Federated Assemblies

Daniel Halpern [1], Ariel D. Procaccia [1], Ehud Shapiro [2,3] Nimrod Talmon [4]

[1]Harvard University
[2]Weizmann Institute of Science
[3]London School of Economics
[4]Ben-Gurion University
dhalpern@g.harvard.edu, arielpro@seas.harvard.edu, ehud.shapiro@weizmann.ac.il, talmonn@bgu.ac.il

Abstract

A *citizens' assembly* is a group of people who are randomly selected to represent a larger population in a deliberation. While this approach has successfully strengthened democracy, it has certain limitations that suggest the need for assemblies to form and associate more organically. In response, we propose *federated assemblies*, where assemblies are interconnected, and each parent assembly is selected from members of its child assemblies. The main technical challenge is to develop random selection algorithms that meet new representation constraints inherent in this hierarchical structure. We design and analyze several algorithms that provide different representation guarantees under various assumptions on the structure of the underlying graph.

1 Introduction

Citizens' assemblies are a popular mechanism for democratic decision making (Stone 2011; Van Reybrouck 2016; Fishkin 2018; Landemore 2020). In the last decade, this paradigm has vastly grown in recognition and influence. In Europe, for example, governments have sponsored citizens' assemblies to inform national policy on constitutional questions (Ireland), climate change (France), and even nutrition (Germany). Technology companies like Meta (Clegg 2023) are also piloting (enormous) citizens' assemblies as a way of obtaining democratic inputs for AI governance and alignment.

While different assemblies may take somewhat different approaches, they all share two distinctive features. First, members of a citizen's assembly are *randomly selected* among volunteers. Second, members of the assembly engage in a long and substantial *deliberation* before reaching any conclusions.

The former feature is of great technical interest, as it is challenging to design a good random selection process. The goal is to achieve *descriptive representation*, in the sense that the assembly should reflect the composition of the population along multiple dimensions like gender, age, ethnicity and level of education; this is seen as a source of legitimacy for citizens' assemblies. However, since the pool of volunteers is typically skewed due to self-selection bias, uniform random selection will not yield descriptive representation and more sophisticated algorithmic solutions are required. Such algorithms, which are designed to achieve descriptive representation while optimizing fairness to volunteers, have been broadly deployed (Procaccia 2022).

Our proposal: Federated assemblies. To our knowledge, the hundreds of citizens' assemblies convened around the world have all been independent of each other: in collaboration with practitioners, different countries, regions and municipalities have organized their own assemblies from the ground up.

By contrast, we propose a novel form of citizens' assemblies: *federated assemblies*. The most basic building block of a federated assembly is two assemblies (say, each representing the residents of a city) that decide to federate, forming a new parent assembly (which represents the residents of both cities and discusses policy questions of mutual interest); crucially, *the members of the parent assembly are selected from the child assemblies*. More generally, a parent assembly can have more than two children, a child assembly can have more than one parent, and the parent assemblies themselves can federate. Overall, a federated assembly is represented by a directed acyclic graph, where nodes correspond to assemblies and an edge from x to y means that x federated with other assemblies to form the parent assembly y. The lowest level (non-federated) assemblies are *leaf assemblies*, which allow people to directly sign up as constituents. Even leaf assemblies need not represent only geographical entities, they can also correspond to issues (such as climate change) or identity groups (such as ethnic groups). Furthermore, people can sign up for multiple leaf assemblies — any that intersect their multi-faceted interests.

An important question, of course, is who would establish these assemblies, manage the infrastructure, and provide funding. One possible solution is to use the conceptual framework of grassroots systems (Shapiro 2023a,b). A federated assemblies grassroots platform (Shapiro 2024) would operate in a decentralized manner on individuals' smartphones without relying on any global resources other than the network itself. This would allow such systems to develop organically, without centralized authorities and external dependencies.

In our view, federated assemblies have several advantages over the current practice of citizens' assemblies. First, the process of forming a new assembly does not have to start from the very beginning, as its members are selected from child assemblies; therefore, the lengthy and costly process of recruiting volunteers can be avoided altogether and the bar for forming an assembly is significantly lowered. Second, in standard citizens' assemblies, the determination of which features to stratify over, and which values to assign to these features — which is made by the organizers — is sometimes controversial and gives rise to manipulation opportunities. By contrast, in federated assemblies, these "features" — which are induced by the structure of the graph — are self-determined. Third, in the spirit of *associative democracy* — "a model of democracy where power is highly decentralized and responsibility for civic well-being resides with like-minded civic associations" (Jones and Marsden 2010) — federated assemblies allow citizens to exercise power by forming organizations that are immediately integrated into a broader framework of governance. We believe that federated assemblies may be especially pertinent in the context of a *global citizens' assembly* — the holy grail of practitioners of deliberative democracy — as such an assembly could form organically as a federation of assemblies representing different countries, regions, and global issues.

Technical challenge and our results. Our proposal is undoubtedly radical and we acknowledge that the devil is in the details; we discuss some limitations in Section 6. Our goal in this paper is to address a key, technically challenging question that arises as we consider the implementation of federated assemblies: how should they be selected?

In the context of federated assemblies, we think of an assembly as satisfying descriptive representation if it reflects both its child assemblies and its constituents. Specifically, we wish to design a random selection process, where the members of each assembly are selected from its child assemblies, so that the following constraints are satisfied:

- *Individual Representation:* Let the assembly's *population* be the union of all (possibly overlapping) populations of its descendant leaf assemblies. Each member of this population should have an equal probability of being represented on the assembly. This constraint can be interpreted as realizing an *equality of power* ideal.
- *Ex ante representation of child assemblies:* The expected number of seats allocated to each child assembly should be proportional to the child assembly's population.[1]
- *Ex post representation of child assemblies:* The number of seats allocated to each child assembly should be proportional to the child assembly's population, rounded down, *ex post*. This ex post guarantee prevents situations where an unlucky draw leads to significant under-representation;[2] it mirrors ex post quotas imposed on different features in the selection of standard citizens' assemblies.

Our results are primarily positive, showing that achieving various properties together is indeed tractable.

We begin by considering only the first two properties, individual representation and ex ante representation of child assemblies (Section 3). We design a simple algorithm (Algorithm 1) that is able to achieve both of these properties.

Next, we throw ex post child representation into the mix (Section 4). This makes the problem much more challenging, so we focus on instances with additional structure. We begin with a very natural class, which we call *laminar instances*, where the assembly graph is a tree, and members are signed up for only one leaf assembly. This captures instances where assemblies represent hierarchical regions, e.g., city-level which feed into state-level which feed into national-level. For such instances, we give an algorithm (Algorithm 2) that achieves all of our desired properties. Next, we generalize laminar instances to a larger class, *semi-laminar*. These are rooted in a laminar instance, with multiple assemblies at each node that federate together, allowing constituents to, for example, organize both geographically and according to shared interests. It turns out that even this level of generality already adds quite a bit of complexity to the problem. We nonetheless devise a surprisingly intricate algorithm (Algorithm 3), under mild regularity conditions, is able to achieve individual representation, ex ante representation of child assemblies, and approximate ex post representation of child assemblies up to an additive error of one.

Finally, in Section 5, we implement an algorithm based on column generation for convex programming, which, given any instance of our problem, computes a distribution satisfying the three properties. In addition to measuring the running time of the algorithm, our empirical results suggest that all of our properties can be achieved simultaneously in the general case, at least in practice.

Related work. There is a growing body of work on algorithms for randomly selecting citizens' assemblies, starting with the paper by Flanigan et al. (2021), along with many followups (Flanigan et al. 2020; Flanigan, Kehne, and Procaccia 2021; Ebadian et al. 2022; Flanigan et al. 2024). The key challenge these papers address arises because of quotas imposed on multiple, overlapping features. By contrast, our community representation constraint amounts to stratified sampling with respect to a partition of the population, which is simple in the case of standard citizens' assemblies (Benadè, Gölz, and Procaccia 2019). The difficulty of our problem stems from its graph structure and the need to also represent child assemblies, leading to technical questions that are quite different from prior work. We do note that the mathematical programming approach used by Flanigan et al. (2021) to implement their (widely used) algorithm also works well in our case.

Conceptually, federated assemblies are somewhat related to *pyramidal democracy* (Pivato 2009). In this scheme, citizens self-organize into small groups at the bottom of the pyramid, with each group nominating a delegate. In the next level, those delegates form groups, each again nominates a

[1] When calculating a child's proportional share, if some populations overlap, we first split the weight of members in the intersection equally across their populations, e.g., if a member is in three child populations, they only contribute 1/3 to each.

[2] As a concrete example, consider selecting an assembly of size 50 to represent 10 equally sized districts, each deserving five seats. Selecting 50 people uniformly at random will satisfy individual representation and ex ante guarantees. However, with probability greater than 5%, there will be a district without even a single representative on the panel.

delegate, and so on. Our proposal differs in several significant ways, but most importantly, the selection of delegates is not random in pyramidal democracy — rather, it is up to each group to decide how to select its delegate — and there are no representation requirements; by contrast, the whole point of our work is to design a random selection process that satisfies representation constraints.

Our technical contribution fits more broadly within the literature on dependent rounding (Gandhi et al. 2006). Within social choice, there are many works trying to achieve both ex ante and ex post guarantees, the closest to ours likely being work on apportionment (Correa et al. 2024) and randomized committee selection (Aziz et al. 2023). In randomized rounding more broadly, arguably the closest in flavor is randomized rounding for flows (Raghavan and Tompson 1987), which has constraints similar to our inheritance ones. Furthermore, rounding the number of seats given to each child assembly (a step in several of our algorithms) is reminiscent of randomized apportionment (Balinski and Young 2010; Pukelsheim 2017). However, our inheritance and ex post child constraints do not fit neatly within existing frameworks and make it challenging to use off-the-shelf techniques directly. Instead, many of our results repeatedly invoke existing schemes, such as variations of the Birkhoff Von-Neumann Theorem (Birkhoff 1946; Budish et al. 2013), in nontrivial ways. We discuss these techniques in more detail when we use them.

2 Model

Let N be a finite set of people and G be a directed acyclic graph. Abusing notation slightly, we write $v \in G$ if v is a node in G.

For a node $v \in G$, let $\textsc{children}(v) = \{v' : v \to v' \text{ in } G\}$ be the set of nodes v' with a directed edge from v to v'. Let $\textsc{leaves}(G) = \{v : |\textsc{children}(v)| = 0\}$ be the leaf nodes of G. Each person $i \in N$ will be signed up for a nonempty set $L_i \subseteq \textsc{leaves}(G)$ of leaf nodes. For a set $L \subseteq \textsc{leaves}(G)$, we will also use $C^L = \{i : L_i = L\}$ for the set of people signed up for exactly the leaf nodes L. We refer to each C^L as an *equivalence class*, which collectively form a partition of N. For a leaf node $\ell \in G$, its *population* $N_\ell = \{i : \ell \in L_i\}$ is the set of all people signed up for it.

Let $\textsc{feds}(G) = \{v : |\textsc{children}(v)| > 0\}$ be the internal nodes of G, which we refer to as *federations*. For a federation $f \in \textsc{feds}(G)$, let $\textsc{descendants}(f) \subseteq \textsc{leaves}(G)$ be the set of all leaf nodes reachable from f in G. For a federation $f \in \textsc{feds}(G)$, its population is defined as $N_f := \bigcup_{\ell \in \textsc{descendants}(f)} N_\ell$. Note that this can equivalently be defined in terms of equivalence classes as $N_f = \bigcup_{L : L \cap \textsc{descendants}(f) \neq \emptyset} C^L$.

An *instance* is a tuple $\mathcal{I} = \langle G, (N_v)_{v \in G} \rangle$ of the graph and the membership relationships. Given an instance $\mathcal{I}$ and a *target assembly size* n, our goal is to choose an *assembly* $A_v \subseteq N_v$ of size n for each node v.[3] We will assume, in general, that each $|N_v| \geq n$ so that this is always possible. Furthermore, we require that each federation's assembly be drawn from its child assemblies, i.e., for $f \in \textsc{feds}(G)$, $A_f \subseteq \bigcup_{c \in \textsc{children}(f)} A_c$. We call this the *inheritance* property. A vector $\boldsymbol{A} = (A_v)_{v \in G}$ satisfying these requirements is an *assembly assignment* (or simply an assignment). Our algorithm for selecting assembly assignments will be random, and thus, their outputs will be distributions over assembly assignments, $\boldsymbol{\mathcal{A}}$. We will call such a distribution a *randomized assembly assignment* and use $\mathcal{A}_v$ to refer to the marginal distribution for the assembly at node v.

We would like our randomized assembly assignments to satisfy various properties. Some are ex post and should hold for all assignments in the support. Others will be ex ante and are simply properties of the distribution that hold in expectation.

Desired Properties. Arguably, the most important requirement is *individual representation*. A randomized assignment $\boldsymbol{\mathcal{A}}$ satisfies *individual representation* if for each node v and $i \in N_v$, $\Pr[i \in \mathcal{A}_v] = n/|N_v|$, that is, each person has an equal chance of being selected to the assembly.

The other flavor of requirements we have on solutions are with respect to child assemblies. For a federation $f \in \textsc{feds}(G)$ and child $c \in \textsc{children}(f)$, we think of $|A_f \cap A_c|$ as the number of seats child c is allocated, and we would like this allocation to be at least as large as c "deserves." The question, however, is how to set these bounds. If the child populations N_c of a federation$f$ are all pairwise disjoint, then a natural choice is that each c should be allocated an $|N_c|/|N_f|$ fraction of the n seats in A_f. For non-disjoint child populations, we generalize this by splitting a person's weight equally among all child populations they are a part of. More formally, for a federation f and member $i \in N_f$, define the *multiplicity of i at f* to be $m(i, f) = |\{c \in \textsc{children}(f) : i \in N_c\}|$, the number of child nodes they are a member of. The *weighted population size* $w_{c,f}$ of a child federation $c \in \textsc{children}(f)$ is defined by $w_{c,f} = \sum_{i \in N_c} \frac{1}{m(i,f)}$. Note that the definition implies that $|N_f| = \sum_{c \in \textsc{children}(f)} w_{c,f}$. Hence, we would say that node c *deserves* a $w_{c,f}/|N_f|$ fraction of the seats in A_f. Define $q_{c,f} := w_{c,f}/|N_f|$ to be this fraction.

However, $n \cdot q_{c,f}$ need not be an integer. Hence, we consider two notions of fair child representation. First, we say that a randomized assignment $\boldsymbol{\mathcal{A}}$ satisfies *ex ante child representation* if, for each federation $f \in \textsc{feds}(G)$ and child $c \in \textsc{children}(f)$, $\mathbb{E}[|\mathcal{A}_f \cap \mathcal{A}_c|] \geq n \cdot q_{c,f}$. Similarly, we will say that a randomized assignment satisfies *ex post child representation* if for each assignment $\boldsymbol{A}$ in the support and each federation $f \in \textsc{feds}(G)$ and child $c \in \textsc{children}(f)$, $|A_f \cap A_c| \geq \lfloor n \cdot q_{c,f} \rfloor$. In other words, even if it is impossible to guarantee exact child representation, we can ensure that all children get at least the nuimber of seats they deserve rounded down.

3 Ex Ante Child Representation

We begin our study with an algorithm that samples from a distribution satisfying inheritance, individual representation, and ex ante child representation, proving that these three properties are compatible. To gain intuition, we first informally present a simpler version of the algorithm tailored to

[3] Note that unlike standard social choice notation, n is not the total number of voters $|N|$.

Algorithm 1: Selection algorithm with ex ante child representation guarantees

```
Input Graph $G$, populations $(N_v)_{v \in G}$, and assembly size $n$
Output Assembly assignment $(A_v)_{v \in G}$
Choose a linear order $\succ$ over $N$ uniformly at random
for $v \in G$ do
    Let $A_v$ be the $n$ highest ranked members of $N_v$ in $\succ$.
return $(A_v)_{v \in G}$
```

Algorithm 2: Selection algorithm for laminar instances with ex post child representation guarantees

```
Input Graph $G$, populations $(N_v)_{v \in G}$, and assembly size $n$
Output Assembly assignment $(A_v)_{v \in G}$
1: for leaf $\ell \in \text{LEAVES}(G)$ do
2:     Choose $A_\ell \subseteq N_\ell$ uniformly at random
3: for federation $f \in \text{FEDS}(G)$ in a topological sort do
4:     Round $(n \cdot q_{c,f})_{c \in \text{CHILDREN}(f)}$ to an integral vector
           $(s_{c,f})_{c \in \text{CHILDREN}(f)}$ such that each $s_{c,f} \geq \lfloor w_{c,f} \rfloor$,
           $\mathbb{E}[s_c] = n \cdot w_{c,f}$, and $\sum_c s_{c,f} = n$
5:     Select $B_{c,f} \subseteq A_c$ with $|B_{c,f}| = s_{c,f}$ uniformly at
           random.
6:     Let $A_f \leftarrow \bigcup_c B_{c,f}$.
7: return $(A_v)_{v \in G}$
```

the special case of assemblies of size $n = 1$. It works by selecting a single random order of all of N uniformly at random; the representative for node v is simply the highest-ranked member of N_v. Note that this allows for strikingly simple arguments for the various properties. Indeed:

- *Inheritance*: If $A_f = \{i\}$ for a federation f then $i \in N_c$ for some $c \in \text{CHILDREN}(f)$. Since $N_c \subseteq N_f$, i will be maximal among N_c, and hence, $A_c = \{i\}$.
- *Individual Representation*: Each $i \in N_v$ is equally likely to be the maximally ranked person among N_v, so they are a member of A_v with probability $1/|N_v|$.
- *Ex ante child representation*: A similar argument to inheritance implies that $A_c = A_f$ with probability $|N_c|/|N_f| \geq w_{c,f}$.

We next show that this can generalize to the case of assemblies of arbitrary size by selecting A_v to be the n highest ranked members of N_v.

Theorem 1. *Algorithm 1 satisfies individual representation and ex ante child representation.*

Proof. We begin by showing inheritance. Fix an arbtirary sample $\succ$ and an internal node f. Suppose $i \in A_f$. Let $c \in \text{CHILDREN}(f)$ be a child such that $i \in N_c$. Note that i is one of the n highest members of N_f in $\succ$. Since $N_c \subseteq N_f$, i must be one of the n highest ranked members of N_c. Therefore, $i \in A_c$. This implies that $A_f \subseteq \bigcup_{c \in \text{CHILDREN}(f)} A_c$.

Next, we show individual representation. Fix a node v and suppose $i \in N_v$. Since $\succ$ restricted to N_v is a randomly selected permutation of N_v, the probability that i is in one of the top n positions is $n/|N_v|$ by symmetry. Hence, $i \in A_v$ with probability $n/|N_v|$.

Finally, we show ex ante child representation. Fix a federation f and a child $c \in f$. As for individual representation, whenever there is a person $i \in N_c$ such that $i \in A_f$, it is also the case that $i \in A_c$. Now, by individual representation, $i \in A_f$ with probability $n/|N_f|$. Furthermore, by linearity of expectation,

$$\begin{aligned}\mathbb{E}[|A_c \cap A_f|] &= \mathbb{E}[\sum_{i \in N_c} \mathbb{I}[i \in A_c \text{ and } i \in A_f] \\ &= \frac{|N_c| \cdot n}{|N_f|} \geq q_{c,f} \cdot n,\end{aligned}$$

completing the proof. □

4 Ex Post Child Representation

In this section, we add ex post child representation to our list of requirements, albeit at a cost to the generality of our results.

Laminar Instances

We begin with a more restricted structure that captures many practical potential implementations of federated assemblies. The assumption is that G is a tree and that each $i \in N$ is signed up for exactly one leaf node, i.e., $|L_i| = 1$. This captures settings such as where the assemblies represent regions that form a hierarchy, e.g., city-level assemblies, which feed into state-level assemblies, which feed into national-level assemblies, and possibly beyond. Following set-theoretic terminology, we call instances satisfying these restrictions *laminar*.

Algorithm 2 works by going through the federations, allocating an integral number of seats to each child, and filling these seats directly from the child's assembly. The rounding (Line 4) can be done in a variety of ways using tools from the dependent rounding literature. Brewer and Hanif (1983) provide a number of classical statistical methods for this; canonical examples from randomized algorithm design include *pipage rounding* (Ageev and Sviridenko 2004) and variations of the *Birkhoff-von Neumann Theorem* (Birkhoff 1946; Budish et al. 2013).

Theorem 2. *Algorithm 2 satisfies individual representation, ex ante child representation, and ex post child representation on laminar instances.*

Before formally giving a formal proof of the properties, we give some intuition by discussing the naturalness of Algorithm 2, which allows for its relatively simple analysis. Specifically, we enforce ex ante and ex post child representation by first allocating the "correct" number of seats to each child. We then go through the tree iteratively, selecting members from their (already determined) child assemblies. We may hope that such ideas could be generalized to non-laminar instances, first allocating seats and then iteratively selecting members from the child assemblies, perhaps not uniformly at random, to account for people being in potentially different numbers of children. However, we give an example where this is not the case —relaxing either of the laminar assumptions (that G is a tree or that each $|L_i| = 1$) can lead to instances where satisfying all the properties is impossible using such algorithms. Instead, it is necessary to induce some other forms of correlation or relax the iterativeness, a challenge that leads to more complicated algorithms. We discuss this more formally in the full version.

Proof of Theorem 2. Showing inheritance, ex ante child representation, and ex post child representation follow immediately from the definition of the algorithm.

It remains to establish individual representation. Fix a person i, and let $\ell \in \textsc{leaves}(G)$ be the leaf node they are signed up for. Note that A_ℓ is simply a random sample of n people from N_ℓ, so clearly $\Pr[i \in \mathcal{A}_\ell] = n/|N_\ell|$. Next, fix a federation $f^* \in \textsc{feds}(G)$ such that $i \in N_{f^*}$. Consider running the algorithm in a different order, sampling all vectors $(s_{c,f})_{c \in \textsc{children} f}$ at the beginning before starting the algorithm, and then running according to these samples. Note that this leads to an equivalent process because each $(s_{c,f})_{c \in \textsc{children} f}$ is sampled independently of everything else in the algorithm. Condition on a specific sample of these vectors. Let $\ell = v_0, v_1, \ldots, v_k = f$ be the path in G that leads from ℓ to f. Note that we can now directly compute the probability $i \in A_{f^*}$ because the only way to do so is if $i \in A_\ell$ and $i \in B_{v_{j-1}, v_j}$ for each $j \geq 1$. Hence, this probability is exactly

$$\frac{n}{|N_\ell|} \cdot \prod_{j=1}^{k} \frac{s^{v_{j-1}, v_j}}{n}.$$

To get the unconditional probability, we can simply take the expectation over all s values, i.e.,

$$\mathbb{E}\left[\frac{n}{|N_\ell|} \cdot \prod_{j=1}^{k} \frac{s^{v_j}_{v_{j-1}}}{n}\right].$$

Since each $s_{c,f}$ and $s_{c',f'}$ is sampled independently for $f \neq f'$, we can push the expectation in to get equality

$$\frac{n}{|N_\ell|} \cdot \prod_{j=1}^{k} \frac{\mathbb{E}\left[s^{v_{j-1}, v_j}\right]}{n} = \frac{n}{|N_\ell|} \cdot \prod_{j=1}^{k} \frac{n \cdot q_{v_{j-1}, v_j}}{n}$$

$$= \frac{n}{|N_\ell|} \prod_{j=1}^{k} \frac{|N_{v_{j-1}}|}{|N_{v_j}|} = \frac{n}{|N_{f^*}|}. \square$$

Semi-Laminar Instances

We now turn to a generalization of laminar instances with a structure we view as quite practical for real-world implementation, as it allows people to organize both geographically and according to shared interests. Conceptually, there is an underlying laminar instance with graph R. In addition, there is a set $\mathcal{T}$ representing a set of *topics*. These may be various causes that people care about, say climate change or animal rights, or region-specific policy questions, such as budget allocation.

More specifically, the graph G has $|R| \cdot (|\mathcal{T}| + 1)$ nodes. $|R| \cdot |\mathcal{T}|$ of these are identified by members of $R \times \mathcal{T}$, i.e., there is a node (r, t) for each combination of region and topic. For each $t \in \mathcal{T}$, the set of nodes $R \times \{t\}$ is connected to form a copy of R. In addition, there is a set of nodes denoted $(r, *)$ for each $r \in R$. Each is a federation whose children are $\{r\} \times \mathcal{T}$. In other words, at each region, we have an assembly that represents all people with respect to all topics of that region.

In this graph, the leaves are nodes in $\textsc{leaves}(R) \times \mathcal{T}$. People can be signed up for any number of topics, but, as the underlying instance is laminar, we assume that each is a member of nodes in one region. Hence, we assume that each $L_i \subseteq \{r\} \times \mathcal{T}$ for some $r \in \textsc{leaves}(R)$.

Abusing notation slightly, we will write $N_r = \{i \mid L_i \subseteq \{r\} \times \mathcal{T}\}$ for all people in this region (technically $N_r = N_{(r,*)}$). Furthermore, a node (r, t) will have at most two parents in G: it will always have $(r, *)$ and possibly another node (r', t) if r has a parent (r') in R. Note that if (r', t) exists, the weighted population $w_{(r,t),(r',t)} = N_{(r,t)}$, the trivial weighting, because the children of (r', t) have disjoint populations. On the other hand, $w_{(r,t),(r,*)} = \sum_{i \in N_{(r,t)}} \frac{1}{|L_i|}$. For brevity, we will use $w_{r,t}$ to denote only the nontrivial weight of the node (r, t). Finally, note that, for $r \in \textsc{leaves}(R)$ and $T \subseteq \mathcal{T}$, we can write $C^{\{r\} \times T}$ for the equivalence class of people in leaf r signed up for topics T. Everybody must fall in exactly one of these equivalence classes.

We refer to an instance taking on the above structure as a *semi-laminar instance*. For such instances, we have the following algorithm shown in Algorithm 3, with additional helper functions formally defined in the full version.[4] The structure is essentially an extension of the algorithm for laminar instances. Ideally, for each topic, we could independently run Algorithm 2, and when we needed to select an assembly $(r, *)$, we could select the "correct" number of members from each $A_{(r,t)}$. However, this does not quite work because if people are signed up for more topics, this will lead to them ending up in $A_{(r,*)}$ more frequently. To account for this, we essentially partition each $A_{(r,t)}$ into two pieces, $B^{\textsc{sel}}_{r,t}$ of *(sel)ectable* people and $B^{\textsc{uns}}_{(r,t)}$ of *(uns)electable* people, and run a separate laminar-like algorithm for each of these. The key idea is that when selecting members of $A_{(r,t)}$ for $A_{(r,*)}$, we will only select from members in $B^{\textsc{sel}}_{r,t}$. While each $i \in N_{(r,t)}$ will have an equal chance of being in $A_{(r,t)}$, the more topics they are signed up for, the less likely they will be in $B^{\textsc{sel}}_{r,t}$, so that, in aggregate, this will not increase their chances of being in $A_{(r,*)}$.

It should be said that the real complexity of this algorithm is hidden in the dependent rounding schemes of the helper functions. It requires careful balance and specific constraints to ensure the probabilities are exact and not subject to strange correlations that naïve implementations can impart. Furthermore, we need to avoid scenarios where the same person is selected twice for $A_{(r,*)}$ from two separate topics, which would not appear to have an easy solution. All of these schemes are implemented using variations of the Birkhoff-von Neumann algorithm. Budish et al. (2013) give a class of constraints that are always possible to guarantee when rounding; we ensure that all of our rounding procedures take this form.

[4] https://arxiv.org/abs/2405.19129

Algorithm 3: Algorithm for semi-laminar instances

```
Input Semi-laminar instance with graph $G$, populations $(N_v)_{v \in G}$, and assembly size $n$
Output Assembly assignment $(A_v)_{v \in G}$
for $(r, t) \in R \times \mathcal{T}$ do
    $s_{r,t} \leftarrow \textsc{Round}(n \cdot \frac{w_{r,t}}{|N_{r,t}|})$
for $r \in \textsc{leaves}(R)$ do
    $(B^{\text{SEL}}_{r,t}, B^{\text{UNS}}_{r,t})_{t \in T} \leftarrow \textsc{SampleLeaves}\left((C^{\{r\} \times T})_{T \subseteq \mathcal{T}}, (s_{r,t})_{t \in \mathcal{T}}, n\right)$
for $r \in \textsc{feds}(R)$ in a topological sort do
    for $t \in \mathcal{T}$ do
        $B^{\text{SEL}}_{r,t}, B^{\text{SEL}}_{r,t} \leftarrow \textsc{SampleFromChildren}\big(s_{r,t}, (B^{\text{SEL}}_{c,t}, B^{\text{UNS}}_{c,t}, w_{c,t}, |N_{c,t}|)_{c \in \textsc{children}(r)}\big)$
for $r \in R$ do
    for $t \in \mathcal{T}$ do
        $A_{(r,t)} \leftarrow B^{\text{SEL}}_{r,t} \cup B^{\text{UNS}}_{r,t}$
    $A_{(r,*)} \leftarrow \textsc{RoundAndSample}(n, (B^{\text{SEL}}_{r,t}, w_{r,t})_{t \in \mathcal{T}})$
return $(A_v)_{v \in G}$
```

Finally, note that all of these extra complexities mean that we impose some additional mild regularity conditions and achieve slightly degraded guarantees, at least for ex post child representation. The regularity conditions require that no weighted population is too close to zero or its maximum, and that no child population is too dominant within its parent.

Theorem 3. *Fix a semi-laminar instance and assembly size n. Suppose there exist $\varepsilon, \delta > 0$ such that (i) for all $r, t \in R \times \mathcal{T}$, $\varepsilon \cdot |N_{r,t}| \le w_{r,t} \le (1 - \varepsilon) \cdot |N_{r,t}|$, (ii) for all federations $f \in \textsc{feds}(G)$ and $c \in \textsc{children}(f)$, $|N_c|/|N_f| \le 1 - \delta$, and (iii) $n \ge \frac{2}{\varepsilon \cdot \delta}$. Furthermore, suppose each $|N_v| \ge 4n$ and for all nonempty equivalence classes C^L, $|C^L| \ge 2$. Then, Algorithm 3 run on this instance satisfies individual representation, ex ante child representation, and approximate ex post child representation in that $|A_f \cap A_c| \ge \lfloor n \cdot q_{c,f} \rfloor - 1$ for all $f \in \textsc{feds}(G)$ and $c \in \textsc{children}(f)$.*

The proof is relegated to the full version. It is quite intricate. The first challenge is to show that every sampling step is feasible and that the algorithm can run to completion. This depends on the regularity conditions, and we must step through each call individually, proving that the output *also* satisfies another condition that we can use in subsequent steps. Furthermore, we must check that all probabilities exactly work out to match the ex ante guarantees. This is made simpler by certain randomized rounding choices which prevent a participant from simultaneously being in multiple child assemblies, and allow us to decompose the probability computations.

5 Experiments

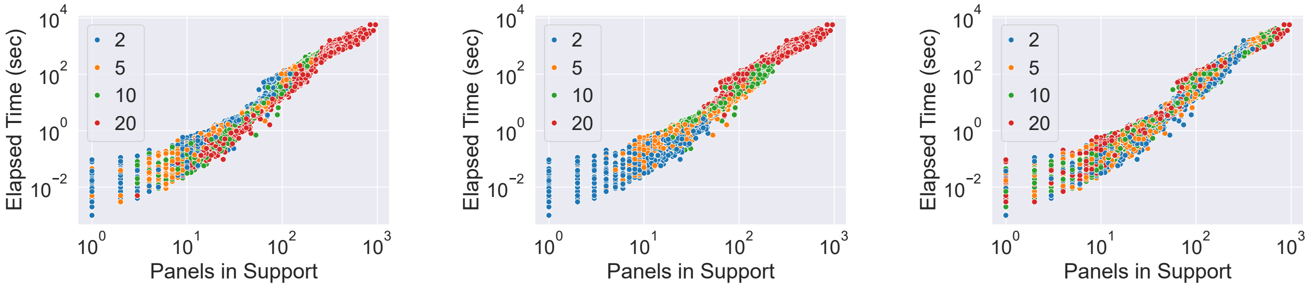
While we have given efficient algorithms for finding randomized solutions satisfying various properties in special cases, we focus now on how easy it is in practice to find randomized assignments in generality. To this end, we sample several thousand instances and use a brute force algorithm to try to find a randomized assignment satisfying all of our ex post and ex ante guarantees.

Algorithm. The computation of randomized assignments satisfying our guarantees is quite challenging. Our algorithm for this task uses convex optimization and integer linear programming (ILP) as subroutines. It is inspired by a related algorithm used for standalone citizens' assemblies, which is an extension of column generation to convex programming (Flanigan et al. 2021). At a high level, we convert our problem to a smoother optimization problem by defining a convex loss over randomized assignments measuring how far a distribution is from satisfying all ex ante guarantees.

The algorithm works as follows. It maintains a set of (deterministic) assembly assignments that all satisfy ex post guarantees $S = \{A^1, A^2, \ldots\}$. We begin with S of size one containing an arbitrary assembly assignment, which can be found by solving an ILP. It then alternates between two steps:

1. It optimizes over distributions with a support of S, finding one that minimizes the squared error over ex ante misrepresentation. In other words, it must choose among probability vectors $\mathbf{p} = (p_1, \ldots, p_{|S|})$ which correspond to selecting each panel $A^i \in S$ with probability p_i. For every ex ante property that should have expected value v, if the distribution has expected value v', then $\mathbf{p}$ incurs a loss of $(v - v')^2$ (summed up over all guarantees). This is a convex objective with linear constraints, and can therefore be solved using convex programming. Furthermore, if the output distribution achieves zero error (or, more accurately, some small additive error of .1% to avoid numerical issues), then every ex ante guarantee is satisfied; in that case, we have found a solution and subsequently return it.
2. If the optimal distribution found from step one does not achieve zero error, then we expand S. For each ex ante guarantee, we have a "loss gradient" of $-2v'(v - v')$ for each guarantee. We then find an assembly assignment satisfying the ex post guarantees that minimizes the inner product with this gradient. In other words, we find a feasible assignment A that brings us closest to the true optimal distribution. Finding this assignment amounts to solving an ILP. We then add A to S and return to step one.

By the same reasoning as in Flanigan et al. (2021), using



(a) Instances colored by the number of equivalence classes.

(b) Instances colored by the number of federations.

(c) Instances colored by the assembly size.

Figure 1: Scatter plots showing the time taken and number of assembly assignments in the support for each of the instances we ran on. Sub-plots show the same plot, colored by a parameter.

standard techniques from column generation, this algorithm will always find an optimal distribution satisfying ex ante and ex post guarantees if such a distribution exists. Unfortunately, however, there is no upper bound on the panels this algorithm will add to S before terminating.

Experimental setup. To sample instances, we first fix a number of equivalence classes (2, 5, 10, or 20) and sample their relative sizes from an exponential distribution. Next, we iterate through a number of federations (2, 5, 10, or 20). Each federation has a randomly selected set of already defined federations or equivalence classes as children.[5] This method allows for arbitrary DAGs to be sampled. We then test these instances with various assembly sizes n (2, 5, 10, or 20).

For each combination of parameters, we sampled and ran 100 instances. All optimizations were solved using Gurobi on an Amazon Web Services (AWS) instance with 128 vCPUs of a 3rd Gen AMD EPYC running at 3.6GHz equipped with 1TB of RAM. We were able to run 64 instances in parallel, giving each thread two processors. Depending on the size of the instance, computation took anywhere from under a tenth of a second to multiple hours.

Results. Of the 6400 instances, on all but 15, the algorithm terminated, returning an optimal solution. These 15 had only the largest number of equivalence classes and federations (20 each). With better optimization, in practice, the problem of finding a randomized assignment satisfying all of our guarantees appears to be both feasible and tractable.

For the >99.7% of instances that did terminate, we use two metrics to measure the complexity of finding the distributions: the elapsed time and the number of assembly assignments added to the support. A scatter plot showing the relationship between these parameters is given in Figure 1. Each subfigure shows the same points but colors them depending on a different parameter so that we may see its effect. Overall, we see that both of these complexity measures are quite similar. Furthermore, increasing the number of equivalence classes and federations strongly correlates with increased complexity while assembly size appears to be relatively unimportant.

6 Discussion

The democratic governance of large-scale digital communities is an open problem. Key challenges include, first, the penetration of fake and duplicate digital identities (a.k.a. sybils), and second, the perils of large-scale online voting, which is considered to be untenable by some leading experts (Park et al. 2021). Federated assemblies can be viewed as a step in an effort to address these challenges. Our approach may address sybils by having the laminar core built from small local communities in which members know each other to be genuine and from communities that federate only if they trust each other to be genuine, for example by having sufficient intersection or actual relationships among them to base this trust on. Large-scale online voting is a nonissue as every federation, no matter how large, is governed by an assembly that engages in "small-scale" democracy.

Our approach does have some limitations. We modeled the problem as static, focusing on sampling a single randomized assignment fairly. In practice, federations are dynamic and assemblies must be periodically updated. On the surface, this is not an inherent limitation, as we could resample fresh assemblies every fixed amount of time. However, this may get more challenging if the system is more dynamic with members coming and going, and may hope to be "ex-post fair over time" ensuring bad randomness leaves no group is consistently receiving the short end of the stick. Furthermore, we may wish to allow for local changes to occur without completely refreshing all assemblies simultaneously, say rotating people in and out one at a time. Modeling this well and defining useful "over-time" fairness properties seems to be potentially impactful future work.

Finally, although we analytically solve well-motivated special cases, we leave open whether a randomized assignment satisfying all of our desiderata exists in the general case. In our extensive experiments, we have not found any infeasible instances, and we are therefore optimistic that existence can be guaranteed.

[5] This means that we allow federations to have direct members and child assemblies. However, this generality only makes the problem harder.

References

Ageev, A. A.; and Sviridenko, M. I. 2004. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8: 307–328.

Aziz, H.; Lu, X.; Suzuki, M.; Vollen, J.; and Walsh, T. 2023. Best-of-both-worlds fairness in committee voting. In *Proceedings of the 19th Conference on Web and Internet Economics (WINE)*, 676.

Balinski, M. L.; and Young, H. P. 2010. *Fair Representation: Meeting the Ideal of One Man, One Vote*. Rowman & Littlefield.

Benadè, G.; Gölz, P.; and Procaccia, A. D. 2019. No Stratification Without Representation. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, 281–314.

Birkhoff, G. 1946. Three observations on linear algebra. *Universidad Nacional de Tucumán, Revista A*, 5: 147–151.

Brewer, K. R. W.; and Hanif, M. 1983. An introduction to sampling with unequal probabilities. In *Sampling With Unequal Probabilities*, 1–19. Springer.

Budish, E.; Che, Y.-K.; Kojima, F.; and Milgrom, P. 2013. Designing random allocation mechanisms: Theory and applications. *American Economic Review*, 103(2): 585–623.

Clegg, N. 2023. Bringing People Together to Inform Decision-Making on Generative AI. Blog post.

Correa, J.; Gölz, P.; Schmidt-Kraepelin, U.; Tucker-Foltz, J.; and Verdugo, V. 2024. Monotone Randomized Apportionment. *arXiv preprint arXiv:2405.03687*.

Ebadian, S.; Kehne, G.; Micha, E.; Procaccia, A. D.; and Shah, N. 2022. Is Sortition Both Representative and Fair? In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Fishkin, J. 2018. *Democracy When the People Are Thinking: Revitalizing Our Politics Through Public Deliberation*. Oxford University Press.

Flanigan, B.; Gölz, P.; Gupta, A.; Hennig, B.; and Procaccia, A. D. 2021. Fair Algorithms for Selecting Citizens' Assemblies. *Nature*, 596: 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 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Flanigan, B.; Kehne, G.; and Procaccia, A. D. 2021. Fair Sortition Made Transparent. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 25720–25731.

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

Gandhi, R.; Khuller, S.; Parthasarathy, S.; and Srinivasan, A. 2006. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3): 324–360.

Jones, N.; and Marsden, H. 2010. Associative Democracy. In H. Anheier, S. T., ed., *International Encyclopedia of Civil Society*. Springer.

Landemore, H. 2020. *Open Democracy: Reinventing Popular Rule for the Twenty-First Century*. Princeton University Press.

Park, S.; Specter, M.; Narula, N.; and Rivest, R. L. 2021. Going from bad to worse: From internet voting to blockchain voting. *Journal of Cybersecurity*, 7(1): tyaa025.

Pivato, M. 2009. Pyramidal Democracy. *Journal of Public Deliberation*, 5(1): article 8.

Procaccia, A. D. 2022. A More Perfect Algorithm. *Scientific American*, 327(5): 52–59.

Pukelsheim, F. 2017. *Proportional Representation*. Springer.

Raghavan, P.; and Tompson, C. D. 1987. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4): 365–374.

Shapiro, E. 2023a. Grassroots Distributed Systems: Concept, Examples, Implementation and Applications. *arXiv preprint arXiv:2301.04391*.

Shapiro, E. 2023b. Grassroots Social Networking: Serverless, Permissionless Protocols for Twitter/LinkedIn/WhatsApp. In *Proceedings of the 3rd International Workshop on Open Challenges in Online Social Networks (OASIS)*.

Shapiro, E. 2024. A Grassroots Architecture to Supplant Global Digital Platforms by a Global Digital Democracy. *arXiv preprint arXiv:2404.13468*.

Stone, P. 2011. *The Luck of the Draw: The Role of Lotteries in Decision Making*. Oxford University Press.

Van Reybrouck, D. 2016. *Against Elections: The Case for Democracy*. Random House.