the claim follows trivially. In the example in Figure 3 it holds that $t' = 3$. We define $\ell_i = \sum_{k=t'}^{t-1} \lambda_{i(k)}$ and $\ell_j = \sum_{k=t'+1}^{t-1} u_{j(k)}$. Note that $\ell_i > \ell_j$, since $\Lambda(t'-1+\alpha) < \Lambda(t')$. By Lemma 7 (i):

$$\lambda_{i(k)} = \frac{\ell_i}{t-t'} \text{ for all } k \in \{t', \dots, t-1\} \tag{4}$$

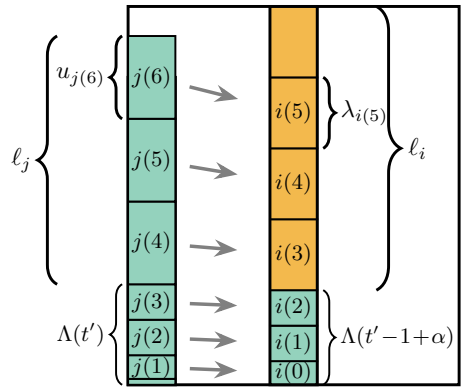


Figure 3: Situation in the proof of Theorem 8. In teal areas, cities receive their upper bounds u_i and in orange areas they receive less than u_i. An arc indicates $u_{j(k)} \ge u_{i(k-1)}$.

We are now ready to prove the claim

$$\begin{aligned}
\sum_{k=0}^{t-2} \frac{\lambda_{i(k)}}{u_{i(k)}} &= t' + \sum_{k=t'}^{t-2} \frac{\lambda_{i(k)}}{u_{i(k)}} \\
&\overset{(4)}{=} t' + \frac{\ell_i}{(t-t')} \sum_{k=t'}^{t-2} \frac{1}{u_{i(k)}} \\
&\overset{(\star)}{\ge} t' + \frac{\ell_i}{(t-t')} \frac{(t-t'-1)^2}{\sum_{k=t'}^{t-2} u_{i(k)}} \\
&\overset{(3)}{\ge} t' + \frac{\ell_i}{(t-t')} \frac{(t-t'-1)^2}{\sum_{k=t'+1}^{t-1} u_{j(k)}} \\
&> t' + \frac{\ell_j}{t-t'} \frac{(t-t'-1)^2}{\ell_j} \\
&= t' + t - t' - 2 + \frac{1}{t-t'} > t-2,
\end{aligned}$$

where $(\star)$ follows from the fact that the arithmetic mean is at least the harmonic mean (applied to the values $\frac{1}{u_i}$). ■

It remains to apply the above claim to all columns and conclude $\sum_{i \in [n]} w_i > t-2$. We refer to the appendix. □

We also show that our upper bound for the approximation guarantee of GREEDYEQUAL is tight:

Theorem 9. *Even under Assumption 1,* GREEDYEQUAL *is not an additive 1-approximation for* MINFEASIBLECITIES.

5 Ex-post Proportionality

Ex-post monotonicity ensures that after randomization, larger selected cities receive at least as many letters as smaller ones. However, it does not guarantee that larger cities receive *more* letters. For example, a city with millions of inhabitants might still receive the same number of letters as one with only tens of thousands — behavior that GREEDYEQUAL in fact encourages. We explore how both the selection probability and the number of letters a city receives (if selected) can grow with the population. To achieve this, we introduce the more general concept of *target letters*.

We assign each city i a number τ_i of target letters, that it should receive if it is selected. This, in term also implies a target selection probability $\frac{\pi_i \ell}{\tau_i}$ for that city. If we let the target letters grow proportionally to the population, then each city gets selected with the same probability. If, on the other hand, the target is equal for all cities, then the target selection probability grows proportionally to the population, which is close to what GREEDYEQUAL achieves. It seems natural to allow for target functions in between those two extremes.

We define a *target letter function* to be a monotone function f taking as input a population size π_i and outputting a target in $\mathbb{R}_{\geq 0}$. However, blindly setting targets without knowing the budget t can lead to infeasibility: For example, small targets will clearly be missed if t is very small. To mitigate this issue, we introduce a scaling factor κ and define for each city i the scaled target letters as

$$\tau_i^\kappa = \max\left(\pi_i \ell, \min\left(u_i, \kappa f\left(\pi_i\right)\right)\right), \tag{5}$$

which makes sure that target letters do not exceed u_i and the target selection probability does not exceed 1. We then determine the value κ such that the total target selection probability (or width) satisfies $\sum_{i\in[n]} \frac{\pi_i \ell}{\tau_i^\kappa} = t$ and set $\tau_i = \tau_i^\kappa$ for all $i \in [n]$. A total width of at most t is a necessary condition for the targets to be achievable but is far from sufficient due to the more complex structure of the problem.

We suggest $f(x) = \sqrt{x}$ as a particularly natural target letter function since it allows target letters and target selection probability to scale in equal measure. We introduce two methods that take as input an instance and the target letters and aim to construct a fair distribution meeting the targets. As in Section 4, we allow for distributions over fractional allocations for the sake of simplicity, which immediately approximates integral ex-post proportionality up to one letter.

Column generation Recall our linear programming approach from Section 3, which we used for deciding whether an instance is feasible or not. It is natural to add an objective function to this LP to minimize, in expectation, a measure of deviation from the targets. Specifically, we minimize $\sum_{a\in A_t} x_a \varphi(a)$, where φ measures the total relative deviation from the targets, i.e.,

$$\varphi(a) = \sum_{i\in[n], a_i>0} \frac{|\tau_i - a_i|}{\tau_i}.$$

This objective penalizes the same absolute deviation from the target more heavily for smaller cities than for larger ones. To optimize the resulting primal LP, we again design a separation oracle for the dual LP (a process also termed *column generation*). This time, the separation problem is more complex, and we formulate a mixed integer linear program to solve it (see appendix). Though not polynomial-time, state-of-the-art solvers scale to large problems in practice.

While COLUMNGENERATION is optimal with respect to the target letters, the resulting distributions have little visual structure (e.g., see Figure 4c), and the algorithm's reliance on optimization solvers makes them hard to explain to the public. We introduce an alternative approach that is arguably more transparent, while still aiming to meet the target letters.

Bucket Approach The idea of BUCKETS is to partition the cities into t disjoint sets (the *buckets*), such that we can then sample exactly one city from each bucket. Each bucket has a *height*, which determines how many letters the selected city from that bucket receives. Within each bucket, we thus need to sample proportional to size. By ex-ante fairness, the height of a bucket $B \subseteq [n]$ is determined by its elements $h = \sum_{i\in B} \pi_i \ell$. To approximate the target letters $\vec{\tau}$, we define the buckets such that the target letters of each city are close to the height of the bucket it belongs to.

To achieve this, we fill the buckets iteratively with cities in increasing order of their size. We move on to the next bucket if adding another city would either (i) increase the total target probability of all cities in the bucket above one, or (ii) would increase the height of the bucket above the maximum number of letters of its smallest city. See appendix.

The bucket approach has the advantage of producing easily explainable distributions (see, e.g., Figure 4b). In particular, it satisfies the *binary outcome* property: each city knows in advance how many letters it will receive if selected. While the method does not guarantee ex-post monotonicity in the worst case, we observe no violations in our data. Moreover, it ensures that selected cities are distributed somewhat evenly across cities of different sizes. On the downside, the approach lacks worst-case approximation guarantees.

Theorem 10. *For any targets and constant c,* BUCKETS *is not an additive c-approximation for* MINFEASIBLECITIES.

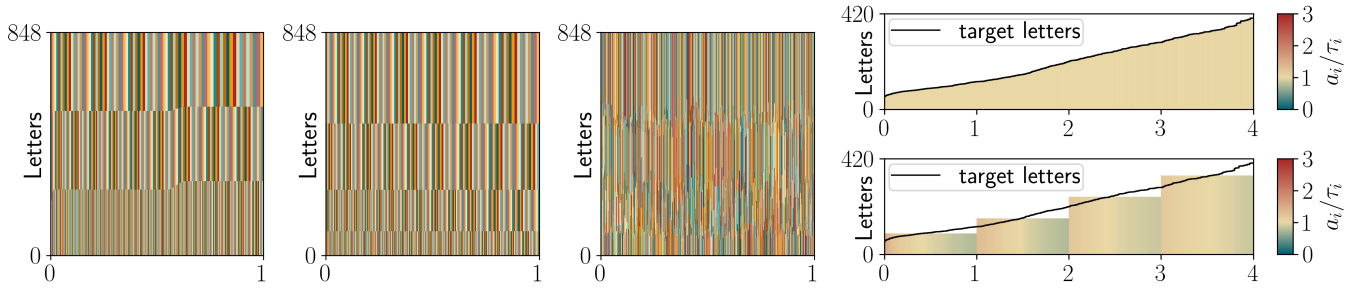
6 Towards Practice

We aim to apply one of our algorithms in future implementations of citizens' assemblies, particularly in Germany. To this end, we tested them on data from the same source (Statistisches Bundesamt 2025) as was used for the assembly on nutrition (Stabsstelle Bürgerräte 2023), where $\ell = 20\,000$ letters were sent, the outreach budget was $t = 80$, and there are $n = 10\,755$ cities with a total population of $84\,\text{M}$. Following suggestions from practitioners, the maximum number of letters a city can receive is given as 50% of the population for cities under 500 inhabitants, 10% for those over $2\,500$, and 250 for populations in between.

In this recent assembly, practitioners divided the country into 42 groups, based on the 16 federal states and on three city size classes ($[0, 20\text{K})$, $[20\text{K}, 100\text{K})$, $[100\text{K}, \infty)$),[7] and sampled the letter allocation per group. This stratification ensures sufficient numbers of invitations within each group for forming the assembly in the second stage of selection. Since the same grouping will likely be used for future assemblies, we test our algorithms in this setup.

Apportionment via Global Targets While a group's number of letters is just proportional to its population, we must decide how to allocate the outreach budget $t = 80$ across groups. Let $\mathcal{G}$ be the partition of $[n]$ into groups. Blindly apportioning the outreach budget t into group budgets t_G for each $G \in \mathcal{G}$ and then applying our algorithms is not ideal for meeting letter targets: Similarly sized cities

[7] Some states consist of only a single or two large cities.



(a) GREEDYEQUAL ($t_G = 3$)

(b) BUCKETS ($t_G = 4$)

(c) COLUMNGENERATION ($t_G = 4$)

(d) Proportionality of COLUMNGENERATION (top) and BUCKETS (bottom)

Figure 4: Probability distribution for the group of small cities in the state of Niedersachsen.

in different groups may receive vastly different numbers of letters when selected, as this number depends on t_G.

We introduce the concept of *global targets*, which help finding an apportionment that keeps letter targets comparable across groups. Given a target letter function (for COLUMNGENERATION and BUCKETS we use $f(x) = \sqrt{x}$ and for GREEDYEQUAL we use a constant function), we compute the global target letters τ_i by finding a scaling factor κ such that the corresponding target widths $\omega_i = \frac{\pi_i \ell}{\tau_i^\kappa}$ sum up to $t = 80$ (compare Section 5).

However, as we have argued in Section 5, within each group G, we need to rescale $\sum_{i \in G} \omega_i$ to a width of t_G to obtain sensible *local targets*. To keep the amount of rescaling required low (and, in turn, local targets close to global targets), we want to assign each group an integer budget t_G close to their fractional target width $\sum_{i \in G} \omega_i$. This is an apportionment problem, for which we use an adjustment of Adam's apportionment method (Balinski and Young 2001). For details, see appendix.

Meeting Local Targets After finding the apportionment, we test our three algorithms (GREEDYEQUAL, COLUMNGENERATION, and BUCKETS) on these 42 groups. All algorithms find distributions for the apportioned t_G, and run in a practical amount of time on consumer hardware. This shows that our algorithms scale to practical problems and are plausible contenders for deployment. We defer results and detailed discussions to the appendix and display the distributions for one group in Figures 4a to 4c.

Figure 4d visualizes how well COLUMNGENERATION and BUCKETS meet their local targets. The figure is the result of ordering all rectangles from Figures 4b and 4c by the city they represent and lining them all up next to each other in increasing order of city sizes. Each rectangle's color represents how close its height is to the target letters of that city. We plot the local targets of cities in black.

In this instance, COLUMNGENERATION meets the target letters almost perfectly, which is true for most of the groups (more precisely, 35 out of 42 and in particular for all but one groups with $t_G \geq 3$). BUCKETS approximates the target letters, with the smaller cities within each bucket receiving slightly too many, and the larger ones slightly too few letters. BUCKETS struggles when there are very small cities, since the smallest city in a bucket bounds its height and limits the number of letters to the other cities in the bucket. This effect appears in 5 out of the 42 groups. Both approaches align more closely with local targets for higher values of t_G.

Meeting Global Targets We observe that the local targets of groups with $t_G > 1$ never deviate from the global targets by more than a factor of 1.5. For groups with $t_G = 1$ the local targets are independent of the target function, as every city must receive all letters of this group when selected, which can lead to arbitrarily high deviations from the global targets. In particular, many of the medium- and large-size groups have a low total share of the population, which leads to a target width significantly below 1. Since each group must have $t_G \geq 1$, these groups are assigned $t_G = 1$, resulting in local targets letters that can be much lower than the global targets. We present a visualization of global and local targets in the appendix.

Germany holds assemblies nationally and at the state level. In the appendix, we also show results for Baden-Württemberg, a particularly active state.

7 Discussion

We introduced a novel two-stage sampling problem, motivated by the practical demands of selecting citizens' assemblies. Our results offer a solid algorithmic foundation and give rise to two compelling open questions: Does there exist an ex-post monotone additive 1-approximation algorithm? And can the representation of city groups — currently addressed via partitioning — be integrated more directly into the model? GREEDYEQUAL and BUCKETS already ensure the ex-post representation of cities of different sizes by design, and one might envision a two-dimensional sampling framework, as is often used in survey sampling (Cox 1987).

While these questions offer exciting directions for theory, our focus remains on practical impact. As Germany's newly elected government just reaffirmed its commitment to citizens' assemblies (CDU, CSU, and SPD 2025), our work offers a suite of implemented algorithms, striking distinct, favorable tradeoffs between different practical desiderata. Based on our discussions with practitioners, we are optimistic that they will soon be used to sample real assemblies.

Acknowledgements

We would like to thank Federico Fioravanti for interesting discussions during an early stage of the project, Jannik Matuschke for helpful discussions about configuration LPs and approximate separation oracles, and Bettina Speckmann for interesting discussions. Moreover, we thank Brett Hennig for mentioning the practical problem during an online talk organized by the European Digital DemocracY (EDDY) network in 2024.

Part of this work was performed while Paul Gölz was at the Simons Institute for the Theory of Computing as a FODSI research fellow, for which he acknowledges the NSF's support through grant DMS-2023505. Jan Maly was supported by the Austrian Science Fund (FWF) under the grants 10.55776/PAT7221724 and 10.55776/COE12, by netidee Förderungen (https://www.netidee.at/) and the Vienna Science and Technology Fund (WWTF) (Grant ID: 10.47379/ICT23025). Ulrike Schmidt-Kraepelin was supported by the Dutch Research Council (NWO) under project number VI.Veni.232.254.

References

Assos, A.; Baharav, C.; Flanigan, B.; and Procaccia, A. 2025. Alternates, Assemble! Selecting Optimal Alternates for Citizens' Assemblies. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 719–738.

Baharav, C.; and Flanigan, B. 2024. Fair, Manipulation-Robust, and Transparent Sortition. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 756–775.

Balinski, M.; and Young, H. P. 2001. *Fair Representation: Meeting the Ideal of One Man, One Vote*. Brookings Institution Press, 2 edition.

Benadè, G.; Gölz, P.; and Procaccia, A. D. 2019. No Stratification Without Representation. In *Proceedings of the ACM Conference on Economics and Computation (EC)*, 281–314.

Bertelsmann. 2024. Forum against Fakes: Citizens' Report on How to Deal with Disinformation. Together for a Strong Democracy. Technical report, Bertelsmann Stiftung, Gütersloh.

Brewer, K. R. W.; and Hanif, M. 1983. *An Introduction to Sampling with Unequal Probabilities*, volume 15. New York, NY: Springer.

Caragiannis, I.; Micha, E.; and Peters, J. 2024. Can a few decide for many? the metric distortion of sortition. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 5660–5679.

CDU; CSU; and SPD. 2025. Verantwortung Für Deutschland. Koalitionsvertrag Zwischen CDU, CSU Und SPD (21. Legislaturperiode). Technical report.

Courant, D. 2021. Citizens' Assemblies for Referendums and Constitutional Reforms: Is There an "Irish Model" for Deliberative Democracy? *Frontiers in Political Science*, 2.

Cox, L. H. 1987. A Constructive Procedure for Unbiased Controlled Rounding. *Journal of the American Statistical Association*, 82(398): 520–524.

Deutscher Bundestag. 2024. Bürgergutachten – Empfehlungen des Bürgerrates "Ernährung im Wandel: Zwischen Privatangelegenheit und staatlichen Aufgaben" an den deutschen Bundestag. Technical Report 20/10300, Deutscher Bundestag.

Do, V.; Atif, J.; Lang, J.; and Usunier, N. 2021. Online Selection of Diverse Committees. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 154–160.

Ebadian, S.; Kehne, G.; Micha, E.; Procaccia, A. D.; and Shah, N. 2022. Is Sortition Both Representative and Fair? In *Proceedings of the 35th Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 3431–3443.

Ebadian, S.; and Micha, E. 2025. Boosting Sortition via Proportional Representation. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 667–675.

Flanigan, B.; Gölz, P.; Gupta, A.; Hennig, B.; and Procaccia, A. D. 2021. Fair Algorithms for Selecting Citizens' Assemblies. *Nature*, 596(7873): 548–552.

Flanigan, B.; Gölz, P.; Gupta, A.; and Procaccia, A. D. 2020. Neutralizing Self-Selection Bias in Sampling for Sortition. In *Proceedings of the 33th Conference on Advances in Neural Information Processing Systems (NeurIPS)*, volume 33.

Flanigan, B.; Kehne, G.; and Procaccia, A. D. 2021. Fair Sortition Made Transparent. In *Proceedings of the 34th Conference on on Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 25720–25731.

Flanigan, B.; Liang, J.; Procaccia, A. D.; and Wang, S. 2024. Manipulation-Robust Selection of Citizens' Assemblies. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(9): 9696–9703.

Gandhi, R.; Khuller, S.; Parthasarathy, S.; and Srinivasan, A. 2006. Dependent Rounding and Its Applications to Approximation Algorithms. *Journal of the ACM (JACM)*, 53(3): 324–360.

Gölz, P.; Maly, J.; Schmidt-Kraepelin, U.; Utke, M.; and Verpoort, P. C. 2025. City Sampling for Citizens' Assemblies. arXiv:2509.07557.

Grötschel, M.; Lovász, L.; and Schrijver, A. 1993. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer.

Halpern, D.; Procaccia, A. D.; Shapiro, E.; and Talmon, N. 2025. Federated Assemblies. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 13897–13904.

INAPP. 2022. Sample Design Summary: ESS Round 11. Technical report.

Kalton, G.; Kali, J.; and Sigman, R. 2014. Handling Frame Problems When Address-Based Sampling Is Used for In-Person Household Surveys. *Journal of Survey Statistics and Methodology*, 2(3): 283–304.

Mehr Demokratie. 2021. Germany's Role in the World. The Recommendations of the Digital Citizens' Assembly. Technical report, Mehr Demokratie e.V., Berlin.

Meir, R.; Sandomirskiy, F.; and Tennenholtz, M. 2021. Representative Committees of Peers. *Journal of Artificial Intelligence Research*, 71: 401–429.

OECD. 2020. *Innovative Citizen Participation and New Democratic Institutions: Catching the Deliberative Wave*. Organisation for Economic Co-operation and Development.

Scherpenzeel, A.; Maineri, A.; Bristle, J.; Pflüger, S.-M.; Mindorova, I.; Butt, S.; Zins, S.; Emery, T.; and Luijkx, R. 2017. Report on the Use of Sampling Frames in European Studies: SERISS Deliverable. Technical report, SERISS - Synergies for Europe's Research Infrastructure in the Social Sciences.

Schulz, A. S.; and Uhan, N. A. 2013. Approximating the Least Core Value and Least Core of Cooperative Games with Supermodular Costs. *Discrete Optimization*, 10(2): 163–180.

Stabsstelle Bürgerräte. 2023. Bürgerrat Ernährung. So funktioniert die Auslosung – Zufallsauswahl im Detail erklärt. Bürgerrat Ernährung, Deutscher Bundestag.

Stadtmüller, S.; Silber, H.; Gummer, T.; Sand, M.; Zins, S.; Beuthner, C.; and Christmann, P. 2023. Evaluating an Alternative Frame for Address-Based Sampling in Germany: The Address Database From Deutsche Post Direkt. *methods, data, analyses*, 17: 17 Pages.

Statistisches Bundesamt. 2025. Auszug GV1Q Aktuell – Gemeindeverzeichnis / Administrativ. https://www.destatis.de/DE/Themen/Laender-Regionen/Regionales/Gemeindeverzeichnis/Administrativ/Archiv/GVAuszugQ/AuszugGV1QAktuell.html. Accessed: 2025-07-01.

Wasmer, M.; Blohm, M.; Walter, J.; Jutz, R.; and Scholz, E. 2017. Konzeption und Durchführung der "Allgemeinen Bevölkerungsumfrage der Sozialwissenschaften" (ALLBUS) 2014. *GESIS Papers*, 2017/20: 74 S.