Manipulation-Robust Selection of Citizens' Assemblies

Bailey Flanigan[1], **Jennifer Liang**[2], **Ariel D. Procaccia**[2], **Sven Wang**[3]

[1] Carnegie Mellon University
[2] Harvard University
[3] Massachusetts Institute of Technology

Abstract

Among the recent work on designing algorithms for selecting citizens' assembly participants, one key property of these algorithms has not yet been studied: their *manipulability*. Strategic manipulation is a concern because these algorithms must satisfy representation constraints according to volunteers' *self-reported* features; misreporting these features could thereby increase a volunteer's chance of being selected, decrease someone else's chance, and/or increase the expected number of seats given to their group. Strikingly, we show that *Leximin* — an algorithm that is widely used for its fairness — is highly manipulable in this way. We then introduce a new class of selection algorithms that use ℓ_p norms as objective functions. We show that the manipulability of the ℓ_p-based algorithm decreases in $O(1/n^{1-1/p})$ as the number of volunteers n grows, approaching the optimal rate of $O(1/n)$ as $p \to \infty$. These theoretical results are confirmed via experiments in eight real-world datasets.

1 Introduction

In a *citizens' assembly*, a panel of randomly-chosen constituents convenes to make a policy recommendation on a political issue. Although citizens' assembly participants are not career politicians, their recommendations are informed by an extensive process of learning from experts and deliberating with one another. As such, citizens' assemblies are appealing because they combine the goals of engaging everyday citizens in democratic decision-making, while also facilitating informed decisions. Citizens' Assemblies are now being used to make increasingly high-profile decisions around the world (Participedia 2023); for example, France recently ran a national-level assembly on the topic of assisted dying, and its outcome is slated to affect policy on palliative care (Bürgerrat 2023).

Because the participants of a citizens' assembly represent their entire underlying constituency, the process by which they are selected is crucial to whether the policy recommendation they produce is perceived as trustworthy. The importance of this selection process has motivated a growing body of research on *selection algorithms* (Ebadian and Micha 2023; Ebadian et al. 2022; Flanigan et al. 2020, 2021; Flanigan, Kehne, and Procaccia 2021), which solve the following task: from among a *pool* of volunteers, randomly sample a *panel* that is (at least approximately) *descriptively representative* of the underlying population. This means that if the population is 48% women, the panel should be approximately 48% women. Because exact representation of all identities cannot be achieved with a finite-size panel, practitioners' main goal is to achieve representation with respect to a handful of key features, such as gender, age, geographic location, education level, and opinion on the issue at hand.

The main algorithmic challenge in selecting descriptively representative participants is *self-selection bias*: different demographic groups agree to participate at vastly different rates, so the pool of volunteers from which the panel is sampled is demographically skewed compared to the underlying population. Consequently, simple sampling techniques do not produce the desired descriptive representation.

Existing work has circumvented the challenge of achieving representation to a large degree. The first selection algorithms, developed by practitioners, were heuristics that searched for representative panels, injecting randomness wherever possible. More recent work has contributed algorithms that not only find representative panels, but do so in a way that achieves other desiderata simultaneously. For example, Flanigan et al. (2021) presents a framework of algorithms that are *maximally fair* to individual pool members: that is, they make pool members' probabilities of being selected as equal as possible, subject to representation constraints. One algorithm within this framework, called *Leximin* (Flanigan et al. 2021), is now widely used in practice.

Beyond the desiderata of representation and maximal fairness, follow-up work has contributed methods for additionally achieving *transparency* (Flanigan, Kehne, and Procaccia 2021). However, at the current frontier of research on selection algorithms, a key desideratum remains yet untouched: their *manipulability*.

In this paper, we initiate the study of selection algorithms' vulnerability to perhaps the most salient type of potential manipulation: *volunteers misreporting their features*. With Theorem 1.1, we now illustrate in detail why the selection process, as it commonly works in practice, can permit — and strongly incentivize — such manipulation.

Example 1.1. We want to select a panel of 10 people to convene on climate policy. We care about descriptive repre-

sentation of one feature only: people's level of concern about climate change. This feature has two possible values: those who are *less concerned* (20% of the population) and *more concerned* (80% of the population). Thus, we will reserve 2 and 8 panel seats for these respective groups.

STAGE 1: RECRUITING THE POOL OF VOLUNTEERS. We send out invitations to 1000 uniformly sampled households in our constituency. In response, 100 people volunteer to participate, but they are strongly self-selected: only 4 are truly *less concerned*, and 96 of them are truly *more concerned*.[1] In preparation for selection, we ask all 100 volunteers to report which group they belong to. Among these volunteers, suppose there is one strategic agent i who is truly *more concerned*, but is willing to misreport their group membership if it increases their chance of being on the panel.

STAGE 2: PANEL SELECTION. Given this pool of volunteers and their self-reported group memberships, a selection algorithm is then used to choose a panel. We assume nothing about this algorithm except that it treats people in the same group uniformly, and it produces a panel with 2 seats for *less concerned* people and 8 seats for *more concerned* people.

It is not hard to see that, in this example, i benefits significantly from misreporting their group membership. If i truthfully reports they are *more concerned*, they will join a group of 96 people for whom the panel has 8 seats, and thus will be chosen with probability $8/96 \approx 8\%$. If i reports that they are *less concerned*, they will join a group of 5 people for whom the panel has 2 seats, and will be chosen with probability $2/5 = 40\%$. By misreporting that they are *less concerned*, i can increase their selection probability by almost 32%. Moreover, with probability 40%, i will be given a panel seat reserved for *less concerned* people, thereby giving the group of *more concerned* people an extra panel seat.

Theorem 1.1 illustrates why such manipulation is of practical concern: the nature of self-selection bias in this example would be fairly easy for constituents to anticipate — surely, people who care less about climate change will be less likely to volunteer — making the optimal manipulation public knowledge.[2] Moreover, we cannot always prevent manipulation through verification; here, people's opinions would be impossible to check. As citizens' assemblies are used for increasingly higher-profile decisions, the political power associated with participating — and thus the incentive to manipulate — will only increase. Theorem 1.1 also shows a fundamental impossibility: when there is self-selection bias, achieving descriptive representation *necessitates* giving different probabilities to different groups, thereby permitting manipulability. In other words, *no* selection algorithm can achieve representation while eliminating manipulation incentives. This motivates our research question:

Research question: What aspects of the selection process can we adjust in practice to *limit* agents' incentives to misreport their features?

[1]These numbers are based on a real-world panel selection task (instance *sf-e* in our empirical analysis).

[2]More generally, there are clear patterns across real-world instances of which groups tend to be most underrepresented among volunteers (e.g., those with less education).

Approach. We focus on two main aspects of the selection process that can be changed in practice: *the size of the pool of volunteers* n, and *the choice of selection algorithm*. The intuition for why increasing n could help is simple: as the pool grows, there are more volunteers per available panel seat. For the correct choice of selection algorithm, this could permit the decrease of *all* volunteers' selection probabilities, thereby diluting the potential gains of manipulation.

Among selection algorithms, we consider only algorithms that achieve maximal fairness, because per Theorem 1.1, manipulation incentives arise from *inequality* in selection probabilities (thus, the goal of equalizing selection probabilities is aligned with limiting manipulation). Specifically, we introduce and study *rounding-based* selection algorithms — a class of maximally fair algorithms that generalizes an algorithm of Flanigan et al. (2020). As discussed in Section 2, rounding-based algorithms closely reflect those used in practice, but enforce a slightly relaxed notion of representation.

Each rounding-based algorithm optimizes a different *fairness objective*: a function measuring *how fairly* the chance to participate is spread over volunteers. We study several such functions: *Leximin,* the objective most commonly used in real-world panel selection (Flanigan et al. 2021); *Nash Welfare*, which has known fairness and transparency properties and is available online for practical use (Flanigan, Kehne, and Procaccia 2021); and all ℓ_p norms, which we newly introduce to the citizens' assembly setting.

Results and Contributions. (1) Manipulation model. Our first contribution is to formally model three realistic manipulation incentives in the assembly selection context: increasing one's own probability of selection, changing someone else's, and — as we saw in Theorem 1.1 — misappropriating seats from other groups. **(2) Impossibilities for existing algorithms.** We then show that, somewhat alarmingly, the state-of-the-art objectives *Leximin* and *Nash Welfare* are *arbitrarily manipulable* on multiple of these counts. Even as n grows large, they permit agents to gain *probability 1* by misreporting, and they allow coalitions to misappropriate a constant fraction of the panel seats. These lower bounds give a key insight: fairness objectives are manipulable when they permit some agents to receive very high selection probabilities. **(3) An optimal selection algorithm.** Motivated by this finding, we study ℓ_p norms, which heavily penalize high probabilities due to their strong convexity. We show that even when agents can costlessly misreport any vector of features, the manipulability of the ℓ_p-norm declines in n at a rate $n^{-(1-1/p)}$, a rate which holds for all three notions of manipulability. We further show that *any selection algorithm* must suffer manipulability at least $\Omega(1/n)$; as $p \to \infty$, our upper bound approaches this lower bound, implying that the ℓ_∞ norm — the objective that minimizes the maximum selection probability — achieves optimal convergence. As a bonus, our analysis handles coalitions of size up to $\Theta(n)$. **(4) Empirical results.** We complement these theoretical results with experiments in eight real-world panel selection datasets. Our empirical results closely track our theory, showing that *Leximin* and *Nash Welfare* suffer high manipulability even as n grows, while the manipulability of the ℓ_2 and ℓ_∞ norms declines quickly.

2 Model

Foundations of Selection Algorithms

At a high level, a *selection algorithm* must select a panel of k agents from the pool of n agents. This panel must be representative of the population with respect to a predefined set of *features* F, where each $f \in F$ has a predefined set of possible *values* V_f. For example, the feature $f = age$ might have possible values $V_{age} = \{18\text{-}40, 41\text{-}60, 61+\}$. We assume that for each feature f, its possible values V_f are exhaustive and mutually exclusive. We define $FV := \bigcup_{f \in F} V_f$ to contain all *feature-value pairs*, (f, v) for all $f \in F, v \in V_f$. For all (f, v), $p_{(f,v)}$ is the fraction of the underlying population with value v for feature f. Then, a *representative* panel contains $p_{(f,v)} \cdot k$ agents with value v for feature f, for all $(f, v) \in FV$. Let $p := (p_{(f,v)} | f \in F, v \in V_f)$.

An *instance* of the panel selection task is then composed of population rates p; a desired panel size k; and the *pool* N, which is defined by all n agents' *true* values of each feature. To define these values, we let $f(i)$ denote i's value for f, thereby implicitly treating each feature as a function $f : [n] \to V_f$. i's values across features are summarized in their *feature vector* $w(i) := (f(i) | f \in F)$. The *pool* of volunteers $N := (w(i) | i \in [n])$ is then an n-tuple containing all agents' feature vectors. We let $\mathcal{W} := \prod_{f \in F} V_f$ be the collection of all possible feature vectors (i.e., all possible *intersections* of feature-value pairs). A generic feature vector is $w \in \mathcal{W}$. We will often reason only about *fractional composition* of a pool N, called $\boldsymbol{\nu}(N)$. This vector is indexed by feature-vector, with w-th entry $\nu_w(N) := |\{i \in [n] : w(i) = w\}|/|N|$ representing the fraction of the pool with vector w.

In practice, organizers must rely on agents to *report* their feature vectors. Agent i's *reported* feature vector is denoted $\tilde{w}(i) \in \mathcal{W}$; in general, we will use tilde $\tilde{\cdot}$ throughout the paper to distinguish reported values from true values. The *reported* pool is then denoted as $\tilde{N} = (\tilde{w}(i) | i \in [n])$. In an instance p, k, N, a *selection algorithm* $\mathcal{A}$ actually receives as input $p, k, \tilde{N}$, and must map it to a panel $K \subseteq \tilde{N}$.

In the next subsection, we will formally define three motives with which an agent might misreport their feature vector. All these motives revolve around controlling a particular resource: *selection probability*. Agent i's selection probability is $\mathbb{P}[i \in K]$, the probability i is chosen for the panel. We define $\pi_i^{\mathcal{A}}(p, k, \tilde{N})$ to be the selection probability given to agent i by algorithm $\mathcal{A}$ on input $p, k, \tilde{N}$. Accordingly, the vector of agents' selection probabilities is $\pi^{\mathcal{A}}(p, k, \tilde{N})$. Since p and k's true values are known to the algorithm, we simply write $\pi^{\mathcal{A}}(\tilde{N})$. A generic vector of selection probabilities is π. Note that there are k available seats for n people, so the average selection probability over agents must be k/n.

Manipulation of Selection Algorithms

In the game we study, we permit all agents to costlessly misreport any feature vector in $\mathcal{W}$. We assume that agents report their feature vector $\tilde{w}(i)$ with knowledge of the entire instance p, k, N, plus full access to the selection algorithm.[3] While the assumption that agents exactly know the true pool N is slightly adversarial, our study of simple manipulation heuristics in Section 5 will shed light the potential for manipulation using less detailed information about the pool.

We do not commit to a specific utility function for agents, because they might manipulate with a variety of different goals. Instead, we define the three measures of manipulability below, each corresponding to a different motive: the *internal* manipulability $\text{MANIP}_{\text{int}}$ captures how much a coalition can increase the selection probability of its members; the *external* manipulability $\text{MANIP}_{\text{ext}}$ captures how much a coalition can harm a non-member; and the *composition* manipulability $\text{MANIP}_{\text{comp}}$ captures how many seats (in expectation) a coalition can misappropriate from any feature-value group. We denote a coalition as C, and we let N_{-C} denote the pool with the feature vectors of $i \in C$ removed. In instance p, k, N, the manipulability of $\mathcal{A}$ by any coalition of size c is defined, per notion, as follows, where $\divideontimes := \max_{C \subseteq [n], |C| = c} \max_{\tilde{\boldsymbol{w}} \in \mathcal{W}^{|C|}}$ is shorthand for taking the worst possible coalition of size c and worst possible strategic reports of its members.

$$\text{MANIP}_{\text{int}}(N, \mathcal{A}, c) := \divideontimes \max_{i \in C} \ \pi_i^{\mathcal{A}}(N_{-C} \cup \tilde{\boldsymbol{w}}) - \pi_i^{\mathcal{A}}(N),$$

$$\text{MANIP}_{\text{ext}}(N, \mathcal{A}, c) := \divideontimes \max_{i \notin C} \ \pi_i^{\mathcal{A}}(N) - \pi_i^{\mathcal{A}}(N_{-C} \cup \tilde{\boldsymbol{w}}),$$

$$\text{MANIP}_{\text{comp}}(N, \mathcal{A}, c) :=$$

$$\divideontimes \max_{(f,v) \in FV} \sum_{i : f(i) = v} \pi_i^{\mathcal{A}}(N_{-C} \cup \tilde{\boldsymbol{w}}) - \sum_{i : f(i) = v} \pi_i^{\mathcal{A}}(N).$$

Rounding-Based Selection Algorithms

We study the manipulability of a class of selection algorithms which we call *rounding-based* selection algorithms. Each rounding-based algorithm is specified by a convex function $g : [0, 1]^n \to \mathbb{R}$; we will refer to the algorithm defined by function g simply as g. Algorithm g proceeds in two steps: Step 1 computes selection probabilities that minimize g, subject to some constraints; then, Step 2 dependently rounds these probabilities to produce a final panel. Since selection probability is the resource sought by manipulating agents — and the selection probabilities are fully determined in Step 1 — only the Step 1 will be of interest in this paper.

Step 1. Find g-optimal selection probabilities. Given instance p, k, N, in this step the algorithm optimizes g over the polytope $\mathcal{R}(N)$, defined such that $\pi \in \mathcal{R}(N) \iff \pi$ satisfies the following constraints:

$$\sum_{i \in N : f(i) = v} \pi_i = k p_{(f,v)} \ \text{ for all } (f, v) \in FV \qquad \text{(C1)}$$

$$\sum_{i \in N} \pi_i = k \qquad \text{(C2)}$$

$$\pi \in [0, 1]^n \qquad \text{(C3)}$$

[3] It is realistic to assume agents know p and k, and can access the selection algorithm: p is found in census data, and for transparency, k might be public and the selection algorithm would be open-sourced. Assuming agents know N is somewhat adversarial, because in practice, the agents report their features simultaneously; however, this assumption reflects the concern that, by comparing census data and the compositions of past pools, agents could infer who tends to participate, and thus the likely composition of N.

(C1) requires *ex-ante* representation for all feature-value pairs; (C2) requires that the panel is the correct size in expectation (required for Step 2), and (C3) requires π to contain valid probabilities. Formally, in step 1 the algorithm g solves the following convex program:

$$\min_{\pi} \; g(\pi) \quad \text{s.t. } \pi \in \mathcal{R}(N) \qquad \text{(OPT-PROB)}$$

Note that without loss of generality, we can assume that the solution of this convex program assigns the same probability to all agents with the same feature vector, since as any feasible solution can be transformed into such a solution, per the definition of $\mathcal{R}(N)$. We will consider only such solutions throughout the paper.

Step 2: Randomized-rounding. This step intakes the selection probabilities found in the previous step, called π^g, and samples a panel K of size k using the discrepancy-based rounding procedure of Flanigan et al. (2020). For our purposes, the key property of this rounding procedure is that it preserves the selection probabilities π^g; we defer the details of this procedure to Appendix A.

Specific choices of g. We will instantiate the rounding-based algorithms above with various convex functions g — all which, when minimized, tend to make selection probabilities more equal. We analyze two choices of g that serve as benchmarks: *Nash Welfare*, and *Leximin*. Nash Welfare is the geometric mean of selection probabilities:

$$\textsf{nash}(\pi) := -\prod_{i \in [n]} \pi_i.$$

Leximin is not itself strictly a function, but a refinement of the objective *Maximin*, which maximizes the minimum selection probability given to any agent:

$$\textsf{maximin}(\pi) := -\min_{i \in [n]} \pi_i.$$

The *Leximin*-optimal solution is computed iteratively: optimize maximin, fix the minimum entry of that solution as a lower bound on any entry of π, then maximize the second-lowest entry; repeat until all entries are fixed.

Finally, we study all ℓ_p norms for $p > 1$, which measure the distance between π and the vector of exactly equal selection probabilities $(k/n, k/n, \ldots, k/n)$:

$$\ell_p(\pi) := \|\pi - (k/n, \ldots, k/n)\|_p^p.$$

Connections to existing algorithms. With rounding-based algorithms defined, we can now compare them to existing selection algorithms. The most closely-related algorithm is that of Flanigan et al. (2020). Their algorithm computes selection probabilities within $\mathcal{R}$ as in our in Step 1, and then rounds them via the same procedure as in our Step 2. The main difference is that their algorithm manually sets selection probabilities to specific values in Step 1 in a way that ends up satisfying the constraints, while algorithm g within our class sets them by optimizing the function g.

Slightly further afield are the most widely-implemented maximally fair algorithms, as introduced by Flanigan et al. (2021). These algorithms differ from ours only in that they enforce representation slightly differently: instead of *ex ante* representation, they require the satisfaction of hard upper and lower demographic quotas *ex post* (e.g., quotas might require that a panel of 10 people contains between 4 and 6 women). As we show in Theorem A.2, our algorithms are formally equivalent to a continuous relaxation of these quota-based algorithms where agents are *divisible*. Moreover, our rounding-based algorithms do, in fact, achieve a relaxed version of these ex-post quotas: they are guaranteed to produce a panel containing within $\pm|F|$ of $kp_{(f,v)}$ agents with each value v of each feature f (Lemma 9, Flanigan et al. (2020)). This panel is found via a rounding scheme based on a discrepancy theorem due to Beck and Fiala (1981).

3 Leximin and Nash are Highly Manipulable

We begin by analyzing the two objectives most closely tied to practice. Strikingly, Theorem 3.1 shows that both *leximin* and *nash* are extremely manipulable: using either algorithm, an individual agent can gain selection probability 1 by misreporting, and a coalition can *deterministically* misappropriate (approaching) *half* of all panel seats for their own group. The proof of this theorem is found in Appendix B; we give a proof sketch below.

Theorem 3.1. *For an arbitrarily large n and for all $c \in [1, k/2)$, there exists an instance p, k, N, $|N| = n$ such that*

$$\begin{aligned} \text{MANIP}_{int}(N, \textsf{leximin}, 1) &= 1 \textit{ and} \\ \text{MANIP}_{int}(N, \textsf{nash}, 1) &= 1; \textit{ moreover,} \\ \text{MANIP}_{comp}(N, \textsf{leximin}, c) &= c \textit{ and} \\ \text{MANIP}_{comp}(N, \textsf{nash}, c) &= c. \end{aligned}$$

Proof sketch. Fix a $c \in [1, k/2)$. All claims are proven by a single instance p, k, N with features f_1, f_2 that take on binary values $\{0, 1\}$ (so the possible feature vectors are $00, 01, 10, 11$). In this instance, we let the population rates of all feature-values be balanced: $p_{f_1,0} = p_{f_1,1} = p_{f_2,0} = p_{f_2,1} = 1/2$. We construct N with the following fractional composition, where ν^* should be thought of as a quantity shrinking in c: $\nu_{00}(N) = \nu_{11}(N) = \nu^*$, $\nu_{10}(N) = 1 - 2\nu^*$, and $\nu_{01}(N) = 0$. We let this pool have some size $|N| = n \geq k^2$, such that its fractional composition can be realized.

First, observe that in this instance, all agents with vector 10 must receive zero selection probability *due to the constraints*: giving them any probability would induce a constraint-violating imbalance in the probability given to agents with $f_1 = 0$ versus $f_2 = 0$, which cannot be rebalanced because the complementary vector 01 does not exist in N. This suggests a manipulation strategy: an agent with 10 could misreport 01, thereby permitting greater fairness by allowing agents with 10 to receive some probability.

Let i with $w(i) = 10$, and define $\tilde{N} := N_{-i} \cup \{01\}$ as the pool resulting from i using the proposed strategy. In instance $p, k, \tilde{N}$, agents with 10 can receive probability; the catch is that, for every unit of probability given to such an agent, a unit must also be given to i, meaning that i must receive $|N|\nu_{10}$ times the probability of any agent with 10. The key observation is that both *leximin* and *nash* prioritize ensuring the *minimum* probability is not too small, with little consideration for what happens to the highest probability. For this reason, both algorithms give i selection probability

1 in the instance $p, k, \tilde{N}$. i has gained probability 1 by misreporting, implying the bounds on $\text{MANIP}_{\text{int}}(N, \textit{leximin}, 1)$ and $\text{MANIP}_{\text{int}}(N, \textit{nash}, 1)$. This argument extends to an entire coalition of $c < k/2$ such agents, implying the bounds on $\text{MANIP}_{\text{comp}}(N, \textit{leximin}, c)$ and $\text{MANIP}_{\text{comp}}(N, \textit{nash}, c)$. □

Takeaway: strongly convex objectives. The key takeaway from this proof is that objectives that do not penalize high selection probabilities can be highly manipulable. A natural class of objectives that *do* penalize high probabilities are *strongly convex* objectives — we formalize this intuition in Theorem B.1. This insight suggests that in future study of selection algorithms, it may be desirable to focus on such objectives. This finding also motivates our focus on ℓ_p norms — a natural class of strongly-convex objectives.

4 ℓ_p-Norms Approach Optimal Manipulability as $p \to \infty$

We now present upper-bounds on all three measures of manipulability for all rounding-based algorithms ℓ_p with $p > 1$. These upper bounds will hold for any instance whose pool satisfies Theorem 4.1, which conceptually requires that the pool has a minimal level of feature vector richness.

Assumption 4.1 (Pool richness)**.** N contains some set of feature-vectors $\mathcal{W}^* \subseteq \mathcal{W}$ such that

1. there is a constant $\kappa^* > 0$ such that $\nu_w(N) \geq \kappa^* + k/n$ for all $w \in \mathcal{W}^*$, and
2. $\mathcal{R}(N)$ contains a solution π^* such that $\pi_i = 0$ for all $i : w(i) \notin \mathcal{W}^*$.

This assumption is likely to hold in practice; in fact, due to how the pool is sampled, *every* feature-vector group's presence in the pool should grow approximately linearly in n. We expand on this in Appendix C. Also, note that the pool used to prove Theorem 3.1 satisfies Assumption 4.1 (Theorem C.1), thus demonstrating a genuine gap between the manipulability of all ℓ_p norms and *leximin*, *nash*.

Theorem 4.2. *Let $p > 1$, and let N be any pool of size n satisfying Assumption 4.1 with $\mathcal{W}^*, \kappa^*, \pi^*$. Let $\kappa \in (0, \kappa^*)$; then, for any coalition size $c \leq \kappa n$, we have that*

$$\text{MANIP}_{\textit{int}}(N, \ell_p, c) \in O\left(k/n^{1-1/p}\right),$$
$$\text{MANIP}_{\textit{ext}}(N, \ell_p, c) \in O\left(k/n^{1-1/p}\right), \textit{ and}$$
$$\text{MANIP}_{\textit{comp}}(N, \ell_p, c) \in O\left(ck/n^{1-1/p}\right).$$

Proof. Fix a pool N with $\mathcal{W}^*, \kappa^*, \pi^*$, as in the theorem statement. Fix any coalition $C \subseteq N$ of size $c \leq \kappa n$. Let $\tilde{N} := N_{-C} \cup \{\tilde{w}(i) | i \in C\}$ be the manipulated pool. For convenience, we will again work with feature-vector-indexed objects. We will again use $\nu_w(N)$ as the frequency of w in N. We also define $t_w(\pi) : \sum_{i:w(i)=w} \pi_i$ as the total probability π gives to agents with vector w. Let the vector of these totals be $t(\pi) = (t_w(\pi) | w \in \mathcal{W})$. We can now reformulate the constraints defining $\mathcal{R}(N)$ in terms of the variable t: let $\mathcal{T}(N) \subseteq \mathbb{R}^{|\mathcal{W}|}$ such that $t(\pi) \in \mathcal{T}(N)$ iff

$$\sum_{w:w_f=v} t_w(\pi) = k p_{(f,v)} \text{ for all } (f, v) \in FV \qquad \text{(C1')}$$
$$\sum_w t_w(\pi) = k \qquad \text{(C2')}$$
$$\frac{t_w(\pi)}{n\nu_w(N)} \in [0, 1] \text{ for all } w \in \mathcal{W} \qquad \text{(C3')}$$

Let $\pi^* \in \mathcal{R}(N)$ be the feasible solution assumed to exist by Assumption 4.1. Then, construct the vector $\tilde{\pi}$ as follows:

$$\tilde{\pi}_i = t_{w(i)}(\pi^*) \,/\, n\nu_{w(i)}(\tilde{N}) \quad \text{for all } i \in N.$$

In effect, the *total* probability assigned to each vector group from π^* to $\tilde{\pi}$ is maintained, despite the potentially changing number of agents in that group from N to $\tilde{N}$. Formally:

Claim 1: For all $w \in \mathcal{W}$, $t_w(\pi^*) = t_w(\tilde{\pi})$. *Proof:*

$$t_w(\tilde{\pi}) = \sum_{i:w(i)=w} \tilde{\pi}_i = \sum_{i:w(i)=w} \frac{t_w(\pi^*)}{n\nu_w(\tilde{N})} = t_w(\pi^*).$$

Claim 2: $\tilde{\pi} \in \mathcal{R}(N)$. *Proof:* We prove this by equivalently showing that $t(\tilde{\pi}) \in \mathcal{T}(\tilde{N})$. The satisfaction of constraints C1' and C2' follow from Claim 1. Moreover, by definition $\frac{t_w(\tilde{\pi})}{n\nu_w(\tilde{N})} \geq 0$ for all w. Then, it just remains to show C3':

$$\frac{t_w(\tilde{\pi})}{n\nu_w(\tilde{N})} = \frac{t_w(\pi^*)}{n\nu_w(\tilde{N})} \leq \frac{t_w(\pi^*)}{n(\nu_w(N) - \kappa)}$$
$$\leq \frac{t_w(\pi^*)}{n(\kappa^* + k/n - \kappa)} \leq \frac{k}{k + n(\kappa^* - \kappa)} \leq 1.$$

Now, we will show that the vectors of probabilities π^*, $\tilde{\pi}$ have maximum entry on the order $1/n$:

Claim 3: $\|\pi^*\|_\infty \leq {}^{k}/_{\kappa^* n}$ and $\|\tilde{\pi}\|_\infty \leq {}^{k}/_{(\kappa^* - \kappa)n}$. *Proof:* For all i with $w(i) \notin \mathcal{W}^*$, $\pi^*_i = \tilde{\pi}_i = 0$ by definition. For i with $w(i) \in \mathcal{W}^*$, we have that

$$\pi^*_i = \frac{t_w(\pi^*)}{n\nu_w(N)} \leq \frac{k}{n\kappa^*} \text{ and } \tilde{\pi}_i = \frac{t_w(\pi^*)}{n\nu_w(\tilde{N})} \leq \frac{k}{n(\kappa^* - \kappa)}.$$

Now, we relate the infinity-norms of any feasible solution and the ℓ_p-optimal solution of OPT-PROB:

Claim 4: For all $\pi \in \mathcal{R}(N)$, $\|\pi^{\ell_p}(N)\|_\infty \leq n^{1/p}\|\pi\|_\infty + 2kn^{-\frac{p-1}{p}}$. *Proof:* By the optimality of $\pi^{\ell_p}(N)$, we have that $\ell_p(\pi^{\ell_p}(N))^{1/p} \leq \ell_p(\pi(N))^{1/p}$. Then, using properties of norms, and the triangle inequality (twice), we obtain that

$$\|\pi^{\ell_p}(N)\|_\infty \leq \ell_p(\pi^{\ell_p}(N))^{1/p} + \|k/n1\|_p$$
$$\leq \ell_p(\pi)^{1/p} + \|k/n1\|_p$$
$$\leq \|\pi\|_p + 2\|k/n1\|_p \leq n^{1/p}\|\pi\|_\infty + 2kn^{\frac{1-p}{p}}.$$

Using that $\pi^* \in \mathcal{R}(N)$, $\tilde{\pi} \in \mathcal{R}(\tilde{N})$, Claims 3 and 4 together imply that $\|\pi^{\ell_p}(N)\|_\infty \leq k/(\kappa^* n^{1-1/p}) + 2k/n^{1-1/p}$ and likewise, $\|\pi^{\ell_p}(\tilde{N})\|_\infty \leq k/((\kappa^* - \kappa)n^{1-1/p}) + 2k/n^{1-1/p}$. Using that the entries of all π are nonnegative, it follows that

$$\|\pi^{\ell_p}(\tilde{N}) - \pi^{\ell_p}(N)\|_\infty \leq \left(\frac{1}{\kappa^* - \kappa} + 2\right) \frac{k}{n^{1-1/p}}. \qquad (1)$$

We've now shown an upper bound on how many any i's probability changes between pool N and pool $\tilde{N}$. This immediately implies the upper bounds on $\text{MANIP}_{\text{int}}(N, \ell_p, c)$ and $\text{MANIP}_{\text{ext}}(N, \ell_p, c)$. Our upper bound on $\|\pi^{\ell_p}(\tilde{N})\|_\infty$ further implies that post-defection, the members of the coalition can have at most $O(ck/n^{1-1/p})$ total selection probability, giving our upper bound on $\text{MANIP}_{\text{comp}}(N, \ell_p, c)$. □

We now show a lower bound that applies to *any* rounding-based algorithm. It shows that up to constants, the manipulability of ℓ_∞ decreases at the *optimal* rate in n.

Theorem 4.3. *There is some $\eta > 0$ such that there exist pools N of arbitrarily large size n which, for any coalition size $c \leq 5n/64$ and all objectives g, satisfy*

$$\text{MANIP}_{int}(N, g, c) \geq \eta\, k/n, \quad \text{MANIP}_{ext}(N, g, c) \geq \eta\, k/n,$$
$$\text{MANIP}_{comp}(N, g, c) \geq \eta\, ck/n.$$

The same pools also satisfy Theorem 4.1.

The proof is in Appendix C and relies on an example exactly like Theorem 1.1: there is one binary feature, where v_1 is severely underrepresented in the pool. The bounds arise from agents with v_0 misreporting v_1.

5 Manipulability of Real-World Instances

Now we compare the manipulability of *leximin*, *nash*, ℓ_2 and ℓ_∞ in eight real-world panel selection instances. Instance details are provided in Appendix D. We present here two representative instances, called *sf(a)* and *hd*, and defer the rest to Appendix D. The datasets were obtained from groups of assembly organizers based in the UK and US, respectively. Each real-world instance consists of p, k, N. To study how manipulability changes as we increase the pool size, we simply copy the pool, leaving p and k fixed. In each instance, we copy the pool until $n \geq 100k$, as practitioners often specify their target pool size in multiples of k.

We will test our selection algorithms against an *individual* manipulator — that is, we measure how much selection probability any agent can gain by misreporting their feature vector. The most powerful individual manipulator could gain $\text{MANIP}_{\text{int}}(N, \mathcal{A}, 1)$ probability against $\mathcal{A}$ — the quantity to which our theoretical bounds apply. Given the computational difficulty of calculating the optimal manipulation (each agent has $|\mathcal{W}| \in \Omega(2^{|F|})$ possible strategies), we test our algorithms against three practically-motivated heuristic strategies: *OPT-1, MU,* and *HP*, defined below. The results are summarized in Figure 1.

OPT-1: *Optimal misreport of one feature*. An agent playing strategy *OPT-1* reports the feature vector that benefits them most, *subject to misreporting their value for at most one feature*. This strategy, in practice, might correspond to a practical setting in which only a few features cannot be validated. When comparing across algorithms, we think of OPT-1 as a proxy for the optimal individual manipulation. As column 1 of Figure 1 shows, the manipulability of ℓ_2 and ℓ_∞ against OPT-1 declines quickly in n, while *leximin* and *nash* remain arbitrarily susceptible to manipulation. The fact that *leximin* and *nash* are so manipulable *even when agents are willing to misreport only one feature* was not implied by our lower bounds, and shows the findings in our theoretical lower bounds are of practical relevance.

MU: *Most underrepresented*. Let $\eta_{(f,v)}(N) := |\{i | f(i) = v\}|/|N|$ be the fraction of agents with value v for feature f. An agent playing strategy *MU* reports the vector containing the most underrepresented value of each feature f — that is, $\tilde{w}_f := \arg\max_{v \in V_f} p_{(f,v)}/\eta_{(f,v)}(N)$. Again, *leximin* and *nash* are arbitrarily manipulable against *MU*, even for large n. The vulnerability of *leximin* and *nash* here is of especially high practical concern, because the *MU* manipulation strategy is perhaps the most likely to be used in practice by less sophisticated manipulators: it is intuitive and requires only ordinal information about (the only $O(|F|)$ many) feature-value frequencies and no access to the algorithm (in contrast, *OPT-1* and *HP* require algorithm access *and* information about the pool's vector-level composition).

HP: *Highest-Probability*. Another reasonable heuristic a manipulator i might use would be to report the vector $\tilde{w}$ that receives the highest selection probability in the true pool; we call this heuristic *HP*. That this strategy's efficacy declines in n intuitively makes sense: misreporting a vector that is already in the pool means joining a vector group whose size is growing linearly in n (at least in these experiments, where we are duplicating N). This intuition alludes to the insight that the most problematic misreports for suboptimal algorithms are those of vectors that do not already exist in the pool — an intuition supported by both the proof of our lower bound in Theorem 3.1, and the fact that the most underrepresented vector (targeted by the much more effective strategy *MU*) is not in the original pool of any instance we study.

Extension: Manipulability and *Selection Bias*

While n is much easier to change in practice than the level of self-selection bias (SSB), the SSB could be decreased by a more targeted recruitment process, motivating our study of this would impact the manipulability. We introduce a measure of SSB in an instance, which roughly captures how severely the algorithm must skew selection probabilities to satisfy the constraints:

$$\Delta_{p,k,N} := \max_{(f,v) \in FV} \frac{p_{(f,v)}}{\eta_{(f,v)}(N)} - \min_{(f,v) \in FV} \frac{p_{(f,v)}}{\eta_{(f,v)}(N)}$$

Figure 2(a) shows that this measure of SSB is highly predictive of manipulability: across instances, the manipulation gain of OPT-1 (scaled by k/n, for standardization) against ℓ_∞ corresponds closely with instances' $\Delta_{p,k,N}$ values, as listed in the figure legend. Proceeding with this measure, we evaluate the impact of decreasing it in two ways. First, in Figure 2(b), we decrease the SSB smoothly by *interpolating* between the original pool N and the "nearest" (by Euclidean distance) pool N' with $\Delta_{p,k,N'} = 0$. Second, in Figure 2(c), we decrease the SSB by successively dropping features from the instance in decreasing order of their *feature-level* SSB, defined as $\Delta_{p,k,N}$ restricted to the values of a given feature. Using either approach, in *sf(a)*, the manipulability of all algorithms except *leximin* against *OPT-1* drops quickly, while *leximin* remains manipulable until extremely low levels of SSB are reached. We defer the details of these methods, plus results for the remaining instances, to Appendix D.

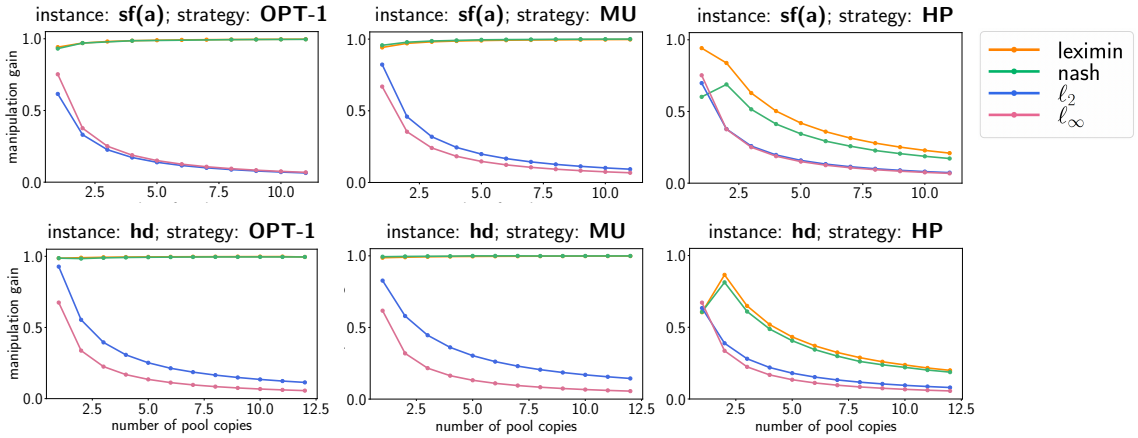
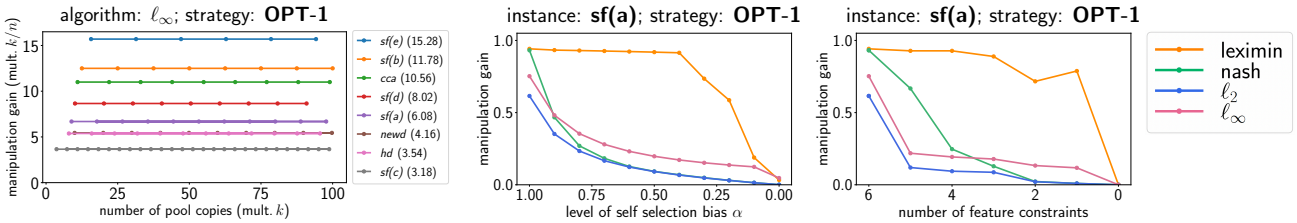


Figure 1: Rounding-based algorithms *leximin*, *nash*, ℓ_2, and ℓ_∞ versus each manipulation strategy in instances *sf(a)* and *hd*.



(a) (b) (c)

Figure 2: The impact of self-selection bias on the manipulability of *leximin*, *nash*, ℓ_2 and ℓ_∞ by an agent playing *OPT-1* strategy.

6 Discussion

Our work illuminates a tradeoff between two goals: ensuring that no one gets too *little* selection probability (as pursued in the related work (Flanigan et al. 2021)), and ensuring that no one gets too *much* probability (which we show is important for limiting manipulation incentives). *leximin* and *nash* prioritize the first goal but, as we show, perform poorly on the second. In contrast, we show that ℓ_p norms can be optimal in regards to the second goal, but they perform poorly on the first: we find that both ℓ_2 and ℓ_∞ give at least one agent zero probability in all eight instances we study (see Appendix D). This begs the question: *is there an objective that both prevents high probabilities (thereby limiting manipulability) as well as low probabilities?* An objective with *optimal* dependency on n for both desiderata at once would give all agents $\Theta(1/n)$ probability.[4]

Another first-order technical extension of this work would be to repeat this analysis within *quota-based* algorithms, as they implement the notion of representation most commonly used (Flanigan et al. 2021). Because the separation between *leximin*, *nash* versus ℓ_p norms is due to fundamental properties of these objectives, we expect them to exhibit roughly similar behavior in quota-based algorithms. However, the combinatorial structure of quotas may make quota-based algorithms much *more* manipulable in the worst case.

[4] $\Theta(1/n)$ is the optimal rate at which manipulability can decline (Theorem 4.3); because any algorithm must divide k probability over n people, the minimum probability can be at most $\Theta(1/n)$.

Even without this extension to quota-based algorithms, our work raises some practical insights. First, it suggests that in general, algorithms permitting high selection probabilities come with risks of manipulability — a property that can be tested in any selection algorithm, maximally fair or not. If one *does* maximize a carefully chosen fairness objective, our work reveals practicable strategies for limiting manipulation incentives: decreasing the SSB (even simply by dropping features that one expects to be highly self-selected), or recruiting a larger pool. Based on our empirical results, even doubling the pool sizes currently used in practice would substantially decrease manipulability.

Beyond the application of assembly selection, our problem is conceptually reminiscent of *strategic classification*, in which agents may misreport their features to increase their probability of receiving a desirable prediction from a machine-learned classifier (Hardt et al. 2016; Dong et al. 2018; Chen, Liu, and Podimata 2020; Ahmadi et al. 2021). Within the strategic classification framework, we can view a selection algorithm as a *constrained* classifier: one which classifies agents as either on or off the panel with some probability based on their features, while satisfying demographic representation constraints on who receives a positive classification. While some existing work is tangentially related (Liu, Garg, and Borgs 2022), to our knowledge this precise problem has not been studied in the strategic classification literature. Our notions of manipulability, and our technical results on the stability of our convex program, may be of interest for this domain.

Acknowledgements

We thank Thibaut Horel and Paul Gölz for helpful technical discussions; the reviewers for their excellent feedback; and with the several organizations that provided real-world citizens' assembly data including the Sortition Foundation, the Center for Blue Democracy, Healthy Democracy, and New Democracy. This work was partially supported by the National Science Foundation under grants IIS-2147187, IIS-2229881 and CCF-2007080; and by the Office of Naval Research under grant N00014-20-1-2488 (AP); the AFOSR Multidisciplinary University Research Initiative (MURI) project ANSRE (SW); and a Fannie and John Hertz Foundation Fellowship and a National Science Foundation Graduate Research Fellowship (BF).

References

Ahmadi, S.; Beyhaghi, H.; Blum, A.; and Naggita, K. 2021. The strategic perceptron. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, 6–25.

Bansal, N. 2019. On a generalization of iterated and randomized rounding. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 1125–1135.

Beck, J.; and Fiala, T. 1981. "Integer-making" theorems. *Discrete Applied Mathematics*, 3(1): 1–8.

Bürgerrat. 2023. French citizens' assembly supports assisted dying. https://www.buergerrat.de/en/news/french-citizens-assembly-supports-assisted-dying/. Accessed: 2023-08-10.

Chen, Y.; Liu, Y.; and Podimata, C. 2020. Learning strategy-aware linear classifiers. *Advances in Neural Information Processing Systems*, 33: 15265–15276.

Dong, J.; Roth, A.; Schutzman, Z.; Waggoner, B.; and Wu, Z. S. 2018. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 55–70.

Ebadian, S.; Kehne, G.; Micha, E.; Procaccia, A. D.; and Shah, N. 2022. Is Sortition Both Representative and Fair? *Advances in Neural Information Processing Systems*, 35.

Ebadian, S.; and Micha, E. 2023. Boosting Sortition via Proportional Representation. Manuscript.

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. *Advances in Neural Information Processing Systems*, 33: 6528–6539.

Flanigan, B.; Kehne, G.; and Procaccia, A. D. 2021. Fair sortition made transparent. *Advances in Neural Information Processing Systems*, 34: 25720–25731.

Hardt, M.; Megiddo, N.; Papadimitriou, C.; and Wootters, M. 2016. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, 111–122.

Liu, L. T.; Garg, N.; and Borgs, C. 2022. Strategic ranking. In *International Conference on Artificial Intelligence and Statistics*, 2489–2518.

Participedia. 2023. https://participedia.net/search?selectedCategory=case&recruitment_method=random,stratified. Accessed: 2023-08-14.